

Санкт-Петербургский государственный университет
Математико-механический факультет

О. Н. ИВАНЦОВА, И. Д. МИРОШНИЧЕНКО

**ПРАКТИКУМ ПО ДИСЦИПЛИНЕ
«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В
ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»**

Санкт-Петербург
2023

Методическое пособие «Практикум по дисциплине «Информационные технологии в профессиональной деятельности»» предназначено для обучающихся междисциплинарного направления по образовательной программе «Физкультура и спорт» дисциплины «Информационные технологии в профессиональной деятельности». Методическое пособие ориентировано на освоение студентами, может быть полезно офисным работникам.

Методические материалы будут полезны тем обучающимся, которые проявляют интерес к эффективному использованию широкого диапазона возможностей офисных приложений MS Windows, в частности, при подготовке документов.

Методическое пособие включает подробное рассмотрение как теоретических вопросов представления данных, так и заданий для самостоятельной работы, а также подробные инструкции по подготовке документов с использованием блокнота и его аналогов, электронных таблиц и графики (то есть бесплатных приложений, включенных в пакет MS Office)

*Авторы: Иванцова О. Н. — кандидат физ.-мат. наук, доцент кафедры
параллельных алгоритмов СПбГУ,
Мирошниченко И. Д. — старший преподаватель кафедры
параллельных алгоритмов СПбГУ*

*Рецензенты: доктор ф.-м. наук, профессор ВМПИ Аксенова О.А.
кандидат ф.-м. наук, доцент СПбГУ Бухвалова В.В.*

Одобрено и рекомендовано для использования обучающимися Учебно-методической комиссией по УГСН 02.00.00 Компьютерные и информационные науки.
Протокол № 05/2.1/02-03-1 от 27.01.2023 г.

Информация и данные.

Двоичное кодирование информации.

Системы счисления

Информация

Мы живем в материальном мире. Все физические объекты, окружающие нас, находятся в состоянии непрерывного движения и изменения, которое сопровождается обменом энергией и ее переходом из одной формы в другую. Все виды энергообмена сопровождаются появлением сигналов. При взаимодействии сигналов с физическими телами в последних возникают определенные изменения свойств — это явление называется **регистрацией сигналов** (или образованием данных).

Данные — это зарегистрированные сигналы.

Физический метод регистрации сигналов может быть любым (например, механическое перемещение физических тел, изменение их электрических характеристик и др.). В соответствии с методом регистрации данные могут храниться и транспортироваться на носителях различных видов (например, бумага, диски и т.д.).

Информация — это продукт взаимодействия данных и адекватных им методов. Информация динамически меняется и существует только в момент взаимодействия данных и методов. Все остальное время она пребывает в состоянии данных. Таким образом, информация существует только в момент протекания информационного процесса. Заметим, что данные являются объективными, а методы — субъективными.

В ходе информационного процесса данные преобразуются из одного вида в другой с помощью методов. Обработка данных включает в себя множество различных операций:

- *сбор данных* — накопление информации для принятия решений;
- *формализация данных* — приведение данных из разных источников к одинаковой форме;
- *фильтрация данных* — отсеивание данных, в которые не нужны для принятия решения;
- *сортировка данных* — для повышения доступности информации;
- *архивация данных* — организация хранения данных в удобной и легкодоступной форме для снижения затрат по хранению данных;
- *защита данных* — предотвращение утраты, воспроизведения и модификации данных;
- *транспортировка данных* — прием и передача данных между удаленными участниками информационного процесса;
- *преобразование данных* — перевод данных из одной формы (структуры) в другую.

Часто связано с изменением типа носителя.

Итак, работа с информацией может иметь огромную трудоемкость, и ее надо автоматизировать.

Кодирование информации

Для автоматизации работы с данными, относящимися к различным типам, важно унифицировать их форму представления — для этого используется прием кодирования, то есть выражение данных одного типа через данные другого типа (например, система записи математических выражений, телеграфная азбука, семафор и др.).

Своя система существует и в вычислительной технике — она называется **двоичным кодированием** и основана на представлении данных последовательностью двух знаков: 0 и 1. Эти знаки называются двоичными цифрами (**B**inary **digi**T) или сокращенно **бит** (**BIT**). Название ввел в обращение американский ученый-статистик Джон Тьюки в 1946 году.

С помощью двоичного кодирования можно, например, кодировать целые и действительные числа.

Системы счисления (способ кодирования чисел)

Система счисления — способ наименования и изображения чисел с помощью символов, имеющих определенные количественные значения.

В зависимости от способа изображения чисел системы делятся на:

- позиционные
- непозиционные

В **непозиционной** системе счисления цифры не меняют своего количественного значения при изменении их расположения в числе.

Примеры: римская, древнерусская (алфавитная и глагольная), древнеегипетская и др.

Единицы	Десятки	Сотни	Тысячи
1 I	10 X	100 C	1000 M
2 II	20 XX	200 CC	2000 MM
3 III	30 XXX	300 CCC	3000 MMM
4 IV	40 XL	400 CD	
5 V	50 L	500 D	
6 VI	60 LX	600 DC	
7 VII	70 LXX	700 DCC	
8 VIII	80 LXXX	800 DCCC	
9 IX	90 XC	900 CM	

Рис. 1. Римская непозиционная система счисления

1	2	3	4	5	6	7	8	9
·ā·	·b·	·r̄·	·d̄·	·e·	·s̄·	·z̄·	·h̄·	·q̄·
10	20	30	40	50	60	70	80	90
·r̄·	·k̄·	·ā·	·m̄·	·h̄·	·z̄·	·o·	·p̄·	·q̄·
100	200	300	400	500	600	700	800	900
·p̄·	·e·	·t̄·	·v̄·	·d̄·	·x̄·	·ψ·	·ω·	·π·
11	12	13	14	15	16	17	18	19
·āi·	·bī·	·rī·	·dī·	·eī·	·sī·	·zī·	·hī·	·qī·
222	319	431	519	631	719	819	919	
·gkb·	·tφi·	·vll·	·цпи·					
1000	2000	20000	43000					
·a·	·b·	·k·	·m·g·					
10000	300000	4000000	80000000					
·a·	·r̄·	·a·	·h̄·					

Рис. 2. Древнерусская алфавитная система счисления

число	значение	описание
1	1	черта
10	10	пятка
100	100	петля веревки
1 000	1 000	кувшинка (или лотос)
10 000	10 000	палец
100 000	100 000	жаба или личинка
1 000 000	1 000 000	человек с поднятыми вверх руками

Рис. 4. Египетская система счисления

1	10	100	1000
А аз	И и*	Р рцы	С слово
В веди	К како	Т твердо	У ук**
Г глаголь	Л люди	Ф ферг	Х хер
Д добро	М мыслете	Ц ци	
Е есть**	Н наш**		
З зело*	Кси**		
З земля**	О он		
Н иже**	П покой		
А фита*	У червь		

Славянская глаголическая нумерация

1	10	100	1000
2	20	200	
3	30	300	
4	40	400	
5	50	500	
6	60	600	
7	70	700	
8	80	800	
9	90	900	

ψ ρ ρ 800 + 60 + 3 = 863

Рис. 3. Славянская глаголическая система счисления

На рисунке 1 представлен пример римской системы счисления, на рисунке 2 – древнерусской алфавитной системы счисления, которая применялась вплоть до XVII века (потом и до сего времени только в церковных старославянских книгах), вытеснена индо-арабской позиционной системой счисления, на рисунке 3 – славянская глаголическая система счисления, на рисунке 4 - египетская система счисления.

Почти каждая буква древнерусского алфавита представляла собой не только букву, но и цифру. Так, например, буква «Аз» означала «один», «Веди» — «два» ... Некоторые буквы, а именно: буки, живете, ша, ща, ер, еры, ерь, ять, ю, А йотированное, Е йотированное и юсы не имели числовых соответствий.

Числительные писались и произносились слева направо за исключением числительных от 11 до 19 (например, 17 — *сем-на-дцать*).

В начале XVIII века иногда применялась смешанная система записи чисел, состоящая и из кириллических, и из арабских цифр. Например, на некоторых медных копейках отчеканена дата 17K1 (1721).

По тому же принципу строилась глаголическая система счисления. В ней использовались буквы глаголицы. Записывались цифры числа начиная с больших значений и заканчивая меньшими, слева направо. Если десятков, единиц, или какого-то другого разряда не было, то его пропускали

Славянская глаголическая система счисления была создана для переписки чисел в священных книгах западных славян. Использовалась она нечасто, но достаточно долго. По организации она в точности повторяет греческую нумерацию, использовалась с VIII по XIII век. Такая запись числа аддитивная, то есть в ней используется только сложение:

$$\text{Ѡ Ѡ Ѡ Ѡ Ѡ} = 800+60+3 = 863$$

Для того, чтобы не перепутать буквы и цифры, использовались титла - горизонтальные черточки над числами, или точки.

В египетской системе счисления числа, не являющиеся степенью 10, записывались путём повторения этих цифр. Каждая цифра могла повторяться от 1 до 9 раз. Например,



число 4622 обозначалось следующим образом:



Фиксированного направления записи чисел не существовало: они могли записываться справа налево или слева направо и даже вертикально. Например, иероглифическая запись  и обратная запись тех же иероглифов  обозначали одно и то же число — «12».

В римской **непозиционной** системе счисления число 30 – XXX представлено суммой X+X+X, причем совершенно неважно на каком месте справа или слева стоит символ X.

В **позиционной** системе счисления количественное значение каждой цифры зависит от ее (места) позиции в числе.

Так, в числе 353 десятичной системы счисления цифра 3 - первая тройка справа налево - означает 3 единицы, а третья по счету справа налево цифра, то есть тоже цифра 3, но она означает уже 300 (три сотни).

Видим, что каждая позиция в позиционной системе счисления имеет свой “вес” - количественное значение. Самая правая позиция в приведенном примере числа 353 в десятичной системе счисления имеет номер позиции 0 и, соответственно, вес 10^0 , а код для этого веса – 3, следующая позиция справа налево имеет вес 10^1 и код 5, наконец, последняя позиция в этом числе имеет вес 10^2 и код 3.

Это можно записать следующим образом: $353_{10} = 3 \cdot 10^2 + 5 \cdot 10^1 + 3 \cdot 10^0$. Такая запись соответствует коду числа в десятичной системе счисления или записи разложения десятичного числа $353_{10} = 3 \cdot 100 + 5 \cdot 10 + 3$, как писали в начальной школе

Количество (p) различных цифр, используемых для изображения числа в позиционной системе счисления, называется **основанием** системы счисления.

Значения цифр лежат в пределах от 0 до $p-1$.

Рассмотрим следующие наиболее распространенные в информатике позиционные системы счисления (буквой A обозначим алфавит):

- Двоичная система счисления имеет в основании ($p = 2$) две цифры 0 и 1 $A_2 = \{0, 1\}$.
- Восьмеричная система счисления имеет в основании ($p = 8$) цифры от 0 до 7 $A_7 = \{0, 1, 2, 3, 4, 5, 6, 7\}$.

- Десятичная система счисления имеет в основании ($p = 10$) цифры от 0 до 9 $A_{10}=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
- Шестнадцатеричная система счисления имеет в основании ($p = 16$) цифры от 0 до 9 и буквы A, B, C, D, E, F $A_{16}=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

Существуют и другие позиционные системы счисления, например:

- система с основанием 60 используются нами для подсчета времени (минут, секунд),
- система с основанием 24 – для подсчета числа часов в сутках,
- 12 (дюжина) — это число также используется во временных расчетах, а ранее дюжина широко использовалась и в торговле (вспомните чертову дюжину),
- следы от основания 40 можно услышать во фразе “сорок сороков”.

Задание для самостоятельной работы.

Подумайте и приведите еще примеры оснований систем счисления.

Двоичное кодирование целых чисел

Алгоритм двоичного кодирования целых чисел состоит в следующем: целое число делится пополам до тех пор, пока частное не будет равно единице. Совокупность остатков от каждого деления, записанная справа налево вместе с последним частным, и образует двоичный код целого десятичного числа.

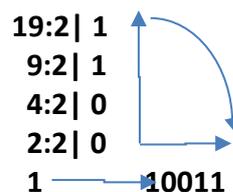
Пример:

Дано: десятичное число 19.

Найти: двоичный код данного числа.

Решение (1 способ):

- 19 : 2 = 9 (остаток от деления 1)
- 9 : 2 = 4 (остаток от деления 1)
- 4 : 2 = 2 (остаток от деления 0)
- 2 : 2 = 1 (остаток от деления 0)



Ответ: 10011

Нетрудно догадаться, что данный алгоритм описывает перевод целых чисел из десятичной системы счисления в двоичную систему счисления.

Используем для обозначения системы счисления нижний индекс, например, 19_p будет обозначать число 19 в системе счисления с основанием p (где $p \geq 10$).

Таким образом, из выше сказанного получаем, что $19_{10} = 10011_2$.

Схематичное изображение справа представляет алгоритм быстрого перевода целых чисел из десятичной системы счисления в двоичную систему счисления (справа от черты стоят остатки от деления на 2, берем эти остатки подряд от нижней единицы вверх и записываем горизонтально слева направо).

Решение (2 способ).

Более сложную цепочку преобразований для перевода из десятичной системы счисления в двоичную систему счисления, но зато наглядно представляющую смысл этого перевода приведем ниже. Возьмем тоже число 353_{10} и представим его запись относительно весов двоичной системы счисления, не пропуская ни одной последовательной степени. Сначала найдем наибольшую степень двойки, которая “помещается” в десятичное число 353_{10} . Это — 2^8 , равное $256_{10} < 353_{10}$, а $512_{10} > 353_{10}$. Запишем $353_{10} = 1 \cdot 2^8 + (353 - 256)_{10} = 1 \cdot 2^8 + 97_{10}$.

Следующая степень двойки – $2^7 = 127_{10}$ не присутствует в остатке числа, то есть в 97_{10} , значит, нужно написать $353_{10} = \underline{1} \cdot 2^8 + \underline{0} \cdot 2^7 + 97_{10}$

Ищем следующую степень двойки $2^6 = 64_{10}$. $64_{10} < 97_{10}$, значит 2^6 “помещается” в 97_{10} . Можем написать $353_{10} = \underline{1} \cdot 2^8 + \underline{0} \cdot 2^7 + \underline{1} \cdot 2^6 + (97 - 64)_{10} = \underline{1} \cdot 2^8 + \underline{0} \cdot 2^7 + \underline{1} \cdot 2^6 + 33_{10}$. Ищем следующую степень двойки - $2^5 = 32_{10}$. $32_{10} < 33_{10}$, записываем $353_{10} = \underline{1} \cdot 2^8 + \underline{0} \cdot 2^7 + \underline{1} \cdot 2^6 + \underline{1} \cdot 2^5 + (33 - 32)_{10} = \underline{1} \cdot 2^8 + \underline{0} \cdot 2^7 + \underline{1} \cdot 2^6 + \underline{1} \cdot 2^5 + 1_{10}$.

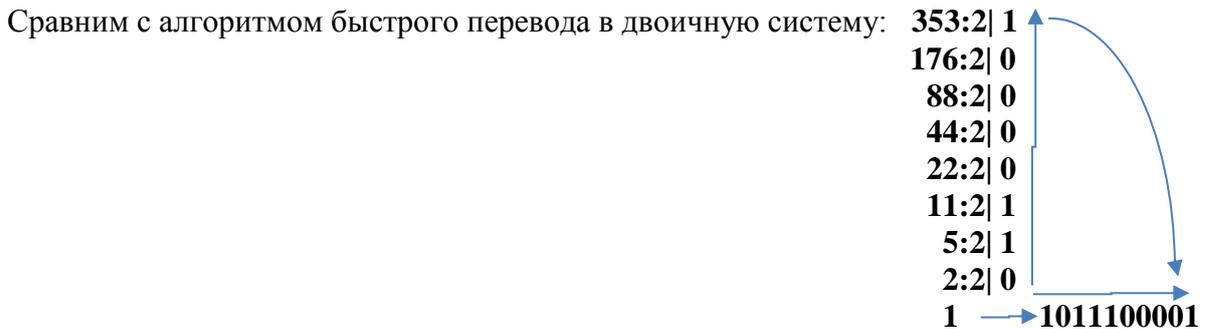
Очевидно, что $2^4 = 16_{10}$ не “помещается” в 1_{10} , то есть $353_{10} = \underline{1} \cdot 2^8 + \underline{0} \cdot 2^7 + \underline{1} \cdot 2^6 + \underline{1} \cdot 2^5 + \underline{0} \cdot 2^4 + 1_{10}$.

Со степенью $2^3=8_{10}$ – та же история: $2^3=8_{10}$ не “помещается” в 1_{10} , то есть $353_{10}=\underline{1}*2^8+\underline{0}*2^7+\underline{1}*2^6+\underline{1}*2^5+\underline{0}*2^4+\underline{0}*2^3+\underline{1}_{10}$. Далее $2^2=4_{10}$, $2^2=4_{10}$ не “помещается” в 1_{10} , значит $353_{10}=\underline{1}*2^8+\underline{0}*2^7+\underline{1}*2^6+\underline{1}*2^5+\underline{0}*2^4+\underline{0}*2^3+\underline{0}*2^2+\underline{1}_{10}$.

Предпоследняя степень двойки $2^1=2_{10}$, $2^1=2_{10}$ не “помещается” в 1_{10} , значит $353_{10}=\underline{1}*2^8+\underline{0}*2^7+\underline{1}*2^6+\underline{1}*2^5+\underline{0}*2^4+\underline{0}*2^3+\underline{0}*2^2+\underline{0}*2^1+\underline{1}_{10}$.

И, наконец, $2^0=1_{10}$, тогда $353_{10}=\underline{1}*2^8+\underline{0}*2^7+\underline{1}*2^6+\underline{1}*2^5+\underline{0}*2^4+\underline{0}*2^3+\underline{0}*2^2+\underline{0}*2^1+\underline{1}*2^0$.

Собирая все эти коды (коэффициенты) при весах, получим код десятичного числа 353_{10} в двоичной системе счисления: 1011100001_2 .



Ваше право пользоваться любым из этих алгоритмов.

Перевод целых чисел из одной системы счисления в другую.

Аналогичные переводы можно осуществить из десятичной (десятеричной) системы в восьмеричную и шестнадцатеричную. Покажем это. Переведем сначала из десятичной системы в восьмеричную то же число 353_{10} .

Воспользуемся 2 способом. Рассмотрим степени **8**: это $8^0=1_{10}$, $8^1=8_{10}$, $8^2=64_{10}$, $8^3=512_{10}$.

Очевидно, наибольшая степень **8**, которая подходит для представления числа 353_{10} , это 8^2 , так как $8^3=512_{10}>353_{10}$. Но, в отличие от двоичного алфавита, в котором только две цифры (и коэффициент принимает значение только 0 или 1), в восьмеричном алфавите восемь цифр (поэтому коэффициентом может быть любое число от 0 до 7), и, следовательно, слагаемыми могут быть любые числа вида: $0*8^n$, $1*8^n$, $2*8^n$, ..., $7*8^n$.

Итак, найдем наибольшее число только что описанного вида, которое “помещается” в 353_{10} . Это $5*8^2=320_{10}$, так как $6*8^2=384_{10}>353_{10}$.

Запишем $353_{10}=5*8^2+(353-320)_{10}=\underline{5}*8^2+33_{10}$.

Следующая степень восьмеричной системы это $8^1=8_{10}$. Очевидно подходящий коэффициент k для степени 8^1 ($k*8^1<33_{10}$) это **4**, то есть $4*8^1=32<33_{10}$.

Получаем $353_{10}=\underline{5}*8^2+\underline{4}*8^1+(33-32)_{10}=353_{10}=\underline{5}*8^2+\underline{4}*8^1+\underline{1}_{10}$.

Осталось найти коэффициент при степени 8^0 . Очевидно, это $1*8^0$.

Получаем $353_{10}=\underline{5}*8^2+\underline{4}*8^1+\underline{1}*8^0$.

Таким образом, восьмеричный код десятичного числа 353_{10} есть 541_8 .



Оба способа дали одинаковые результаты.

Теперь выполним перевод из десятичной системы в шестнадцатеричную для того же числа 353_{10} .

Применим 2 способ. Рассмотрим степени основания **16**: $16^0=1_{10}$, $16^1=16_{10}$, $16^2=256_{10}$. Очевидно, следующую степень рассматривать нет смысла. Таким образом, максимальная степень **16**, которая “помещается” в 353_{10} , это 16^2 . Причем, коэффициентом при этой степени может быть только **1**, потому что $2*16^2=512_{10}>353_{10}$.

Значит, $353_{10}=1*16^2+(353-256) = \underline{1}*16^2+\underline{97}_{10}$.

Теперь для следующей степени 16^1 найдем коэффициент, чтобы полученное число было максимальным и “помещалось” в 97_{10} . Этот коэффициент – число 6 , так как $6 \cdot 16^1 = 96_{10}$. Таким образом, $353_{10} = 1 \cdot 16^2 + 6 \cdot 16^1 + (97 - 96)_{10} = \underline{1} \cdot 16^2 + \underline{6} \cdot 16^1 + \underline{1} \cdot 16^0$.

Получили шестнадцатеричное представление числа $353_{10} = 161_{16}$.

Проверим результат, получаемый *1 способом*: $353:16 \mid 1$
 $16 \cdot 22 = 352 \quad 353 - 352 = 1 (\text{остаток})$
 $22:16 \mid 6$
 1
 161_{16}

Видим, что оба способа дали одинаковые результаты.

Рассмотрели способы перевода чисел из $(10) \rightarrow (2)$, $(10) \rightarrow (8)$, $(10) \rightarrow (16)$.

Рассмотрим теперь алгоритм перевода

- ♦ из двоичной системы счисления в десятичную,
- ♦ из восьмеричной системы в десятичную и
- ♦ из шестнадцатеричной системы в десятичную систему счисления.

Рассмотрим некоторое двоичное число: 1101010011_2

Как выяснено было раньше, каждая позиция этого числа согласно позиционной системе счисления имеет свой вес, соответствующий степени 2 (считая от 0 степени справа налево).

Переведем это двоичное число в десятичное число согласно выше приведенным правилам.

$$\begin{aligned} 1101010011_2 &= 1 \cdot 2^9 + 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = \\ &= 1 \cdot 512 + 1 \cdot 256 + 0 \cdot 127 + 1 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 512 + 256 + 64 + 16 + 2 + 1 = \\ &= (512 + 16 + 2) + (256 + 64) + 1 = 530 + 320 + 1 = \mathbf{851_{10}} \end{aligned}$$

Переведем восьмеричное число в десятичное число согласно выше приведенным правилам.

$$357_8 = 3 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0 = 3 \cdot 64 + 5 \cdot 8 + 7 \cdot 1 = 192 + 40 + 7 = 239_{10}.$$

Переведем шестнадцатеричное число в десятичное число согласно выше приведенным правилам:

$$3FA_{16} = 3 \cdot 16^2 + F \cdot 16^1 + A \cdot 16^0 = 3 \cdot 256 + 15 \cdot 16 + 10 \cdot 1 = 768 + 240 + 10 = 1018_{10}.$$

Напомним, что $F_{16} = 15_{10}$, $E_{16} = 14_{10}$, $D_{16} = 13_{10}$, $C_{16} = 12_{10}$, $B_{16} = 11_{10}$, $A_{16} = 10_{10}$.

Рассмотрели способы перевода чисел из $(2) \rightarrow (10)$, $(8) \rightarrow (10)$, $(16) \rightarrow (10)$.

Для перевода из двоичной системы в восьмеричную или шестнадцатеричную существуют специальные алгоритмы, называемые методами триад (для восьмеричной системы) и тетрады (для шестнадцатеричной системы).

Суть методов в том, что число 8 есть 3 -ья степень числа 2 : $8_{10} = 2^3$, а 16 есть 4 -ая степень числа 2 : $16_{10} = 2^4$. Иначе говоря, каждый символ 8 -ричной системы счисления представляется 3 -мя битами, а 16 -ричной системы – 4 -мя битами.

Покажем, как эффективно число в двоичной системе счисления перевести в восьмеричную систему счисления $(2) \rightarrow (8)$. Возьмем для этого двоичное число 1101010011_2 .

Справа налево отделим по 3 символа: $1101010011_2 = 1 \underline{101} \underline{010} \underline{011}_2$

Если необходимо, слева можно добавить незначащие нули, как в нашем примере:

$$1101010011_2 = 1 \underline{101} \underline{010} \underline{011}_2 = \underline{001} \underline{101} \underline{010} \underline{011}_2$$

Затем каждую триаду независимо друг от друга заменим на восьмеричную цифру:

$$1101010011_2 = 1 \underline{101} \underline{010} \underline{011}_2 = \underline{001} \underline{101} \underline{010} \underline{011}_2 = 1 \ 5 \ 2 \ 3_8$$

Каждая триада переводится обычным способом, например:

$$000_2 = 0_8 \quad 100_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 4_8$$

$$001_2 = 1_8 \quad 101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5_8$$

$$010_2 = 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 2_8 \quad 110_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6_8$$

$$011_2 = 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 3_8 \quad 111_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 7_8$$

Покажем, как эффективно число в двоичной системе счисления перевести в шестнадцатеричную систему счисления $(2) \rightarrow (16)$.

Возьмем для этого двоичное число 1101010011_2 .

Справа налево отделим по 4 символа: $1101010011_2 = 11 \underline{0101} \underline{0011}_2$.

Если необходимо, слева можно добавить незначащие нули, как в нашем примере:

$$1101010011_2 = 11 \underline{0101} \underline{0011}_2 = \underline{0011} \underline{0101} \underline{0011}_2$$

Затем каждую тетраду независимо друг от друга заменим на шестнадцатеричную цифру:
1101010011₂=11 0101 0011₂=0011 0101 0011₂=3 5 3₁₆

Каждая тетрада переводится обычным способом, например:

0000₂=0₁₆	1000₂=1*2³+0*2²+0*2¹+0*2⁰=8₁₆
0001₂=1₁₆	1001₂=1*2³+0*2²+0*2¹+1*2⁰=9₁₆
0010₂=0*2³+0*2²+1*2¹+0*2⁰=2₁₆	1010₂=1*2³+0*2²+1*2¹+0*2⁰=A₁₆
0011₂=0*2³+0*2²+1*2¹+1*2⁰=3₁₆	1011₂=1*2³+0*2²+1*2¹+1*2⁰=B₁₆
0100₂=0*2³+1*2²+0*2¹+0*2⁰=4₁₆	1100₂=1*2³+1*2²+0*2¹+0*2⁰=C₁₆
0101₂=0*2³+1*2²+0*2¹+1*2⁰=5₁₆	1101₂=1*2³+1*2²+0*2¹+1*2⁰=D₁₆
0110₂=0*2³+1*2²+1*2¹+0*2⁰=6₁₆	1110₂=1*2³+1*2²+1*2¹+0*2⁰=E₁₆
0111₂=0*2³+1*2²+1*2¹+1*2⁰=7₁₆	1111₂=1*2³+1*2²+1*2¹+1*2⁰=F₁₆

Остались два последних способа перевода чисел из одной системы счисления в другую, а именно, из *восьмеричной* системы счисления в *двоичную* систему счисления и из *шестнадцатеричной* системы счисления – в *двоичную* систему счисления.

Покажем, как эффективно число из восьмеричной системы счисления перевести в двоичную систему счисления (8)->(2).

Возьмем для этого восьмеричное число **357₈**.

Используем тот же принцип триад, что и в предыдущих примерах, но в обратном порядке. Для каждой восьмеричной цифры приготовим по 3 знакоместа, и каждую простую восьмеричную цифру переведите в двоичный вид.

$$357_8 = \underline{\underline{3}} \quad \underline{\underline{5}} \quad \underline{\underline{7}} = 011 \ 101 \ 111_2$$

Покажем, как эффективно число из шестнадцатеричной системы счисления перевести в двоичную систему счисления (16)->(2).

Возьмем для этого шестнадцатеричное число **AF7₁₆**.

Используем тот же принцип тетрад, что и в предыдущих примерах, но в обратном порядке. Для каждой шестнадцатеричной цифры приготовим по 4 знакоместа, и каждую простую шестнадцатеричную цифру переведите в двоичный вид.

$$AF7_{16} = \underline{\underline{A}} \quad \underline{\underline{F}} \quad \underline{\underline{7}} = 1010 \ 1111 \ 0111_2$$

Перевод дробных чисел из одной системы счисления в другую.

Известно, что в десятичной системе счисления, числа, представляющие обыкновенную дробь имеют вид p/q , делитель которой q является степенью двойки, пятерки или десятки, можно представить точной десятичной дробью:

$$\frac{1}{2} = 0.5, \quad \frac{3}{4} = 0.75, \quad \frac{5}{16} = 0.0625 * 5 = 0.3125,$$

$$\frac{1}{5} = 0.2, \quad \frac{3}{125} = 3 * 0,008 = 0,024,$$

$$\frac{1}{10} = 0.1, \quad \frac{1}{1000} = 0.001.$$

При этом, если знаменателем дроби являются числа 2 или 5, такую дробь легко перевести в дробь со знаменателем 10 (или десятичную дробь)

$$\frac{1}{2} * \frac{5}{5} = \frac{5}{10} = 0,5$$

$$\frac{1}{4} * \frac{25}{25} = \frac{25}{100} = 0,25$$

$$\frac{1}{5} * \frac{2}{2} = \frac{2}{10} = 0,2$$

$$\frac{1}{25} * \frac{4}{4} = \frac{4}{100} = 0,04$$

Остальные числа представляются либо периодической, либо бесконечной дробью, в которой цифры сгруппировать не получается.

Пример десятичной периодической дроби (где повторяется одна или несколько цифр). В данном случае повторяются цифры **543**.

$$0,0\underline{\underline{543}}\underline{\underline{543}}\underline{\underline{543}}\dots = 0,0(543)$$

В отношении других систем счисления аналогичным образом можно определить правила представления дробной части числа в выбранной системе счисления.

Покажем это на примере двоичной дроби.

Любое число можно рассматривать как сумму целой и дробной части. Как представляются целые числа в двоичной системе счисления понятно, рассмотрим представление дробной части.

Идея представления кода дробной части числа аналогична идее представления целого числа в двоичной системе только по степеням $\frac{1}{2}$, а не 2. Возьмем в качестве примера десятичную дробь **0,625**.

Вспомним десятичные значения степеней $\frac{1}{2}$.

$$\frac{1}{2}: (\frac{1}{2})^1 = 2^{-1} = 0.5$$

$$(\frac{1}{2})^2 = 2^{-2} = 0.25$$

$$(\frac{1}{2})^3 = 2^{-3} = 0.125.$$

Как и для целых чисел, будем находить максимальное число, представляющее степень $\frac{1}{2}$ и меньше или равное остатку дробного числа.

Так как **0,625** больше $0.5 = (\frac{1}{2})^1 = 2^{-1}$, запишем **0,625** следующим образом:

$$0,625 = 1 * (\frac{1}{2})^1 + (0,625 - 0,5) = 1 * (\frac{1}{2})^1 + 0,125$$

Очевидно, 0,125 меньше, чем $0.25 = 2^{-2} = (\frac{1}{2})^2$. Значит,

$$0,625 = 1 * (\frac{1}{2})^1 + (0,625 - 0,5) = 1 * (\frac{1}{2})^1 + 0,125 = 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 0,125.$$

Далее, видим, что $(\frac{1}{2})^3 = 2^{-3} = 0.125$, и это совпадает с остатком рассматриваемого числа, значит, можно записать:

$$0,625 = 1 * (\frac{1}{2})^1 + (0,625 - 0,5) = 1 * (\frac{1}{2})^1 + 0,125 = 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 1 * (\frac{1}{2})^3.$$

То есть можем заключить, что $0,625_{(10)} = 0,101_{(2)}$. И это точная дробь.

Возьмем другое число, например, **0,6425**. Это число делится на 5, но не является степенью ни 5, ни тем более 2. Проведем аналогичные рассуждения.

Так как **0,6425** больше $0.5 = (\frac{1}{2})^1 = 2^{-1}$, запишем **0,6425** следующим образом:

$$0,6425 = 1 * (\frac{1}{2})^1 + (0,6425 - 0,5) = 1 * (\frac{1}{2})^1 + 0,1425.$$

Очевидно, 0,1425 меньше, чем $0.25 = 2^{-2} = (\frac{1}{2})^2$. Значит,

$$0,6425 = 1 * (\frac{1}{2})^1 + (0,6425 - 0,5) = 1 * (\frac{1}{2})^1 + 0,1425 = 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 0,1425.$$

Далее, видим, что $(\frac{1}{2})^3 = 2^{-3} = 0.125$, и это меньше, чем 0,1425, значит, можно продолжить вычисления:

$$\begin{aligned} 0,6425 &= 1 * (\frac{1}{2})^1 + (0,6425 - 0,5) = 1 * (\frac{1}{2})^1 + 0,1425 = \\ &= 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 1 * (\frac{1}{2})^3 + (0,1425 - 0,125) = 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 1 * (\frac{1}{2})^3 + 0,0175. \end{aligned}$$

Очевидно, что $(\frac{1}{2})^4 = 0,0625$ больше, чем 0,0175 и даже $(\frac{1}{2})^5 = 0,03125$ больше, чем 0,0175, поэтому запишем:

$$0,6425 = 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 1 * (\frac{1}{2})^3 + 0 * (\frac{1}{2})^4 + 0 * (\frac{1}{2})^5 + 0,0175.$$

Посмотрим еще $(\frac{1}{2})^6 = 0,0156$, это меньше, чем 0,0175. Значит, можем записать:

$$\begin{aligned} 0,6425 &= 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 1 * (\frac{1}{2})^3 + 0 * (\frac{1}{2})^4 + 0 * (\frac{1}{2})^5 + 1 * (\frac{1}{2})^6 + (0,0175 - 0,0156) = \\ 0,6425 &= 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 1 * (\frac{1}{2})^3 + 0 * (\frac{1}{2})^4 + 0 * (\frac{1}{2})^5 + 1 * (\frac{1}{2})^6 + 0,001875. \end{aligned}$$

Производя дальнейшие вычисления, увидим, что степени 7, 8 и 9 числа $\frac{1}{2}$ все еще меньше остатка рассматриваемого числа:

$$(\frac{1}{2})^7 = 0,0078125, \quad (\frac{1}{2})^8 = 0,00390625, \quad (\frac{1}{2})^9 = 0,001953125.$$

Только $(\frac{1}{2})^{10}$ дает значение **0,0009765625** меньше, чем **0,001875**:

$$\begin{aligned} 0,6425 &= 1 * (\frac{1}{2})^1 + 0 * (\frac{1}{2})^2 + 1 * (\frac{1}{2})^3 + 0 * (\frac{1}{2})^4 + 0 * (\frac{1}{2})^5 + 1 * (\frac{1}{2})^6 + 0 * (\frac{1}{2})^7 + 0 * (\frac{1}{2})^8 + 0 * (\frac{1}{2})^9 + \\ &+ 1 * (\frac{1}{2})^{10} + 0,0009765625 \approx 0,1010010001_{(2)}. \end{aligned}$$

С точки зрения двоичной системы счисления – это бесконечная дробь, так как не кончается точной степенью ($\frac{1}{2}$). То, что это бесконечная двоичная дробь можно определить гораздо проще. Найдем простые делители числа $6425 = 5 * 1285 = 5 * 5 * 257$. Ввиду того, что 257 – простое число, дробь **0,6425** ($0,1010010001_{(2)} \dots$) оказывается бесконечной с точки зрения двоичной системы счисления.

Таким же способом можно представить десятичную дробь в любой системе счисления.

Кстати, целое десятичное число также можно перевести в любую систему счисления ранее рассмотренными способами для целых чисел.

Сложение чисел в системах счисления.

В определении системы кодирования присутствуют 2 фактора:

- алфавит и
- правила образования новых элементов.

Для систем счисления как частного случая системы кодирования правила образования новых элементов – это правила образования новых чисел. Рассмотрим эти правила на примере образования новых чисел в рамках десятичного алфавита $\{0, 1, 2, \dots, 9\}$.

Пусть первое число в десятичной системе – это 0 . Следующее число получится увеличением исходного на 1 , т.е. $0+1=1$. Каждое следующее будем получать аналогично: $1+1=2, 2+1=3, \dots, 8+1=9$. Особенность появится, как только в разряде единиц, получаем 10 единиц, то есть число, равное основанию системы счисления. В этом случае, так как это уже вес следующего разряда, переносим единицу в следующий разряд (слева направо), а в числе единиц ставим 0 : $9+1=10_{10}$.

Рассмотрим теперь двоичную систему счисления. Первое число получается аналогично десятичному: $0+1=1_2$. Особенность появляется на втором шаге: $1+1=2$, но в двоичной системе это невозможно, так как основание двоичной системы есть 2 . В следующий разряд слева направо помещаем единицу, а разряд, где получили 2 единицы, обнуляем: $1+1=10_2$. Согласно основному правилу, получаем следующее число: $10+01=11_2$.

10	11	100	101	110	111	1000	1001	1010	1011	1100
<u>01</u>	<u>.01</u>	<u>001</u>	<u>001</u>	<u>001</u>	<u>001</u>	<u>0001</u>	<u>0001</u>	<u>0001</u>	<u>0001</u>	<u>0001</u>
11	100	101	110	111	1000	1001	1010	1011	1100	1101
3_{10}	4_{10}	5_{10}	6_{10}	7_{10}	8_{10}	9_{10}	10_{10}	11_{10}	12_{10}	13_{10}

Как только в некотором знакоместе появляется 2 единицы, в следующее слева направо знакоместо помещается единица, а в текущее знакоместо помещается 0 .

Посмотрим, как это правило действует для восьмеричной системы. Для первых восьми чисел (для 0_8-7_8) видим, что $0_8=0_{10}, 0_8+1_8=1_8=1_{10}$ восьмеричные числа совпадают с десятичными, а далее смотрите таблицу:

01	02	03	04	05	06	07	10	11	12	13
<u>01</u>										
02	03	04	05	06	10	10	11	12	13	14
2_8	3_8	4_8	5_8	6_8	7_8	10_8	11_8	12_8	13_8	14_8
2_{10}	3_{10}	4_{10}	5_{10}	6_{10}	7_{10}	8_{10}	9_{10}	10_{10}	11_{10}	12_{10}

14	...	17	20	...	27	...	77	...	777	7777
<u>01</u>		<u>01</u>	<u>01</u>		<u>01</u>		<u>01</u>		<u>001</u>	<u>0001</u>
15		20	21		30		100		1000	10000
15_8	...	20_8	21_8	...	30_8	...	100_8	...	1000_8	10000_8
13_{10}	...	16_{10}	17_{10}	...	24_{10}	...	64_{10}	...	512_{10}	4096_{10}

Для восьмеричной системы, как только некоторое знакоместо накапливает 8 единиц, в старший разряд (левый по отношению к текущему) помещается единица, а текущий разряд обнуляется.

Теперь посмотрим, как это правило действует для шестнадцатеричной системы. Для первых десяти шестнадцатеричных чисел (для $0_{16}-9_{16}$) видим, что $0_{16}=0_{10}, 0_{16}+1_{16}=1_{16}=1_{10}$ шестнадцатеричные числа совпадают с десятичными (см. таблицу):

01	02	03	...	09	0A	0B	0C	0D	0E	0F
<u>01</u>	<u>01</u>	<u>01</u>		<u>01</u>						
02	03	04		0A	0B	0C	0D	0E	0F	10
2_{16}	3_{16}	4_{16}	...	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}
2_{10}	3_{10}	4_{10}	...	10_{10}	11_{10}	12_{10}	13_{10}	14_{10}	15_{10}	16_{10}

10	11	...	19	1A	...	1F	20	...	9F	...	FF	100
<u>01</u>	<u>01</u>		<u>01</u>	<u>01</u>		<u>01</u>	<u>01</u>		<u>01</u>		0 1	<u>01</u>
11	12		1A	1B		20	21		A0		100	101
11 ₁₆	12 ₁₆	...	1A ₁₆	1B ₁₆	...	20 ₁₆	21 ₁₆	...	A0 ₁₆	...	100 ₁₆	101 ₁₆
17 ₁₀	18 ₁₀	...	26 ₁₀	27 ₁₀	...	32 ₁₀	33 ₁₀	...	160 ₁₀	...	256 ₁₀	257 ₁₀

Для шестнадцатеричной системы, как только некоторое знакоместо накапливает 16 единиц, в старший разряд (левый по отношению к текущему) помещается единица, а текущий разряд обнуляется (см. таблицу).

Используя это правило сложения, можно выполнять операцию сложения над двумя (и более) числами одной системы счисления. При этом, как обычно, сложение в разрядах производится справа налево), например:

10001101 ₂	36775 ₈	AFFD752 ₁₆	1010101 ₂
<u>110110₂</u>	<u>4775₈</u>	<u>1 35DEF₁₆</u>	11011 ₂
11000011 ₂	43772 ₈	B13 3 541 ₁₆	<u>1011101₂</u>
			11001101 ₂
а	б	в	г

Примеры **а** и **б** более менее понятны, примеры **в** и **г** сложнее.

Рассмотрим пример **в**.

$2+F=2+15_{10}=17_{10}=(16+1)_{10}=11_{16}$	цифра единиц – 1 (нулевой разряд), 1 – перенос в старший разряд (первый разряд)	1 ₁₆
$(5+E+1)_{16}=(5+14+1)_{10}=20_{10}=14_{16}$	цифра 4 - в первом разряде, цифра 1 – перенос во второй разряд	41 ₁₆
$(1+7+D)_{16}=(1+7+13)_{10}=21_{10}=15_{16}$	цифра 5 – во втором разряде, цифра 1 - перенос в третий разряд	541 ₁₆
$(1+D+5)_{16}=(1+13+5)_{10}=19_{10}=13_{16}$	цифра 3 – в третьем разряде, цифра 1 - перенос в четвертый разряд	3541 ₁₆
$(1+F+3)_{16}=(1+15+3)_{10}=19_{10}=13_{16}$	цифра 3 – в четвертом разряде, цифра 1 - перенос в пятый разряд	33541 ₁₆
$(1+F+1)_{16}=(1+15+1)_{10}=17_{10}=11_{16}$	цифра 1 – в пятом разряде, цифра 1 - перенос в шестой разряд	133541 ₁₆
$(1+A)_{16}=(1+10)_{10}=11_{10}=B_{16}$	цифра B в шестом разряде	B133541 ₁₆

Рассмотрим пример **г**.

$(1+1+1)_2=3_2=1+10_2=11_2$	цифра единиц – 1 (нулевой разряд), 1 – перенос в старший разряд (первый разряд)	1 ₂
$(1+0+1+0)_2=2_{10}=10_2$	цифра 0 - в первом разряде, цифра 1 – перенос во второй разряд	01 ₂
$(1+1+0+1)_2=(3)_{10}=11_2$	цифра 1 – во втором разряде, цифра 1 - перенос в третий разряд	101 ₂
$(1+0+1+1)_2=3_2=11_2$	цифра 1 – в третьем разряде, цифра 1 - перенос в четвертый разряд	1101 ₂
$(1+1+1+1)_2=4_{10}=2+10_2=100_2$	цифра 0 – в четвертом разряде, цифра 0 - перенос в пятый разряд, цифра 0 – перенос в шестой разряд	01101 ₂
$(0+10+0)_2=2_{10}=10_2=10_2$	цифра 0 – в пятом разряде, цифра 1 - перенос в шестой разряд	001101 ₂
$(1+1+1)_{16}=3_{10}=11_2$	цифра 1 - в шестом разряде, цифра 1 - перенос в седьмой разряд	11001101 ₂

Задания 1-3 для самостоятельной работы

Задание 1. Выполните следующие задания по теме Системы счисления.

- Возьмите номер Вашей зачетки (или 3 последних цифры логина корпоративной почты *nml*) и переведите в двоичный, восьмеричный и шестнадцатеричный вид:
(10)₁₀→(2)₂, (10)₁₀→(8)₈, (10)₁₀→(16)₁₆.
- Создайте произвольную последовательность **0** и **1** размером не менее 8 символов и переведите в восьмеричную, шестнадцатеричную и десятичную системы счисления
(2)₂→(10)₁₀, (2)₂→(8)₈, (2)₂→(16)₁₆.

3. Возьмите ранее определенное число nml (если цифра n , m или l больше 7, замените её любой допустимой), считая его восьмеричным числом и переведите его в двоичную и десятичную систему счисления: $(8) \rightarrow (2)$, $(8) \rightarrow (10)$;
4. Возьмите ранее определенное число nml (считая его шестнадцатеричным числом) и переведите его в двоичную и десятичную систему счисления: $(16) \rightarrow (10)$, $(16) \rightarrow (2)$;

Задание 2.

1. Попробуйте представить десятичное число $222, nml_{10}$ в двоичной системе счисления.
2. Попробуйте представить десятичное число nml_{10} в семеричной системе счисления.

Задание 3.

1. Сложите три любых двоичных числа не менее 5-ти значных (не переводя в десятичную систему счисления).
2. Сложите три восьмеричных числа не менее 4-х значных (не переводя в десятичную систему счисления).
3. Сложите 2 шестеричных (с основанием счисления 6) числа не менее 4-значных (не переводя в десятичную систему счисления).

Кодирование числовых и текстовых данных

Кодирование числовых данных

Как было выяснено ранее, минимальное количество информации, которое может содержаться в одном знаке для двоичного алфавита, то есть в знаке, в котором одновременно может содержаться знак 0 или 1 , есть бит.

Очевидно, с помощью одного бита можно закодировать самые разные понятия, принимающие какие-либо два значения. Эти два значения можно отобразить абстрактно числовыми значениями: нулем и единицей. Например, обозначить нулем (0) или единицей (1) следующие понятия, соответственно: да или нет, черное или белое, истина или ложь и т.п.

Если количество бит увеличить до двух (то есть использовать два двоичных знака), то можно выразить уже четыре различных понятия (значения):

00 01 10 11 (север, восток, юг, запад).

Аналогично, тремя битами можно закодировать восемь различных значений: **000 001 010 011 100 101 110 111** (например, градации некоторого опроса).

Таким образом, увеличивая на единицу количество бит, можно с каждым разом увеличивать в два раза количество понятий (значений), которое может быть выражено (закодировано) в двоичной системе счисления.

Общая формула вычисления количества информации, которое можно закодировать с помощью нулей и единиц имеет следующий вид:

$$N = 2^m,$$

где N — количество независимых кодируемых значений;

m — количество бит, требуемых для кодирования этой информации.

Иначе эту формулу записывают следующим образом $m = \log_2 N$.

В информатике эти формулы очень востребованы.

Рассмотрим двоичное кодирование различного рода информации:

- чисел (целых и вещественных),
- символов,
- графики,
- аудио и видео информации

и представление этой информации в ЭВМ.

Прежде всего, обратимся к *числовой информации*.

В оперативной памяти ЭВМ целые и вещественные (действительные) числа представляются по-разному, для разных типов чисел и операции, обозначаемые одинаково, выполняются по-разному. Это обязательно учитывается при работе с числовой информацией.

Кроме представленных вариантов представления целого числа в 32 бита, существует и более длинные представления: в 64 бита, то есть 8 байтов и даже 10 байтов.

Для кодирования *вещественных* (действительных) *чисел* используется стандартно 4 байта, то есть 32 бита, но также, как и в случае целых чисел, используются и более длинные представления в 64 бита (8 байтов) и в 80 битов (10 байтов).

Рассмотрим стандартное представление вещественного числа (4 байта).

Если целое число представляется единственным образом, то для вещественного числа в обыденной жизни можно представить бесконечно много представлений. Рассмотрим, например, число **3,14**.

Как его можно представить?

3,14 = 314 * 10⁻² = 0,314 * 10¹ = 0,000314 * 10⁴ = 31,4 * 10⁻¹ и тому подобное.

Для вычислительной машины такая многозначность невозможна. Поэтому для начала решили представления действительных чисел нормировать, то есть договорились представлять действительные числа в виде числовой мантииссы (*m*), модуль которой лежит в границах от **0** до **1** ($0 \leq m < 1$).

Что из нашей последовательности подойдет?

0,314 * 10¹ = 0,000314 * 10⁴ и тому подобное... Но единственности, очевидно, и в этом случае не получили.

Чтобы все же добиться единственности представления вещественного числа, вещественное число решили предварительно преобразовать, нормализовав его, то есть приведя в так называемую *нормализованную форму*.

Нормализованная форма (экспоненциальная форма) вещественного (действительного) числа *a* — это запись следующего вида (рис. 9):

$$a = m * p^q, \quad 0,314 * 10^1$$

где *m* — мантиисса, правильная *p*-ичная дробь, то есть $1/p \leq m < 1$ (основное свойство нормализации состоит в том, что в этом случае у мантииссы первая цифра после запятой не будет равна нулю),

p — основание системы счисления (в машинном представлении, конечно, *p*=2, в нашем демонстрационном варианте *p*=10), в таком случае, *p^q*- степень представления числа,

q — целое число, *порядок (характеристика)* числа.



Рис. 9. Представление вещественного числа в нормализованной форме (4 байта = 32 бита)

Примеры представления нормализованного числа:

$$3,1415926_{10} = 0,31415926 \cdot 10^1$$

$$1250000_{10} = 0,125 \cdot 10^7$$

$$0,123456789_{10} = 0,123456789 \cdot 10^0$$

$$0,000076_{10} = 0,76 \cdot 10^{-4}$$

$$1000,00012_{10} = 0,100000012 \cdot 10^4$$

$$0,10111_2 = 0,10111 * 2^0 \quad \text{Умножение можно представлять знаками * и \cdot}$$

$$0,011_2 = 0,11 * 2^{-1}$$

$$0,0000101_2 = 0,101 * 2^{-4}$$

При желании можно показать представление и комплексных чисел. При этом представление практически не отличается от общепринятого. Отдельно вычисляются реальная часть числа (**Re**) и мнимая часть числа (**Im**) в требуемой системе счисления, а в дальнейшем записывают результат

$$Z = Re(Z) + iIm(Z)$$

В большинстве языков программирования операции над мнимыми числами означают операции над $Re(Z)$ и $Im(Z)$, хотя существуют языки, в которых есть операции непосредственно над комплексными числами.

Кроме нормализованной формы представления действительного числа мы часто видим представление действительного числа в *стандартной форме*. Стандартная форма отличается от нормализованной представлением мантиссы: в нормальной форме мантисса лежит в диапазоне $1/p \leq m < 1$ ($a = m * p^q$, $0,314 * 10^1$), а в стандартной форме — в диапазоне $1/p * 10 \leq m < 10$ ($a = m * p^{q-1}$, $3,14 * 10^0$).

Кодирование текстовых (символьных) данных

Кодирование символьных данных связано с таблицами кодирования, которые существуют изначально со времен появления различных операционных систем. Причем, каждой операционной системе сопоставляется своя кодовая таблица (мы знаем CP Windows – 1251, DOS CP-768, и прочее).

При кодировании символьных данных каждому символу кодовой таблицы, содержащей алфавит, цифры, разделители, знаки операций и еще многие другие символы, сопоставляется целое число, представленное в двоичном кодировании. До 2000 года все кодовые таблицы (для DOS, Windows, Mac, Linux) представляли один символ одним байтом. Легко вычислить по формуле $N=2^m$, что восьми бит одного байта достаточно для кодирования 256 различных символов. Этого хватает для выражения всех символов английского и русского языков, как строчных, так и прописных, а также знаков препинания, символов основных математических операций и некоторых других специальных символов. Впоследствии была введена кодовая таблица Unicode, которая кодирует один символ двумя байтами, что позволяет иметь практически 256 таблиц кодирования.

Все было бы технически просто, если бы в разных кодовых таблицах была бы хоть какая-то универсальность. Однако, изначально, каждая фирма, создавая свое системное программное обеспечение, пыталась привязать к себе клиента, как можно жестче, отсюда такое изобилие действующих стандартов.

Таблица 1. Коды ASCII. Кодировка символов, предложенная IBM

		младший полубайт															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
старший полубайт	0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0000	0	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
0001	1	␣	␣	!	␣	␣	␣	␣	␣	↑	↓	→	←	↔	␣	␣	␣
0010	2	␣	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0011	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	4	␣	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0110	6	␣	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	␣
1000	8	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
1001	9	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
1010	A	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
1011	B	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
1100	C	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
1101	D	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
1110	E	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
1111	F	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣

Чтобы снять многие организационные и технические несоответствия, в конце прошлого века Институт стандартизации США ввел в действие систему кодирования ASCII (American Standard Code for Information Interchange — стандартный код информационного обмена США). Тем самым для представления символов английского языка противоречия были сняты.

Система ASCII (1 байт - 1 символ) разделяет таблицу кодирования на три части:

- ♦ коды с 0_{10} (00_{16}) по 31_{10} ($1F_{16}$) — управляющие и непечатаемые символы (ESC, Enter, стрелочки, табуляция и тому подобное), с их помощью можно управлять аппаратурой,
- ♦ коды с 32_{10} (20_{16}) по 127_{10} ($7F_{16}$) — символы латинского алфавита (блоками прописные и строчные, цифры, знаки препинания и основных операций и тому подобное), (01111111) 127_2
- ♦ коды с 128_{10} (10_{16}) по 255_{10} (FF_{16}) — национальный алфавит и дополнительные спецсимволы

Рассмотрите символы, входящие в базовую (рис. 10) и расширенную (рис. 11) части стандарта ASCII.

		младший полубайт																	
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111		
старший полубайт	Базовая часть	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
		0010	2	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
		0011	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
		1100	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
		0101	5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
		0110	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
		0111	7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
				112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

Рис. 10. Базовая часть ASCII: латинский алфавит, цифры и наиболее употребительные спецсимволы

		младший полубайт																			
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111				
старший полубайт	Расширенная часть	8	9	A	B	C	D	E	F												
		1000	8	ç	ú	ë	á	ä	å	ç	è	é	ê	ë	ì	í	î	ï	À	Á	
		1001	9	è	é	ê	ë	ü	û	ü	ÿ	ö	Û	ç	£	¥	ℳ	ƒ			
		1010	A	í	ó	ú	ä	å	ª	º	¿	¬	½	¾	¿	«	»				
		1011	B	☼	☽	☾															
		1100	C	┌	┐	└	┘	├	┤	├	┤	├	┤	├	┤	├	┤	├	┤	├	┤
		1101	D	┌	┐	└	┘	├	┤	├	┤	├	┤	├	┤	├	┤	├	┤	├	┤
		1110	E	α	β	Γ	χ	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	π		
1111	F	≡	±	≥	≤		∫	÷	∞	°	·	√	∩	∪	∩	∪	∩	∪	∩		
		240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255				

Рис. 11. Расширенная часть ASCII: национальный алфавит, греческий алфавит и дополнительные спецсимволы

В таблице 1 и на рисунках 10 и 11 коды символов в ячейках таблицы представлены в десятичной системе счисления. Тем не менее, коды символов представляют в шестнадцатеричном виде следующим образом:

- ♦ по горизонтали в качестве заголовков столбцов откладывают шестнадцатеричные символы, представляющие младший полубайт кода от **0** до **F** (последние 4 бита),
- ♦ по вертикали в качестве заголовков строк откладывают шестнадцатеричные символы, представляющие старший полубайт кода от **0** до **F** (левые 4 бита).

		младший полубайт																
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
Расширенная часть	старший полубайт	1000	В	Ь	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	
			128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
	1001	9	ђ	'	'	'	'	'	'	'	'	'	'	'	'	'	'	'
			144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
	1010	A		У	У	У	У	У	У	У	У	У	У	У	У	У	У	У
			160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
	1011	B	*	±	l	i	l	μ			ё	№	e	»	j	S	s	i
			176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
1100	C	A	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	
		192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	
1101	D	P	C	Г	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	
		208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	
1110	E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	
		224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	
1111	F	р	с	г	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я	
		240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	

Рис. 12. Расширенная часть таблицы Windows-1251 с национальным алфавитом

Обратите внимание, что шестнадцатеричному изображению полубайта сопоставлено его двоичное изображение (слева или сверху).

Разнообразие же стандартов кодировок обусловлено тем, что для каждого национального языка существует свой стандарт кодировки. К тому же и для одного языка может быть создано несколько кодировок, например, кодировка символов русского языка в операционной системе Windows, известная как кодировка *Windows-1251* (рис. 12), была введена компанией *Microsoft*. В дальнейшем, учитывая распространенность продуктов этой фирмы в нашей стране, эта кодировка закрепились и стала широко использоваться.

Другая распространенная кодировка символов русского языка (прежде всего, используемая в Linux) носит название КОИ-8 (код обмена информацией, восьмизначный, рис. 13).

128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	nbsp	Ј	•	²	•	÷
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Рис. 13. Расширенная часть кодовой таблицы КОИ-8

В КОИ-8 порядок русских букв кажется случайным, однако это не так.

- Русские буквы поставлены в четкое соответствие с латинскими. Если убрать 8 бит, строчные русские буквы превращаются в заглавные латинские, а заглавные русские – в строчные латинские.
- КОИ-8 позволяет сохранить читаемость русского текста при потере 8 бита. “Привет всем!” превращается в “pRIWET WSEM”.

На рисунке 14 представлена расширенная часть кодовой таблицы MAC, на рисунке 15 — расширенная часть кодовой таблицы ISO.

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
ƒ	°	Ы	£	§	•	¶	Ы	©	©	™	Á	á	è	à	è
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
Ë	±	≤	≥	¿	µ	г	É	Ю	ю	Я	я	É	é	Й	ò
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
Э	Ю	¬	√	f	≈	Δ	«	»	...	nbsp	Ó	ó	Й	й	я
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
–	—	“	”	‘	’	÷	„	ÿ	Ы	У	У	№	Ë	ë	я
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Рис. 14. Расширенная часть кодовой таблицы MAC

І	І	І	І	І	І	І	І	І	І	І	І	І	І	І	І
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
І	І	І	І	І	І	І	І	І	І	І	І	І	І	І	І
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
nbsp	Ë	Ъ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	shy	Ў	Џ
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
№	ë	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ	ѕ	ў	џ
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Рис. 15. Расширенная часть кодовой таблицы ISO

Основные недостатки восьмибитных кодировок (1 байт – 1 символ):

- множество вариантов кодировок для одного и того же языка;
- невозможность использования в одном тексте разных систем письма (за исключением сочетаний базового латинского алфавита с национальным алфавитным письмом);
- невозможность использования для языков с иероглифической системой письма.

Как уже упоминалось, для устранения этих недостатков в 1991 году появилась система кодирования **UNICODE**, основанная на шестнадцатитбитном кодировании символов (один символ - два байта). Она позволяет обеспечить уникальные коды для 65 536 различных символов.

Система включает символы всех современных и многих древних систем письма, например, включает алфавиты типа японского kana, корейского hangul и китайского wotofo. Самая большая часть посвящена тысячам объединенных знаков для японских, корейских и китайских иероглифов.

Объем текстовой информации определяется следующим образом: подсчитывается общее количество символов в тексте, учитывая пробелы, знаки препинания, цифры и другие знаки, присутствующие в используемой кодовой таблице и добавляется непечатный символ конца текста:

$$V_{\text{Текста}} = K_{\text{символов}} + 1.$$

С помощью таблиц кодов можно также решать задачи кодирования.

Попробуем закодировать какое-нибудь слово. Будем предполагать, что используется система кодирования *ASCII*, а для национального блока (расширенной таблицы) таблицы кодов используется кодировка *Windows-1251*. Ответ запишем в шестнадцатеричной системе счисления.

Пример:

Дано: **КОТ** (слово на русском языке, написанное заглавными буквами).

Найти: соответствующий код цепочки символов **ASCII₁₆**?

Решение:

Найдем код буквы **К** (данный в десятичном представлении) в шестнадцатеричном представлении:

$$K = 202_{10}$$

$$202 : 16 \mid 10 \uparrow$$

12 $\xrightarrow{\hspace{2cm}}$ $12_{10}10_{10} = CA_{16}$

Учтем, что $10_{10} = A_{16}$,

$$11_{10} = B_{16},$$

$$12_{10} = C_{16},$$

$$13_{10} = D_{16},$$

$$14_{10} = E_{16},$$

$$15_{10} = F_{16},$$

тогда $K = 202_{10} = CA_{16}$

Выполним аналогичную операцию для буквы **О**:

$$O = 206_{10}$$

$$206 : 16 \mid 14 \uparrow$$

12 $\xrightarrow{\hspace{2cm}}$ $12_{10}14_{10} = CE_{16}$

$$O = 206_{10} = CE_{16}$$

$$T = 210_{10}$$

$$210 : 16 \mid 2 \uparrow$$

13 $\xrightarrow{\hspace{2cm}}$ $13_{10}2_{10} = D2_{16}$

$$T = 210_{10} = D2_{16}$$

Таким образом, **КОТ₁₀** в шестнадцатеричном представлении будет выглядеть следующим образом:

Ответ: КОТ = CA CE D2₁₆

Теперь решим обратную задачу. Зная код в системе кодирования **ASCII** (*Windows-1251*), записанный в шестнадцатеричной системе счисления, найдем закодированное слово.

Пример:

Дано: **CF A8 D1₁₆**

Найти: слово, соответствующее коду.

Решение:

Вспомним, что каждый символ таблицы кодов представлен двумя шестнадцатеричными цифрами, то есть имеем три символа: **CF₁₆ A8₁₆ D1₁₆**.

Рассмотрим каждый символ, переводя из шестнадцатеричной системы в десятичную для таблицы кодов *Windows-1251*:

$$CF_{16} = C \cdot 16^1 + F \cdot 16^0 = 12 \cdot 16 + 15 \cdot 1 = 192 + 15 = 207_{10} \quad 207_{10} = П_{16}$$

$$A8_{16} = A \cdot 16^1 + 8 \cdot 16^0 = 10 \cdot 16 + 8 \cdot 1 = 160 + 8 = 168_{10} \quad 168_{10} = Ё_{16}$$

$$D1_{16} = D \cdot 16^1 + 1 \cdot 16^0 = 13 \cdot 16 + 1 \cdot 1 = 208 + 1 = 209_{10} \quad 209_{10} = С_{16}$$

Ответ: CF A8 D1₁₆ = ПЁС

Задание 4 (по системам счисления и кодированию).

Перекодируйте слово в коды таблицы кодов **ASCII**:

Cat = ASCII₁₆?

Dog = ASCII₁₆?

Получите слово, записанное в кодах *Windows-1251* из цепочки шестнадцатеричных кодов:

C4 E0₁₆ = слово?

CD E5 F2₁₆ = слово?

Кодирование графических изображений

Растровое кодирование

Рассмотрим, как осуществляется кодирование графических изображений. Схематично рисунок состоит из линий и закрашенных областей. Значит, в идеале нужно закодировать все особенности этого изображения так, чтобы его можно было бы в точности *восстановить* из кода (например, распечатать на принтере).

Из геометрии мы знаем, что и линия, и область состоят из бесконечного числа точек. Каждая точка имеет цвет, который тоже нужно закодировать. Но так, как точек *бесконечно* много, следует вывод, что для их кодирования требуется бесконечно много памяти. Поэтому «поточечным» способом изображение закодировать не удастся. Однако идею «поточечного» кодирования всё-таки можно использовать.

Рассмотрим чёрно-белый рисунок ромба. Представим себе, что на изображение ромба наложена сетка, которая разбивает его на квадратики. Такая сетка называется *растром*. Определим для каждого квадратика ромба на сетке чёрный или белый цвет. Договоримся, что для тех квадратиков, в которых большая часть квадратика оказалась закрашена чёрным цветом, выбирается чёрный цвет, для тех квадратиков, в которых большая часть квадратика оказалась закрашена белым цветом, выбирается белый цвет (рис. 16).

В результате этих последовательных действий получим, так называемый, растровый рисунок, состоящий из квадратиков-пикселей.

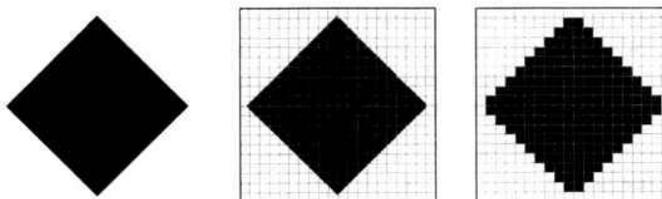


Рис. 16. Получение растра изображения

Пиксель (англ. *pixel* — *pictures element*, элемент рисунка) — это наименьший элемент рисунка, для которого можно независимым образом задать свой цвет.

Наложив на рисунок сетку и, тем самым, разбив «обычный» рисунок на квадратики, выполнили операцию *дискретизации*, то есть *разделили единый объект на отдельные равномерно отстоящие элементы*. Таким образом, из единого рисунка (ромба) получили *дискретный объект* — набор пикселей.

Теперь применим к чёрно-белому рисунку, полученному в результате дискретизации, алгоритм двоичного кодирования, который состоит в следующем:

- заменить в каждой строке рисунка белые пиксели нулями, а чёрные — единицами;
- выписать числовые значения каждой строки в таблицу одну за другой (рис. 17).

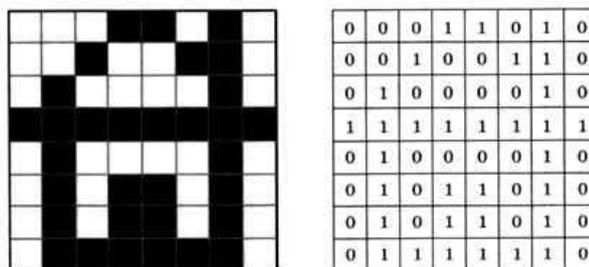


Рис. 17. Применение алгоритма двоичного кодирования к графическому изображению

Ширина рисунка, предложенного для демонстрации алгоритма двоичного кодирования графики — 8 пикселей, поэтому каждая строка таблицы содержит 8 двоичных разрядов (битов). Чтобы не путаться в написании очень длинной цепочки нулей и единиц, удобно использовать шестнадцатеричную систему счисления, кодируя по 4 соседних бита (тетраду) одной шестнадцатеричной цифрой. Например, для первой строки получим код **00011010**=**1A**₁₆,

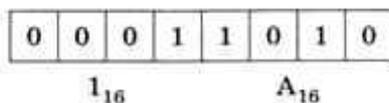


Рис. 18. Изображение первой строки кодировки

а для всего рисунка 18 получим строку **1A2642FF425A5A7E**₁₆.

В результате дискретизации (выбрав определенного размера ячейки сетки) получили представление графического изображения в цифровом (двоичном коде) виде. Однако, при этом рисунок несколько исказился: вместо ромба был получен набор квадратиков. Причина искажения в том, что в некоторых квадратиках части исходного рисунка были закрашены разными цветами, а в закодированном изображении каждый пиксель, согласно принятому алгоритму, обязательно имеет один цвет. Таким образом, часть исходной информации при кодировании оказалась потерянной. Это наглядно проявится, например, при увеличении рисунка — квадратик увеличатся, и рисунок ещё больше исказится. Чтобы уменьшить потери информации, нужно уменьшать размер пикселя, то есть увеличивать разрешение (размер дискретизации).

Разрешение — это количество пикселей, приходящихся на единицу линейного размера изображения.

Разрешение обычно измеряется в пикселях на дюйм. Используется английское обозначение *ppi* (*pixels per inch*).

Например, разрешение 254 ppi означает, что на дюйм (25,4 мм) приходится 254 пикселя, так что каждый пиксель представляет квадратик исходного изображения размером 0,1 x 0,1 мм². Если провести дискретизацию рисунка размером 10 см x 15 см с разрешением 254 ppi, высота закодированного изображения будет 100/0,1 = 1000 пикселей, а ширина — 1500 пикселей.

Чем больше разрешение, тем точнее кодируется рисунок (меньше информации теряется), однако одновременно растёт и объём файла.

Из приведенного объяснения понятно, что объём файла в битах складывается из количества пикселей (точек по ширине умноженных на точки по длине), умноженного на количество битов, требуемых для кодирования цвета каждой точки:

$$V_{\text{Изображения}} = K_{\text{Точек}} * I, \text{ где } I - \text{глубина цвета в битах.}$$

00 01 10 11 четыре возможности для кодирования цвета

000 001 010 011 100 101 110 111 восемь возможностей для кодирования цвета

Глубина цвета — это количество битов, используемое для кодирования цвета пикселя, иначе степень 2.

В случае **00 01 10 11** - глубина цвета - 2,

в случае **000 001 010 011 100 101 110 111** – глубина цвета 3

Кодирование цвета

Что делать, если рисунок цветной? В этом случае для кодирования цвета пикселя уже не обойтись одним битом.

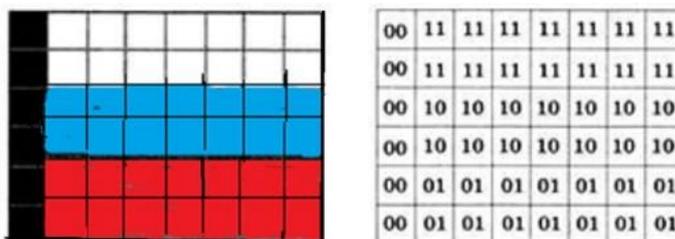


Рис. 19. Кодирование рисунка

Например, в показанном на рис. 19, изображении российского флага 4 цвета: чёрный, синий, красный и белый. Для кодирования одного из четырёх вариантов нужно 2 бита, поэтому код каждого цвета (и код каждого пикселя) будет состоять из двух битов.

Пусть **00** обозначает чёрный цвет, **01** — красный, **10** — синий и **11** — белый. Получаем таблицу справа на рисунке 19.

Проблема только в том, что при выводе на экран нужно как-то определить, какой цвет соответствует тому или другому коду. То есть информацию о цвете для вывода на экран нужно выразить в виде числа (или набора чисел).

Человек воспринимает свет как множество электромагнитных волн. Определенная длина волны соответствует некоторому цвету. Например, волны длиной 500-565 нм — это зелёный цвет. Так называемый «белый» свет на самом деле представляет собой смесь волн, длины которых охватывают весь видимый диапазон.

Согласно современному представлению о цветном зрении (теории Юнга—Гельмгольца), глаз человека содержит чувствительные элементы (рецепторы) трёх типов. Каждый из них воспринимает весь поток света, но первые наиболее чувствительны в области красного цвета, вторые — в области зелёного цвета, а третьи — в области синего цвета. Цвет — это результат возбуждения всех трёх типов рецепторов. Поэтому считается, что любой цвет (то есть ощущения человека, воспринимающего волны определённой длины) можно имитировать, используя только три световых луча (красный, зелёный и синий) разной яркости. Следовательно, любой цвет (в том числе и «белый») приближённо раскладывается на три составляющих — красную, зелёную и синюю. Меняя силу этих составляющих, можно составить любые цвета. Эта модель цвета получила название RGB по начальным буквам английских слов «*red*» (красный), «*green*» (зелёный) и «*blue*» (синий).

В модели **RGB** яркость каждой составляющей (или, как говорят, каждого канала) чаще всего кодируется целым числом от 0 до 255. При этом код цвета — это тройка чисел (**R, G, B**) — яркости отдельных каналов. Цвет (0, 0, 0) — это чёрный цвет, а (255, 255, 255) — белый. Если все составляющие имеют равную яркость, получаются оттенки серого цвета: от чёрного до белого.

Чтобы сделать светло-красный (розовый) цвет, нужно при максимальной яркости красного цвета (255, 0, 0) одинаково увеличить яркость зелёного и синего каналов, например, цвет (255, 150, 150) — это розовый. Равномерное уменьшение яркости всех каналов создаёт тёмный цвет, например, цвет с кодом (100, 0, 0) — тёмно-красный.

При кодировании цвета на веб-страницах также используется модель **RGB**, но яркости каналов записываются в шестнадцатеричной системе счисления (от 00_{16} до FF_{16}), а перед кодом цвета ставится знак #. Например, код красного цвета записывается как #FF0000, а код синего — как #0000FF. Коды некоторых цветов приведены в таблице основных цветов в системе RGB ниже.

Ввиду того, что для каждого из цветов на яркость отводится по 1 байту, получим по 256 вариантов яркости каждого из трёх основных цветов. Это позволяет закодировать $256^3 = 256 \cdot 256 \cdot 256 = 2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16\,777\,216$ оттенков, что более, чем достаточно для человека. Так как $256 = 2^8$, каждая из трёх составляющих занимает в памяти 8 битов, или 1 байт, а вся информация о каком-то цвете — 24 бита (3 байта).

Таблица 2. Основные цвета в системе RGB

Цвет	Код (R, G, B)	Код на веб-странице
Красный	(255, 0, 0)	#FF0000
Зеленый	(0, 255, 0)	#00FF00
Синий	(0, 0, 255)	#0000FF
Белый	(255, 255, 255)	#FFFFFF
Чёрный	(0, 0, 0)	#000000
Серый	(128, 128, 128)	#808080
Пурпурный ¹	(255, 0, 255)	#FF00FF
Голубой	(0, 255, 255)	#00FFFF
Желтый	(255, 255, 0)	#FFFF00
Темно-пурпурный	(128, 0, 128)	#800080
Светло-желтый	(255, 255, 128)	#FFF800

¹ Пурпурный цвет получается при смешении синего и красного.

Величина, которая определяется количеством битов, требующихся для кодирования информации о цвете, называется *глубиной цвета*.

Так, например, в известной формуле $N=2^m$, за N обозначим количество цветов, тогда m — это глубина цвета, тогда при **16 777 216** цветов требуется 3 байта, то есть 24 бита:

$$16\ 777\ 216=2^{24}, \text{ то есть } m=24, \text{ где } m \text{ — глубина цвета.}$$

Глубина цвета — это количество битов, используемое для кодирования цвета пикселя.

24-битовое кодирование цвета часто называют режимом *истинного* цвета (англ. *True Color* — истинный цвет). Для вычисления объёма рисунка в байтах при таком кодировании нужно определить общее количество пикселей (перемножить ширину и высоту) и умножить результат на 3, так как цвет каждого пикселя кодируется тремя байтами. Например, рисунок размером 20 x 30 пикселей, закодированный в режиме истинного цвета, будет занимать

$$20 \cdot 30 \cdot 24(\text{битов}) = 14400 \text{ (битов)} \quad 14400/8 = 1800 \text{ (байтов)}$$

$20 \cdot 30 \cdot 3 = 1800$ байтов. Конечно, здесь не учитывается сжатие (уменьшение объёма файлов с помощью специальных алгоритмов), которое применяется во всех современных форматах графических файлов. Кроме того, в реальных файлах есть заголовок, в котором записана служебная информация, например, размеры рисунка).

Кроме режима истинного цвета используется также 16-битное кодирование (англ. *High Color* — «высокий» цвет), когда на красную и синюю составляющие отводится по 5 битов, а на зелёную, к которой человеческий глаз более чувствителен, — 6 битов. В режиме *High Color* можно закодировать $2^{16} = 65\ 536$ различных цветов. В мобильных телефонах иногда применяют 12-битное кодирование цвета (4 бита на канал, **4096** цветов).

Как правило, чем меньше цветов используется, тем больше будет искажаться цветное изображение. Таким образом, при кодировании цвета тоже есть неизбежная потеря информации, которая «добавляется» к потерям, вызванным дискретизацией. Однако, при увеличении количества используемых цветов растёт объём файла. Например, в режиме *истинного* цвета файл получится в два раза больше, чем при 12-битном кодировании.

Очень часто (например, в схемах, диаграммах и чертежах) количество цветов в изображении невелико (не более 256, то есть 1 байт). В этом случае применяют *кодирование с палитрой*.

Цветовая палитра — это таблица, в которой каждому цвету, заданному в виде составляющих в модели RGB, сопоставляется числовой код.

Кодирование с палитрой выполняется следующим образом:

- выбирается количество цветов N для конкретного случая (как правило, не более 256, допустим 4);
- из палитры истинного цвета (**16 777 216** цветов) выбираются любые N цветов (наши 4) и для каждого из них находятся составляющие в модели **RGB** (**FFFFFF** (255, 255, 255), **FF0000** (255, 0, 0), **00FF00** (0, 255, 0), **0000FF** (0, 0, 255));
- каждому из выбранных цветов присваивается номер (код) от **0** до $N - 1$ (**112-310 – FFFFFFFF, 002-010-FF0000, 012-110- 00FF00, 102-210-0000FF**);
 - ♦ составляется палитра: сначала записываются RGB-составляющие цвета, имеющего код **0**, затем — составляющие цвета с кодом **1** и т. д.;
 - ♦ цвет каждого пикселя кодируется не в виде значений RGB-составляющих, а как *номер* цвета в палитре.

Например, при кодировании изображения российского флага (рисунок 15 выше) были выбраны 4 цвета:

- цвет **010** - чёрный: RGB-код (**0, 0, 0**): двоичный код 00₂;
- цвет **110** - красный: RGB-код (**255, 0, 0**): двоичный код 01₂;
- цвет **210** - синий: RGB-код (**0, 0, 255**): двоичный код 10₂;
- цвет **310** - белый: RGB-код (**255, 255, 255**): двоичный код 11₂;

Поэтому палитра, которая обычно записывается в специальную служебную область в начале файла (эту область называют *заголовком файла*), представляет собой четыре трёхбайтных блока:

0	0	0	255	0	0	0	0	255	255	255	255
цвет 0=00 ₂			цвет 1=01 ₂			цвет 2=10 ₂			цвет 3=11 ₂		

Код каждого пикселя занимает всего два бита.

Чтобы примерно оценить информационный объём рисунка с палитрой, включающей N (4 цвета) цветов, нужно:

- определить размер палитры: $3 \cdot N$ (4) байтов, или $24 \cdot N$ битов;
- определить *глубину цвета* (количество битов на пиксель), т. е. найти наименьшее натуральное число k , такое что $2^k \geq N$ ($N=4$, $k=2=2_2$, $N=2^k$);
- вычислить общее количество пикселей M , перемножив размеры рисунка ($20 \times 30 = 600$ пикселей (точек));
- определить информационный объём рисунка (без учёта палитры): $M \cdot k$ битов ($600 \times 2 = 1200$ битов);
- добавить к полученному на предыдущем шаге размер заголовка: 3 байта или 24 бита ($1200 + 24 = 1224$ бита $= 1224 / 8 = 153$ байта).

В таблице ниже приведены данные по некоторым вариантам кодирования с палитрой.

Таблица 3. Варианты кодирования с палитрой

Количество цветов	Размер палитры, байтов	Глубина цвета, битов на пиксель
2	6	1
4	12	2
16	48	4
256	768	8
216	648	8

Палитры с количеством цветов более 256 на практике не используются.

RGB-кодирование лучше всего описывает цвет, который излучается некоторым устройством, например, экраном монитора или ноутбука (рис. 20, а). Когда смотрят на изображение, отпечатанное на бумаге, ситуация совершенно другая. Человек видит не прямые лучи источника, попадающие в глаз, а отражённые от поверхности. «Белый свет» от какого-то источника (солнца, лампочки), содержащий волны во всём видимом диапазоне, попадает на бумагу, на которой нанесена краска. Краска поглощает часть лучей (их энергия уходит на нагрев), а оставшиеся попадают в глаз, это и есть тот цвет, который мы видим (рис. 20, б).

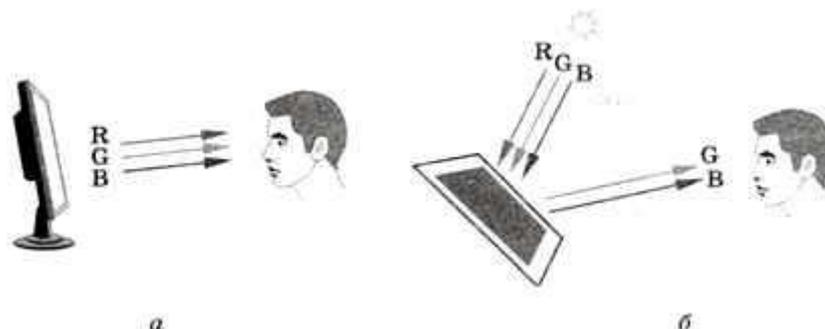


Рис. 20. Отражение света от монитора и от изображения

Например, если краска поглощает красные лучи, остаются только синие и зелёные (рис. 20, б) — видят голубой цвет.

В этом смысле красный (#FF0000) и голубой (#00FFFF) цвета дополняют друг друга также, как и пары: зелёный (#00FF00) — пурпурный (#FF00FF) и синий (#0000FF) — жёлтый (#FFFF00).

Действительно, если из белого цвета (его RGB-код #FFFFFF) «вычесть» зелёный (#00FF00), то получится пурпурный цвет (#FF00FF), а если «вычесть» синий (#0000FF), то получится жёлтый цвет (#FFFF00).

На трёх *дополнительных* цветах — голубом, пурпурном и жёлтом — строится цветовая модель **СМУ** (англ. *Cyan* — голубой, *Magenta* — пурпурный, *Yellow* — жёлтый), которая применяется для вывода изображения на печать. Значения $C = M = Y = 0$ говорят о том, что на белую бумагу не наносится никакая краска, поэтому все лучи отражаются, виден белый цвет. Если нанести на бумагу голубой цвет, красные лучи будут поглощаться, останутся только синие и зелёные. Если сверху нанести ещё жёлтую краску, которая поглощает синие лучи, останется только зелёный цвет.

При наложении голубой, пурпурной и жёлтой красок теоретически должен получиться чёрный цвет, — все лучи поглощаются. Однако на практике всё не так просто. Краски не идеальны, поэтому вместо чёрного цвета получается грязно-коричневый. Кроме того, при печати черных областей приходится «выливать» тройную порцию краски в одно место. Нужно также учитывать, что обычно на принтерах часто распечатывают чёрный текст, а цветные чернила значительно дороже чёрных.

Чтобы решить эту проблему, в набор красок добавляют чёрную краску, это так называемый ключевой цвет (англ. *Key color*), поэтому получившуюся модель обозначают **СМУК**. Изображение, которое печатает большинство принтеров, состоит из точек этих четырёх цветов, которые расположены в виде узора очень близко друг к другу. Это создаёт иллюзию того, что в рисунке есть разные цвета.

Кроме цветовых моделей RGB и CMY (СМУК) существуют и другие модели. Наиболее интересная из них — модель **HSB**¹ (англ. *Hue* — тон, оттенок; *Saturation* — насыщенность, *Brightness* — яркость), которая ближе всего к естественному восприятию человека. *Тон* — это основа цвета, например: синий, зелёный, жёлтый.

Насыщенность — это чистота тона, при уменьшении насыщенности до нуля получается серый цвет. *Яркость* определяет, насколько цвет светлый или тёмный. Любой цвет при снижении яркости до нуля превращается в чёрный.

Строго говоря, цвет, кодируемый в моделях RGB, СМУК и HSB, зависит от устройства, на котором этот цвет будет изображаться. Для кодирования «абсолютного» цвета применяют модель **Lab** (англ. *Lightness* — светлота, *a* и *b* — параметры, определяющие тон и насыщенность цвета), которая является международным стандартом. Эта модель используется, например, для перевода цвета из модели **RGB** в модель **СМУК** и обратно.

Обычно изображения, предназначенные для печати, готовятся на компьютере (в режиме **RGB**), а потом переводятся в цветовую модель **СМУК**. При этом стоит задача — получить при печати такой же цвет, что и на мониторе. И вот тут возникают проблемы. Дело в том, что при выводе пикселей на экран монитор получает некоторые числа (RGB-коды), на основании которых нужно «выкрасить» пиксели тем или иным цветом. Отсюда следует важный вывод.

Цвет, который человек видит на мониторе, зависит от характеристик и настроек монитора.

Это значит, что, например, красный цвет ($R = 255, G = B = 0$) на разных мониторах будет разным. Наверняка вы видели этот эффект в магазине, где продают телевизоры и мониторы, — одна и та же картинка на каждом из них выглядит по-разному. Что же делать?

Во-первых, выполняется *калибровка* монитора — настройка яркости, контрастности, белого, чёрного и серого цветов. Во-вторых, профессионалы, работающие с цветными изображениями, используют *цветовые профили* мониторов, сканеров, принтеров и других устройств. В профилях хранится информация о том, каким реальным цветам соответствуют

¹ **HSB** иногда называют **HSV** (англ. *Hue* — тон, оттенок; *Saturation* — насыщенность, *Value* — величина).

различные RGB-коды или CMYK-коды. Для создания профиля используют специальные приборы — калибраторы (колориметры), которые «измеряют» цвет с помощью трёх датчиков, принимающих лучи в красном, зелёном и синем диапазонах. Современные форматы графических файлов (например, формат PSD программы Adobe Photoshop) вместе с кодами пикселей хранят и профиль монитора, на котором создавался рисунок.

Для того чтобы результат печати на принтере был максимально похож на изображение на мониторе, нужно (используя профиль монитора) определить «абсолютный» цвет (например, в модели *Lab*), который видел пользователь, а потом (используя профиль принтера) найти CMYK-код, который даст при печати наиболее близкий цвет.

Проблема состоит в том, что не все цвета RGB-модели могут быть напечатаны. В первую очередь это относится к ярким и насыщенным цветам.

Например, ярко-красный цвет ($R = 255, G = B = 0$) нельзя напечатать, ближайший к нему цвет в модели CMYK ($C = 0, M = Y = 255, K = 0$) при обратном переводе в RGB может дать значения² в районе $R = 237, G = 28, B = 26$. Поэтому при преобразовании ярких цветов в модель CMYK (и при печати ярких рисунков) они становятся тусклее. Это обязательно должны учитывать профессиональные дизайнеры.

Итак, при растровом кодировании рисунок разбивается на пиксели (дискретизируется). Для каждого пикселя определяется единый цвет, который чаще всего кодируется с помощью RGB-кода. На практике эти операции выполняет сканер (устройство для ввода изображений) или цифровой фотоаппарат.

Растровое кодирование имеет

достоинства:

- универсальный метод (можно закодировать любое изображение);
- единственный метод для кодирования и обработки размытых изображений, не имеющих чётких границ, например, фотографий;

и недостатки:

- при дискретизации всегда есть потеря информации;
- при изменении размеров изображения и повороте искажается цвет и форма объектов на рисунке, поскольку при увеличении размеров надо как-то восстановить недостающие пиксели, а при уменьшении — заменить несколько пикселей одним;
- размер файла не зависит от сложности изображения, а определяется только разрешением и глубиной цвета; как правило, растровые рисунки имеют большой объём.

Существует много разных форматов хранения растровых рисунков. В большинстве из них используют сжатие, то есть уменьшают размер файла с помощью специальных алгоритмов. В некоторых форматах применяют сжатие без потерь, при котором исходный рисунок можно в точности восстановить из сжатого состояния. Ещё большую степень сжатия можно обеспечить, используя сжатие с потерями, при котором незначительная часть данных (почти не влияющая на восприятие рисунка человеком) теряется.

Чаще всего встречаются следующие форматы файлов:

- **ВМР** (англ. *bitmap* — битовая карта; файлы с расширением *bmp*) — стандартный формат растровых изображений в операционной системе Windows; поддерживает кодирование с палитрой и в режиме истинного цвета;
- **JPEG** (англ. *Joint Photographic Experts Group* — объединённая группа фотографов-экспертов; файлы с расширением *jpg* или *jpeg*) — формат, разработанный специально для кодирования фотографий; поддерживает только режим *истинного цвета*; для уменьшения объёма файла используется *сильное сжатие*, при котором изображение немного искажается, поэтому не рекомендуется использовать его для рисунков с чёткими границами;

² Точные цифры зависят от профилей монитора и принтера.

- **GIF** (англ. *Graphics Interchange Format* — формат для обмена изображениями; файлы с расширением *gif*) — формат, поддерживающий только кодирование с палитрой (от 2 до 256 цветов); в отличие от предыдущих форматов части рисунка могут быть прозрачными, то есть на веб-странице через них будет «просвечивать» фон; в современном варианте формата GIF можно хранить анимированные изображения; используется сжатие без потерь, то есть при сжатии изображение не искажается;
- **PNG** (англ. *Portable Network Graphics* — переносимые сетевые изображения; файлы с расширением *png*) — формат, поддерживающий как режим истинного цвета, так и кодирование с палитрой; части изображения могут быть прозрачными и даже полупрозрачными (32-битное кодирование RGBA, где четвёртый байт задает прозрачность); изображение сжимается без искажения; анимация не поддерживается.

Свойства рассмотренных форматов сведены в таблицу ниже.

Таблица 4. Характеристики форматов цветов

Формат	Истинный цвет	С палитрой	Прозрачность	Анимация
BMP	Да	Да	—	—
JPEG	Да	—	—	—
GIF	—	Да	Да	Да
PNG	Да	Да	Да	—

Как известно, все виды информации хранятся в памяти компьютера в виде двоичных кодов, то есть цепочек из нулей и единиц. Получив такую цепочку, абсолютно невозможно сказать, что это — текст, рисунок, звук или видео. Например, код **11001000₂** может обозначать число **200**, код буквы «И», одну из составляющих цвета пикселя в режиме истинного цвета, номер цвета в палитре для рисунка с палитрой 256 цветов, цвета 8 пикселей чёрно-белого рисунка и т. п.

Как же компьютер разбирается в двоичных данных? В первую очередь нужно ориентироваться на расширение имени файла. Например, чаще всего файлы с расширением *txt* содержат текст, а файлы с расширениями *bmp*, *gif*, *jpg*, *png* — рисунки.

Однако расширение файла можно менять, как угодно. Например, можно сделать так, что текстовый файл будет иметь расширение *bmp*, а рисунок в формате *JPEG* — расширение *txt*. Поэтому в начало всех файлов специальных форматов (кроме простого текста — *txt*) записывается заголовок, по которому можно «узнать» тип файла и его характеристики.

Например, файлы в формате *BMP* начинаются с символов «BM», а файлы в формате *GIF* — с символов «GIF». Кроме того, в заголовке указывается размер рисунка и информация о его палитре, способ сжатия и т. п. Используя эту информацию, программа «расшифровывает» основную часть файла и выводит данные на экран характеристики, например, количество цветов в фиксированной таблице цветов.

Векторное кодирование

Для чертежей, схем, карт применяется другой способ кодирования, который позволяет не терять качество при изменении размеров изображения. Рисунок строится из простейших геометрических фигур (графических примитивов): линий, многоугольников, сглаженных кривых, окружностей, эллипсов. Такой рисунок называется векторным.

Векторный рисунок — это рисунок, построенный из простейших геометрических фигур, параметры которых (размеры, координаты вершин, углы наклона, цвет контура и заливки) хранятся в виде чисел.

Векторный рисунок можно «разобрать» на части, раставив мышью его элементы, а потом снова собрать полное изображение (рис. 21).



Рис. 21. Разбор векторного рисунка на части

Сделать что-то подобное с растровым рисунком не удастся.

При векторном кодировании для отрезка хранятся координаты его концов, для прямоугольников и ломаных — координаты вершин. Окружность и эллипс можно задать координатами прямоугольника, в который вписана фигура. Сложнее обстоит дело со сглаженными кривыми. На рисунке 22 изображена линия с опорными точками А, Б, В, Г и Д.

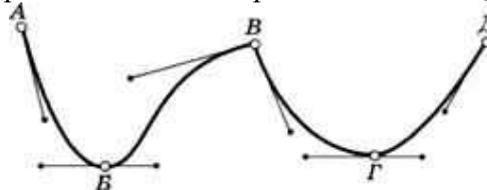


Рис. 22. Векторное представление линии

У каждой из этих точек есть «рычаги» (управляющие линии), перемещая концы этих рычагов, можно регулировать наклон касательной и кривизну всех участков кривой. Если оба рычага находятся на одной прямой, получается сглаженный узел (Б и Г), если нет, то угловой узел (В). Таким образом, форма этой кривой полностью задаётся координатами опорных точек и координатами рычагов. Кривые, заданные таким образом, называют кривыми Безье в честь их изобретателя — французского инженера Пьера Безье.

Векторный рисунок можно рассматривать как программу, в соответствии с которой строится изображение на конкретном устройстве вывода, с учётом особенностей этого устройства (например, разрешения экрана).

Векторный способ кодирования рисунков обладает значительными преимуществами по сравнению с растровым:

- при кодировании нет потери информации, если изображение может быть полностью разложено на простейшие геометрические фигуры (например, чертеж, схема, карта, диаграмма);
- объём файлов напрямую зависит от сложности рисунка — чем меньше элементов, тем меньше места занимает файл; как правило, векторные рисунки значительно меньше по объёму, чем растровые;
- при изменении размера векторного рисунка не происходит никакого искажения формы элементов, при увеличении наклонных линий не появляются «ступеньки», как при растровом кодировании (рис. 23).

Самый главный недостаток этого метода — он практически непригоден для кодирования изображений, в которых объекты не имеют чётких границ, например, для фотографий.



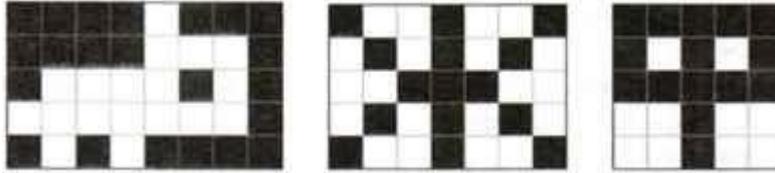
Рис. 23. Растровый и векторный рисунок

Среди форматов векторных рисунков отметим следующие:

- **WMF** (англ. *Windows Metafile* — метафайл *Windows*; файлы с расширением *wmf* и *emf*) — стандартный формат векторных рисунков в операционной системе *Windows*;
- **CDR** (файлы с расширением *cdr*) — формат векторных рисунков программы *CorelDRAW*;
- **AI** (файлы с расширением *ai*) — формат векторных рисунков программы *Adobe Illustrator*;
- **SVG** (англ. *Scalable Vector Graphics* — масштабируемые векторные изображения; файлы с расширением *svg*) — векторная графика для веб-страниц в Интернете.

Задание 5 (по теме кодирование графической информации)

1. Постройте двоичные коды для чёрно-белых рисунков и запишите их в шестнадцатеричной системе счисления:



Какие сложности у вас возникли? Как их можно преодолеть?

2. Постройте чёрно-белый рисунок шириной 8 пикселей, закодированный шестнадцатеричной последовательностью **2466FF6624**₁₆.
3. Постройте чёрно-белый рисунок шириной 5 пикселей, закодированный шестнадцатеричной последовательностью **3A53F88**₁₆.
4. Рисунок размером 10 x 15 см кодируется с разрешением 300 ppi. Оцените количество пикселей в этом рисунке.
5. Постройте шестнадцатеричный код для цветов, имеющих RGB-коды **(100, 200, 200)**, **(30, 50, 200)**, **(60, 180, 20)**, **(220, 150, 30)**.
6. Как бы вы назвали цвета, заданные на веб-странице в виде кодов **#CCCCCC**, **#FFCCCC**, **#CCCCFF**, **#000066**, **#FF66FF**, **#CCFFFF**, **#992299**, **#999900**, **#99FF99**?
Найдите десятичные значения составляющих RGB-кода.
7. Что такое глубина цвета? Как связаны глубина цвета и объём файла?
8. Какова глубина цвета, если в рисунке используется **65 536** цветов? **256** цветов? **16** цветов?
9. Для жёлтого цвета найдите красную, зелёную и синюю составляющие при 12-битном кодировании.
10. Сколько места в файле занимает палитра, в которой используются 64 цвета? 128 цветов?
11. Сколько байтов будет занимать код рисунка размером 40 x 50 пикселей в режиме истинного цвета? При кодировании с палитрой 256 цветов? При кодировании с палитрой 16 цветов? В чёрно-белом варианте (два цвета)?
12. Сколько байтов будет занимать код рисунка размером 80 x 100 пикселей в кодировании с глубиной цвета 12 битов на пиксель?
13. Для хранения растрового изображения размером 32 x 32 пикселя отвели 512 байтов памяти. Каково максимально возможное число цветов в палитре изображения?
14. Для хранения растрового изображения размером 128 x 128 пикселей отвели 4 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?
15. В процессе преобразования растрового графического файла количество цветов уменьшилось с 1024 до 32. Во сколько раз уменьшился информационный объём файла?
16. В процессе преобразования растрового графического файла количество цветов уменьшилось с 512 до 8. Во сколько раз уменьшился информационный объём файла?
17. Разрешение экрана монитора — 1024 x 768 точек, глубина цвета — 16 битов. Какой объём памяти требуется для хранения полноэкрannого изображения в данном графическом режиме?
18. После преобразования растрового 256-цветного графического файла в чёрно-белый формат (2 цвета) его размер уменьшился на 70 байтов. Каков был размер исходного файла (без учёта заголовка)?
19. Сколько памяти нужно для хранения 64-цветного растрового графического изображения размером 32 x 128 точек?
20. Какова ширина (в пикселях) прямоугольного 64-цветного растрового изображения, информационный объём которого 1,5 Мбайт, если его высота вдвое меньше ширины?
21. Какова ширина (в пикселях) прямоугольного 16-цветного растрового изображения, информационный объём которого 1 Мбайт, если его высота вдвое больше ширины?

Кодирование звуковой и видеоинформации. Оцифровка звука

Звук — это колебания среды (воздуха, воды), которые воспринимает человеческое ухо.

С помощью микрофона звук преобразуется в аналоговый (непрерывный) электрический сигнал. В любой момент времени аналоговый сигнал на выходе микрофона (ток или напряжение) может принимать любое значение в некотором интервале (рис. 24).



Рис. 24. Источники, приемники, представление звуковых сигналов

Как известно, современные компьютеры обрабатывают только *дискретные* сигналы (двоичные коды). Поэтому для работы со звуком необходима звуковая карта³ — специальное устройство, которое преобразует аналоговый сигнал, полученный с микрофона, в двоичный код, то есть в цепочку нулей и единиц. Эта процедура называется оцифровкой⁴.

Ситуация напоминает ту, с которой сталкиваются при кодировании рисунка: любая линия состоит из бесконечного числа точек, поэтому, чтобы закодировать «по точкам», нужна бесконечная память. Здесь тоже придётся использовать дискретизацию — представить аналоговый сигнал в виде набора чисел, то есть записать в память только значения сигнала в отдельных точках, взятых с некоторым шагом T по времени (рис. 25).

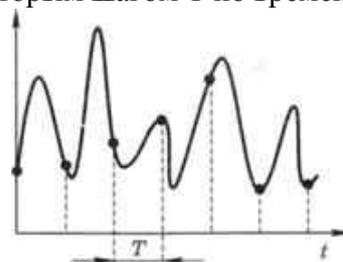


Рис. 25. Дискретизация звукового сигнала

Число T называется тактом или интервалом дискретизации, а обратная ему величина $1/T$ — частотой дискретизации. Частота дискретизации обозначается буквой f и измеряется в герцах (Гц) и килогерцах (кГц). Один герц — это один отсчёт в секунду, а 1 кГц — 1000 отсчётов в секунду. Чем больше частота дискретизации, тем точнее записывается сигнал, тем меньше информации теряется. Однако, при этом возрастает количество отсчётов, то есть информационный объём закодированного звука.

Для кодирования звука в компьютерах чаще всего используются частоты дискретизации 8 кГц (минимальное качество, достаточное для распознавания речи), 11 кГц, 22 кГц, 44,1 кГц (звуковые компакт-диски), 48 кГц (фильмы в формате DVD), а также 96 кГц и 192 кГц (высококачественный звук в формате DVD-audio). Выбранная частота влияет на качество цифрового звука.

Дело в том, что наушники и звуковые колонки — это аналоговые (не цифровые) устройства, и при проигрывании звука через звуковую карту компьютеру нужно как-то восстановить исходный аналоговый сигнал и передать его на наушники или звуковые колонки. В памяти есть только значения, снятые с интервалом T , остальная информация была потеряна при кодировании. В простейшем случае по ним можно восстановить ступенчатый сигнал, который будет существенно отличаться от исходного (до кодирования) (рис. 26). В современных звуковых картах для повышения качества звука этот ступенчатый сигнал сглаживается с помощью специальных фильтров.

³ В современных персональных компьютерах функции звуковой карты часто выполняет специальная микросхема материнской платы — аппаратный аудиокодек.

⁴ Оцифровка — это преобразование аналогового сигнала в цифровой код.

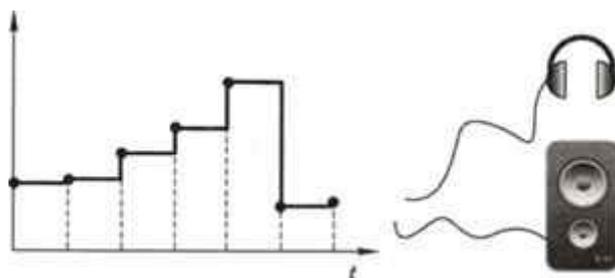


Рис. 26. Сглаживание ступенчатого сигнала

Для повышения качества звука, то есть для большего соответствия между сигналом, принятым микрофоном, и сигналом, выведенным из компьютера на колонки, нужно увеличивать частоту дискретизации, однако при этом увеличивается и объём файла. Как же выбрать оптимальную частоту при кодировании? Ответ на этот вопрос во многом зависит от свойств звука, который нужно закодировать.

С точки зрения математики, любой сигнал можно представить в виде суммы очень большого числа колебаний разных частот (гармоник). Если выбрать частоту дискретизации больше, чем удвоенная частота самой быстрой гармоники, то теоретически по отдельным отсчётам можно точно восстановить исходный аналоговый сигнал. Этот результат известен в радиотехнике как теорема Котельникова-Шеннона.

К сожалению, на практике всё несколько сложнее. Дело в том, что в реальных сигналах содержатся гармоники с очень высокими частотами, так что частота дискретизации, полученная с помощью теоремы Котельникова-Шеннона, будет также высока и объём файла — недопустимо велик. Однако средний человек слышит только звуки с частотами от 16 Гц до 20 кГц, поэтому все частоты выше 20 кГц можно «потерять» практически без ухудшения качества звука (человек не почувствует разницу!). Удвоив эту частоту (по теореме Котельникова-Шеннона), получаем оптимальную частоту дискретизации около 40 кГц, которая обеспечивает наилучшее качество, различимое на слух. Поэтому при высококачественном цифровом кодировании звука на компакт-дисках и в видеофильмах чаще всего используют частоты 44,1 и 48 кГц. Более низкие частоты применяют тогда, когда важно всячески уменьшать объём звуковых данных (например, для трансляции радиопередач через Интернет), даже ценой ухудшения качества.

Кроме того, что при кодировании звука выполняется дискретизация с потерей информации, нужно учитывать, что на хранение одного отсчёта в памяти отводится ограниченное место. При этом вносятся дополнительные ошибки.

Представим себе, что на один отсчёт выделяется 3 бита. При этом код каждого отсчёта — это целое число от 0 до 7. Весь диапазон возможных значений сигнала, от 0 до максимально допустимого, делится на 8 полос, каждой из которых присваивается номер (код). Все отсчёты, попавшие в одну полосу, получают одинаковый код (рис. 27).

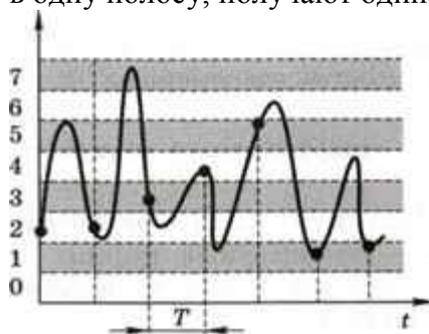


Рис. 27. Деление диапазона значений сигнала на 8 полос

Преобразование измеренного значения сигнала в целое число называется дискретизацией по уровню или *квантованием*. Эту операцию выполняет аналого-цифровой преобразователь (АЦП) — специальный блок звуковой карты.

Разрядность кодирования — это число битов, используемое для хранения одного отсчёта.

Недорогие звуковые карты имеют разрядность **16-18** битов, большинство современных — **24** бита, что позволяет использовать $2^{24} = 16\,777\,216$ различных уровней.

Объём данных, полученный после оцифровки звука, зависит от *разрядности кодирования и частоты дискретизации*:

$$V_{\text{зв. файла}} = R_{\text{разрядность зв. карты (в байтах)}} * Ч_{\text{частота дискр. (в герцах)}} * t_{\text{сек.}}$$

Например, если используется **16-разрядное** кодирование с частотой **44 кГц**, то за **1 с** выполняется **44 000** измерений сигнала, и каждое из измеренных значений занимает 16 битов (2 байта). Поэтому за 1 секунду накапливается $44\,000 \cdot 2 = 88\,000$ байтов данных, а за 1 минуту: $88\,000 \cdot 60 = 5\,280\,000$ байтов = $5280000 \text{ байтов} / 1024 = 5156,25 \text{ Кбайтов} = 5156,25 / 1024 \text{ Кб} = 5,035400... \approx 5 \text{ Мбайт}$.

Если записывается стереозвук (левый и правый каналы), это число нужно удвоить.

С помощью оцифровки можно закодировать любой звук, который принимает микрофон. В частности, это единственный способ кодирования человеческого голоса и различных природных звуков (шума прибоя, шелеста листвы и тому подобное).

Однако у этого метода есть и недостатки:

- при оцифровке звука всегда есть потеря информации (из-за дискретизации);
- звуковые файлы имеют, как правило, большой размер, поэтому в большинстве современных форматов используется сжатие.

Среди форматов оцифрованных звуковых файлов наиболее известны:

- **WAV** (англ. *Waveform Audio File Format*; файлы с расширением *wav*) — стандартный формат звуковых файлов в операционной системе *Windows*; сжатие данных возможно, но используется редко;
- **MP3** (файлы с расширением *mp3*) — самый популярный формат звуковых файлов, использующий сжатие с потерями: для значительного уменьшения объёма файла снижается качество кодирования для тех частот, которые практически неразличимы для человеческого слуха;
- **WMA** (англ. *Windows Media Audio*; файлы с расширением *wma*) — формат звуковых файлов, разработанный фирмой *Microsoft*; чаще всего используется сжатие для уменьшения объёма файла;
- **Ogg Vorbis** (файлы с расширением *ogg*) — свободный (не требующий коммерческих лицензий) формат сжатия звука с потерями.

Все эти форматы являются **потокowymi**, то есть можно начинать прослушивание до того момента, как весь файл будет получен (например, из Интернета).

Инструментальное кодирование звука

Существует еще один, принципиально иной способ кодирования звука, который можно применить только для кодирования *инструментальных мелодий*. Он основан на стандарте **MIDI** (англ. *Musical Instrument Digital Interface* — цифровой интерфейс музыкальных инструментов). В отличие от оцифрованного звука в таком формате хранятся последовательность нот, коды инструментов (можно использовать 128 мелодических и 47 ударных инструментов), громкость, тембр, время затухания каждой ноты и так далее. Фактически это программа, предназначенная для проигрывания звуковой картой, в памяти которой хранятся образцы звуков реальных инструментов (волновые таблицы, англ. *wave tables*).

Современные звуковые карты поддерживают многоканальный звук, то есть в звуковом файле может храниться несколько «дорожек», которые проигрываются одновременно. Таким образом, получается полифония — многоголосие, возможность проигрывать одновременно несколько нот. Количество голосов для современных звуковых карт может достигать **1024**.

Звук, закодированный с помощью стандарта **MIDI**, хранится в файлах с расширением *mid*. Существуют специальные клавиатуры, которые позволяют вводить звук и сразу сохранять его в формате *mid*.

Для проигрывания **MIDI**-файла используют синтезаторы — электронные устройства, имитирующие звук реальных инструментов (рис. 28). Простейшим синтезатором является звуковая карта компьютера.



Рис. 28. Изображение синтезатора

Главные *достоинства* инструментального кодирования:

- кодирование мелодии (нотной записи) происходит без потери информации;
- файл имеет значительно меньший объём по сравнению с оцифрованным звуком той же длительности.

Однако, произвольный звук (например, человеческий голос) в таком формате закодировать невозможно. Кроме того, производители сами выбирают образцы звуков (так называемые *сэмплы*, от англ. *samples* — образцы), которые записываются в память звуковой карты (нет единого стандарта). Поэтому звучание **MIDI**-файла может немного отличаться на разной аппаратуре.

Кодирование видеоинформации

Для того чтобы сохранить видео в памяти компьютера, нужно закодировать звук и изменяющееся изображение, причём требуется обеспечить их *синхронность* (одновременность). Для кодирования звука чаще всего используют оцифровку с частотой 48 кГц. Изображение состоит из отдельных растровых рисунков, которые меняются с частотой не менее 25 кадров в секунду, так что глаз человека воспринимает смену кадров как непрерывное движение. Это значит, что для *каждой секунды* видео нужно хранить в памяти *25 изображений*.

Если используется размер 768 x 576 точек (стандарты **PAL/SECAM**) и глубина цвета 24 бита на пиксель, то закодированная 1 секунда видео (без звука) будет занимать примерно 32 Мбайт, а 1 минута — около 1,86 Гбайт.

$$V_{\text{видео}} / 1 \text{сек} = (768 \times 576) * (24/8) * 25 \text{ байт} = 442368 * 3 * 25 \text{ байт} = 33177600 \text{ байт} = 33177600 / 1024 \text{ Кб} = 32400 \text{ Кбайт} = 32400 / 1024 \text{ Мбайт} = 31,640625 \text{ Мб} \approx 32 \text{ Мб}$$

Здесь $768 \times 576 = K_{\text{точек}}$ — количество точек фрагмента изображения,

24 бита = I — глубина цвета на бит,

25 кадров = $K_{\text{кадров}}$ — количество изображений в секунду.

Чтобы получить объём видео изображения в битах, надо перемножить.

$V_{\text{видео}} / 1 \text{сек} = K_{\text{точек}} * I * K_{\text{кадров}}$ в битах, чтобы получить в байтах надо результат разделить на количество битов в байте, то есть на 8. Чтобы получить результат в Кбайтах, надо результат разделить еще на 1024. Чтобы получить результат в Мбайтах, надо результат разделить еще на 1024.

$$V_{\text{видео}} / 1 \text{мин} = 31,640625 \text{ Мб} * 60 \text{ сек} = 1898,4375 \text{ Мб} = 1898,4375 / 1024 = 1,85394287109375 \text{ Гб} \approx 1,86 \text{ Гб}$$

Это недопустимо много, поэтому в большинстве форматов видеоизображений используется сжатие с потерями. Это значит, что некоторые незначительные детали теряются, но «обычный» человек (непрофессионал) не почувствует существенного ухудшения качества. Основная идея такого сжатия заключается в том, что за короткое время изображение изменяется очень мало, поэтому можно запомнить базовый кадр, а затем сохранять только изменения. Как только изображение существенно изменится, выбирается новый базовый кадр.

В последние годы часто используются форматы видео высокой чёткости (англ. **HD** — *High Definition*) — 1280 x 720 точек и 1920 x 1080 точек, предназначенные для просмотра на широкоформатных экранах с соотношением сторон 16:9.

Наиболее известны следующие видеоформаты:

- **AVI** (англ. *Audio Video Interleave* — чередующиеся звук и видео; файлы с расширением *avi*) — формат видеофайлов, разработанных фирмой *Microsoft* для системы *Windows*; может использовать разные алгоритмы сжатия;
 - **WMV** (англ. *Windows Media Video* - файлы с расширением *wmv*) — система кодирования видео, разработанная фирмой *Microsoft*; может использовать разные алгоритмы сжатия;
 - **MPEG** (файлы с расширением *mpg, mpeg*) — формат кодирования видеoinформации, использующий один из лучших алгоритмов сжатия, который разработала экспертная группа по вопросам движущегося изображения (англ. *Motion Picture Experts Group*);
 - **MP4** (файлы с расширением *mp4*) — формат видеофайлов, позволяющий хранить несколько потоков видео высокой чёткости, а также субтитры;
 - **MOV** (англ. *Quick Time Movie*; файлы с расширением *mov*) — формат видеофайлов, разработанный фирмой *Apple*;
- WebM** — открытый (не требующий оплаты лицензии) видеоформат, который поддерживается в современных браузерах без установки дополнительных модулей.

Задание 6. (по теме кодирование видео информации)

1. Заполните пустые ячейки таблицы, вычислив объёмы звуковых файлов (без сжатия):

Частота дискретизации, кГц	8	8	11	11	22	22	44,1	44,1	48	48
Глубина кодирования, битов	8	8	16	16	16	8	24	8	8	24
Моно/стерео	моно	стерео	моно	стерео	моно	стерео	моно	стерео	стерео	стерео
Время звучания, с	16	8	64	32	32	32	256	128	4	4
Объём файла, Кб										

2. Заполните пустые ячейки таблицы, вычислив время звучания записи (объёмы файлов приведены без учёта сжатия):

Частота дискретизации, кГц	8	8	11	11	22	22	44,1	44,1	48	48
Глубина кодирования, битов	16	24	8	8	8	24	16	24	16	16
Моно/стерео	моно	стерео	моно	стерео	моно	стерео	моно	стерео	моно	стерео
Объём файла, Кб	125	375	1375	1375	6875	4125	11025	33075	375	1875
Время звучания, с										

3. Заполните пустые ячейки таблицы, вычислив глубину кодирования звука (объёмы файлов приведены без учёта сжатия):

Частота дискретизации, кГц	8	8	11	11	22	22	44,1	44,1	48	48
Моно/стерео	моно	стерео	моно	стерео	моно	стерео	моно	стерео	моно	моно
Время звучания, с	16	4	238	64	320	16	256	64	40	80
Объём файла, Кб	375	125	4125	4125	20625	1375	11025	11025	1875	11250
Глубина кодирования, битов										

4. Заполните пустые ячейки таблицы, вычислив частоту дискретизации звука (объёмы файлов приведены без учёта сжатия):

Глубина кодирования, битов	16	8	16	24	16	16	8	24	16	24
Моно/стерео	моно	стерео	стерео	стерео	стерео	моно	моно	стерео	моно	стерео
Время звучания, с	64	4	64	64	16	128	320	8	4	4
Объём файла, Кб	1375	375	11025	4125	1375	11025	6875	375	375	1125
Частота дискретизации, кГц										

5. Кадры видеозаписи закодированы в режиме истинного цвета (24 бита на пиксель) и сменяются с частотой 25 кадров в секунду, запись содержит стереофонический звук. Остальные параметры для разных вариантов заданы в таблице. Оцените объём 1 минуты видеозаписи в мегабайтах (с точностью до десятых). Сколько минут такой записи поместится на стандартный CD-диск объёмом 700 Мбайт?

Ширина кадра, пиксели	320	320	640	640	720	720	720	720	1920	1920
Высота кадра, пиксели	240	240	480	480	480	480	576	576	1080	1080
Частота дискретизации, кГц	11	48	48	48	22	248	22	48	48	48
Глубина кодирования, битов	24	16	24	16	16	16	24	24	16	24
Степень сжатия	10	8	6	4	10	12	8	6	8	10
Объём файла, Кб										
Поместится на CD-диск, мин										

Приемы и методы работы со сжатыми данными

Алгоритм RLE. Алгоритм Хаффмана

Теоретические основы сжатия данных

Регулярно возникает необходимость сжимать данные перед их размещением в архивах или при передаче по каналам связи. Соответственно, существует и обратная необходимость восстановления данных из уплотненных архивов.

Характерной особенностью большинства типов данных является определенная *избыточность*. Например, у видеоданных избыточность больше, чем у графических, а степень избыточности графических данных в несколько раз больше, чем текстовых. Кроме того, степень избыточности данных зависит от принятой системы кодирования. Например, кодирование информации средствами русской азбуки на 20-30% больше, чем кодирование адекватной информации средствами английского языка.

Для человека избыточность информации связана с представлением о ее качестве, так как избыточность улучшает восприятие в неблагоприятных условиях (просмотр телепередачи при наличии помех, восстановление поврежденных графических материалов и т.д.). При обработке информации избыточность также играет важную роль. Например, при преобразовании данных избыточность используется для повышения их качества. Однако при хранении готовых документов или их передаче избыточность можно уменьшить, что дает эффект *сжатия данных*.

Если методы сжатия данных применяются к готовым документам, то термин *сжатия данных* подменяют термином *архивация данных*, а программные средства, выполняющие эти операции, называют *архиваторами*.

Объекты сжатия. В зависимости от того, в каком объекте размещены данные, подвергаемые сжатию, различают:

- уплотнение (архивацию) файлов;
- уплотнение (архивацию) папок;
- уплотнение дисков.

Уплотнение файлов применяют для уменьшения их размеров при передаче по каналам связи или транспортировки на внешнем носителе малой емкости.

Уплотнение папок (архивирование) используют как средство архивации данных перед длительным хранением, в частности, при резервном копировании.

Уплотнение дисков служит целям повышения эффективности использования их рабочего пространства, и применяют к дискам недостаточной емкости.

Обратимость сжатия. Несмотря на изобилие алгоритмов сжатия данных, теоретически есть только три способа уменьшения их избыточности:

- изменение содержания данных;
- изменение структуры данных;
- изменение содержания и структуры данных.

Если при сжатии данных происходит изменение их содержания, метод сжатия необратим и при восстановлении данных из сжатого файла не происходит полного восстановления исходной последовательности. Такие методы называют также *методами сжатия с регулируемой потерей информации*. Они применимы для данных, формальная утрата части содержания которых, не приводит к значительному снижению потребительских свойств. Это относится к мультимедийным данным. Такие методы обеспечивают более высокую степень сжатия, чем обычные обратимые методы, но их нельзя применять к текстовым документам, базам данных и программному коду.

Форматы сжатия с потерей информации:

- **.JPG** для графических данных;
- **.MPG** для видеоданных;
- **.MP3** для звуковых данных.

Если при сжатии данных происходит только изменение их структуры, то метод сжатия *обратим*. Из результирующего кода можно восстановить исходный массив, применяя обратный метод. Обратимые методы применяются для сжатия любых типов данных.

Форматы сжатия без потери информации:

- **.GIF, .TIF, .PCX** и многие другие для графических данных;
- **.AVI** для видеоданных;
- **.ZIP, .ARJ, .RAR, .LZH, .LH** и подобные для любых типов данных.

Алгоритмы обратимых методов. При исследовании методов сжатия следует иметь в виду существование следующих теорем:

- для любой последовательности данных существует предел сжатия, который не может быть превышен без потери части данных.
- для любого алгоритма можно указать последовательность данных, для которой он дает лучшую степень сжатия, чем другие методы.
- для любого алгоритма можно указать последовательность данных, для которой данный алгоритм вообще не позволит получить сжатия.

Таким образом, наивысшую эффективность методы сжатия демонстрируют для данных разных типов и разных объемов.

Алгоритм RLE

В основу алгоритмов *RLE (Run-Length Encoding)* положен принцип выявления повторяющихся последовательностей данных и замены их простой структурой, в которой указывается *код данных* и *коэффициент повтора*.

Например, рассмотрим последовательность:

0; 0; 0; 127; 127; 0; 255; 255; 255; 255

Заменим на сжатую последовательность, в которой указано число, а за ним количество повторов (идуших подряд) данного числа.

Первое число — **0**, количество повторов — **3**.

Следующее число — **127**, количество повторов — **2**.

Затем следует **0**, количество повторов — **1**.

И, наконец, **225**, количество повторов — **4**.

В результате получаем: **0; 3; 127; 2; 0; 1; 255; 4**

Итак, в исходной последовательности — **10** элементов, а в сжатой последовательности — **8** элементов.

Вспомним, что для кодирования целых чисел (от **0** до **255**) достаточно одного байта, тогда объем исходных данных ($V_{исх}$) равен **10** байтам, а сжатых данных ($V_{сж}$) равен **8** байтам.

Подсчитаем коэффициент сжатия по формуле:

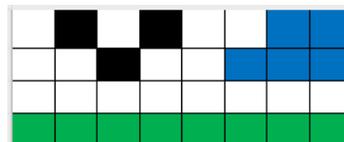
$$k = V_{сж} / V_{исх} \quad (***)$$

В рассмотренном примере коэффициент сжатия $k = 8$ байт / 10 байт = $0,8$.

Программные реализации этого алгоритма отличаются простотой, высокой скоростью работы, но в среднем обеспечивают недостаточное сжатие. Наилучшими объектами для сжатия являются графические файлы, в которых большие одноцветные участки изображения кодируются длинными последовательностями одинаковых байтов. Этот метод может дать эффект для баз данных, с таблицами с фиксированной длиной полей. Для текстовых данных методы, основанные на данном алгоритме, мало эффективны.

Пример:

Дано: цветное изображение
Найти: коэффициент сжатия k



Решение:

Метод кодирования цвета — индексный. В данном случае для кодирования цвета одной точки изображения используется **1** байт. Следовательно, количество цветов — **256**. Код цвета каждой точки раstra выражается его номером (индексом) в палитре, которая обязательно прилагается к графическим данным.

Итак, пусть палитра такова:

Индекс	Цвет		
0			белый
1			черный
2			синий
3			зеленый

Информацию о цвете изображения можно закодировать в виде последовательности чисел, которые соответствуют коду цвета точки из палитры. Будем применять кодирование по строкам изображения, начиная с верхнего левого угла. В результате получим:

0; 1; 0; 1; 0; 0; 2; 2; 0; 0; 1; 0; 0; 2; 2; 2; 0; 0; 0; 0; 0; 0; 0; 0; 3; 3; 3; 3; 3; 3; 3; 3

Применим алгоритм **RLE** для сжатия данной последовательности:

0; 1; 1; 1; 0; 1; 1; 1; 0; 2; 2; 2; 0; 2; 1; 1; 0; 2; 2; 3; 0; 8; 3; 8

Подсчитаем коэффициент сжатия по формуле (***) на предыдущей странице. Учтывая, что для кодирования одного элемента последовательностей используется один байт, получаем:

$V_{исх} = 32$ байта

$V_{сж} = 24$ байта

$k = 24$ байта / 32 байта = $0,75$.

Ответ: $k = 0,75$

Кодирование графического изображения

Рассмотрим каким образом можно восстановить битовый (черно-белый) рисунок по заданной последовательности нулей и единиц.

Пример: (обратная задача)

Дано: последовательность, кодирующая столбцы (сверху вниз): **1F, 31, 75, F1, 75, 31, 1F, 01**

Найти: ч/б рисунок (**8x8**), сжать по строкам (алгоритм **RLE**), $k = ?$

Решение:

Данная последовательность представлена в шестнадцатеричной системе счисления. Для восстановления рисунка надо перевести ее в двоичную. Дополнить (если это необходимо) каждое полученной двоичное число до четырех знаков, дописав нужное количество нулей перед ним.

$1_{16} = 0001_2$

$F_{16} = 15_{10} = 1111_2$ $15 : 2 = 7$ (остаток от деления 1)

$3_{16} = 0011_2$ $7 : 2 = 3$ (остаток от деления 1)

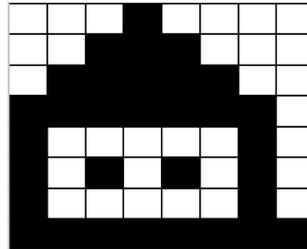
$7_{16} = 0111_2$ $3 : 2 = 1$ (остаток от деления 1)

$5_{16} = 0101_2$ $5 : 2 = 2$ (остаток от деления 1)

$0_{16} = 0000_2$ $2 : 2 = 1$ (остаток от деления 0)

Перепишем последовательность **1F, 31, 75, F1, 75, 31, 1F, 01** в двоичном виде:
0001 1111, 0011 0001, 0111 0101, 1111 0001,
0111 0101, 0011 0001, 0001 1111, 0000 0001

Теперь, учитывая, что нулем кодируется белый цвет, а единицей — черный, закрасим столбцы в растре 8x8 сверху вниз:



Применим теперь кодирование *по строкам* изображения, начиная с верхнего левого угла. В результате получим:

0; 0; 0; 1; 0; 0; 0; 0; 0; 0; 1; 1; 1; 0; 0; 0; 0; 0; 1; 1; 1; 1; 1; 0; 0; 0; 1; 1; 1; 1; 1; 1; 0; 0; 1; 0; 0; 0; 0; 0; 1; 0; 1; 0; 1; 0; 1; 0; 1; 0; 0; 0; 0; 0; 1; 0; 1; 1; 1; 1; 1; 1; 1; 1

Применим алгоритм **RLE** для сжатия данной последовательности:

0; 3; 1; 1; 0; 6; 1; 3; 0; 4; 1; 5; 0; 2; 1; 7; 0; 1; 1; 1; 0; 5; 1; 1; 0; 1; 1; 1; 0; 1; 1; 1; 0; 1; 1; 1; 0; 1; 1; 1; 0; 5; 1; 1; 0; 1; 1; 8

Подсчитаем коэффициент сжатия по формуле (**). Учтем, что для кодирования одного элемента последовательностей используется один байт (так как считаем полученный рисунок черно-белым, а не битовым) получаем:

$V_{исх} = 64$ байта

$V_{сж} = 52$ байта

$k = 52 \text{ байта} / 64 \text{ байта} = 0,81$

Ответ: **$k = 0,81$**

Алгоритм Хаффмана

В основе алгоритма лежит кодирование не байтами, а битовыми группами.

Перед началом кодирования производится *частотный анализ* документа (файла) и выявляется частота повтора каждого из встречающихся символов.

Чем чаще встречается символ, тем меньшим количеством битов он кодируется.

Образующаяся в результате кодирования структура прикладывается к сжатому документу в качестве таблицы соответствия (*словаря*).

В связи с обязательным приложением словаря на файлах малых размеров алгоритм малоэффективен.

Пример:

Пусть имеется текст, содержащий **100** символов и проведен анализ частоты вхождения символов.

Дано: **A – 10, B – 20, C – 30, D – 5, E – 25, F – 10.**

Найти: коэффициент сжатия **k**

Решение:

Для удобства запишем данные задания в таблицу:

Символ	A	B	C	D	E	F
Число вхождений	10	20	30	5	25	10

Расставим символы по убыванию частот: **C(30), E(25), B(20), A(10), F(10), D(5)**

Символ	C	E	B	F	A	D
Число вхождений	30	25	20	10	10	5

Какой из символов, имеющих одинаковую частоту вхождения, поставить в таблице раньше, в принципе, не имеет значения, поменяется лишь кодировка.

Строим дерево, соединяя символы (или узлы) с наименьшими частотами, и, записывая в получившийся новый узел сумму частот.

Возьмем из последней таблицы 2 символа с наименьшей частотой. В нашем случае это **D** (с частотой вхождения **5**) и какой-либо из символов **F** или **A** (частота вхождения которых равна **10**). Можно выбрать любой из них, например, **A**.

Соединим символы **D (5)** и **A (10)** и запишем сумму частот – **15**.

Символ	C	E	B	F	A	D
Число вхождений	30	25	20	10	10	5

Номер в рамке - сумма частот символов **D** и **A**.

Расположим этот новый узел (то есть узлы **A** и **D**) непосредственно за узлом с самой большой частотой вхождения (то есть за **C**) — **C(30)**, **A(10)**, **D(5)**, а далее узлы с возрастающей частотой, то есть сначала **F(10)**, а затем **B(20)** и **E(25)**.

Символ	C	A	D	F	B	E
Число вхождений	30	10	5	10	20	25

Затем снова ищем два символа с самыми низкими частотами вхождения, исключая при этом из просмотра узлы **D** и **A** и рассматривая вместо них новый "узел" с суммарной частотой вхождения **15**. Теперь самая низкая частота вхождения у узла **F (10)** и нового "узла" (**15**).

Снова сделаем операцию слияния узлов.

Соединяем **F (10)** и "узел" с частотой вхождения (**15**), записываем сумму – **25**.

Символ	C	A	D	F	B	E
Число вхождений	30	10	5	10	20	25

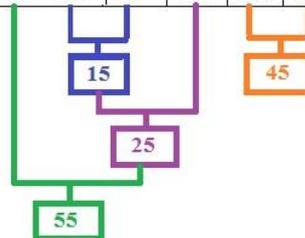
Суммарная частота вхождений "узла" с частотой **15** и только что полученного "узла" с частотой **25** равна **40**. Эта частота вхождения больше, чем частота вхождения для узла **C(30)**, поэтому переходим к поиску узлов, имеющих минимальную частоту вхождений среди оставшихся узлов (еще не рассмотренных – **B(20)**, **E(25)**).

Рассматриваем таблицу снова для следующих двух символов (**B** и **E**). Каждый из узлов **B** и **E** имеет частоту вхождения меньше, чем **40**. Соединяем узел **B(20)** и узел **E(25)**, записываем сумму – **45**.

Символ	C	A	D	F	B	E
Число вхождений	30	10	5	10	20	25

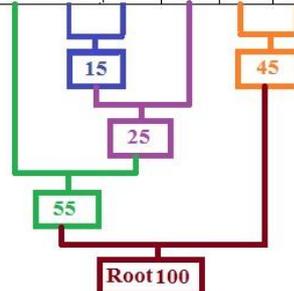
Имеем теперь узлы **C(30)**, "узел" с частотой **25** и "узел" с частотой **45**. Среди них выбираем 2 узла с минимальной частотой вхождения. Анализируя, видим, что частоту вхождения **25** имеют опять 2 узла. Выбираем любой из них. Для рисунка удобнее взять "суммарный узел **A-D-F**". Соединяем узел **C(30)** и этот суммарный узел с частотой **25**, записываем сумму – **55**.

Символ	C	A	D	F	B	E
Число вхождений	30	10	5	10	20	25



Наконец, соединяем “узел” с частотой вхождения **55** и “узел” с частотой вхождения **45**, записываем сумму – **100**.

Символ	C	A	D	F	B	E
Число вхождений	30	10	5	10	20	25



Видим, что описанный алгоритм продолжается, пока все "дерево" не будет сформировано, то есть пока все не сведется к одному узлу. Обратите внимание на то, что дерево должно быть **бинарным**, то есть из корня должны исходить лишь две ветви.

Следующий шаг начинается, когда дерево построено. Шаг состоит в том, что начинается кодирование символов, записывая путь от корня дерева к каждому символу, считая, что **левый поворот** кодируется **0** (нулем), а **правый поворот** – **1** (единицей).

Начинается процесс кодирования всегда из корня (**Root**) – формального узла. Затем кодируется первый символ (лист дерева **C**). Далее прослеживается путь вверх по дереву с учетом всех поворотов ветвей: если выполняется левый поворот, то в бит записывается **0**, если выполняется правый поворот, то в бит записывается **1**.

Так для **C**, требуется выполнить левый поворот к **55** (и записать **0**), затем снова выполнить левый поворот к самому символу (и записать **0**). Таким образом, получаем **код Хаффмана** для **символа C** в нашем примере - **00**.

Для следующего символа (**A**) требуется выполнить повороты: левый, правый, левый, левый, что соответствует последовательности **0100**.

Выполнив движение от корня к конкретному символу указанным способом для всех символов, получим

C = 00 (2 бита)	повороты: левый, левый.
A = 0100 (4 бита)	повороты: левый, правый, левый, левый.
D = 0101 (4 бита)	повороты: левый, правый, левый, правый.
F = 011 (3 бита)	повороты: левый, правый, правый.
B = 10 (2 бита)	повороты: правый, левый.
E = 11 (2 бита)	повороты: правый, правый.

Таким образом, **словарь** (таблица соответствия) будет выглядеть так:

B = 10	F = 011
C = 00	D = 0101
E = 11	A = 1100

При кодировании текста заменим символы, содержащиеся в тексте, на соответствующие им коды из словаря.

Подсчитаем теперь коэффициент сжатия по формуле (***)

Как мы помним, каждый символ в системе ASCII кодируется 8-ю битами (один байт). При использовании алгоритма Хаффмана, число бит для представления символов в соответствии с нашим словарем составляет:

A – 4 бита, **B** – 2 бита, **C** – 2 бита, **D** – 4 бита, **E** – 2 бита, **F** – 3 бита.

Количество символов в исходном тексте равно числу в корне дерева, а именно:

$$A(10)+B(20)+C(30)+D(5)+E(25)+F(10)= 10 + 20 + 30 + 5 + 25 + 10 = 100$$

Вычислим объём, требуемый для исходного текста и закодированного:

$$V_{исх} = 100 \text{ символов} * 1 \text{ байт} = 100 \text{ байт} = 800 \text{ бит}$$

$V_{сж}$ – это сумма произведений частоты символа на число бит кода

$$V_{сж} = 10 * 4 \text{ бит} + 20 * 2 \text{ бит} + 30 * 2 \text{ бит} + 5 * 4 \text{ бит} + 25 * 2 \text{ бит} + 10 * 3 \text{ бит} \\ = 40 \text{ бит} + 40 \text{ бит} + 60 \text{ бит} + 20 \text{ бит} + 50 \text{ бит} + 30 \text{ бит} = 240 \text{ бит}$$

$$k = 240 \text{ бит} / 800 \text{ бит} = 0,3$$

Ответ: $k = 0,3$

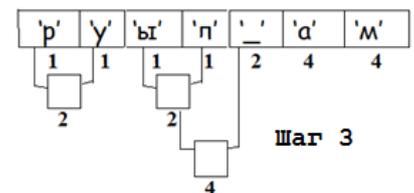
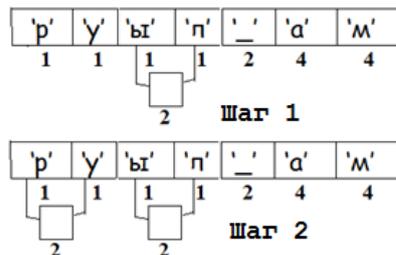
Замечание. В данном примере коэффициент сжатия считается без учета словаря, который необходим для восстановления исходного текста.

Синтетические алгоритмы. Рассмотренные выше алгоритмы в чистом виде на практике не применяются из-за того, что эффективность каждого сильно зависит от начальных условий. Поэтому современные средства архивации данных используют более сложные алгоритмы, основанные на комбинации нескольких теоретических методов. Общий принцип таких *синтетических алгоритмов* — это предварительный просмотр и анализ исходных данных для индивидуальной настройки алгоритма на особенности обрабатываемого материала.

Рассмотрим теперь пример сжатия фразы «МАМА МЫЛА РАМУ», используя алгоритм RLE.

"МАМА_МЫЛА_РАМУ"

Символ	Частотность
'М'	4
'А'	4
'_'	2
'Ы'	1
'Л'	1
'Р'	1
'У'	1



M=0

"МАМА_МЫЛА_РАМУ"

a=11
_ =0011

011011000100001011100001100001
1 В 1 0 В 8 6 1
1В10В861

Задание 7 для самостоятельной работы

(выполнять НЕОБЯЗАТЕЛЬНО, предлагается исключительно для самоконтроля):

1. Нарисовать цветное изображение (размер произвольный). Количество цветов не больше 256. Указать палитру цветов. Найти коэффициент сжатия (сжимать по строкам, используя алгоритм RLE).

Практическое занятие №1. Калькулятор Приемы работы с инструментом Калькулятор

1. Запустите один из видов приложения **Калькулятор** (**Пуск** → **Найти** → **Калькулятор**). Изображения представлены в Microsoft Office 2010 (рис. 29). Либо (**Пуск** – **Программы** – **Стандартные** — **Калькулятор** (Windows 8))

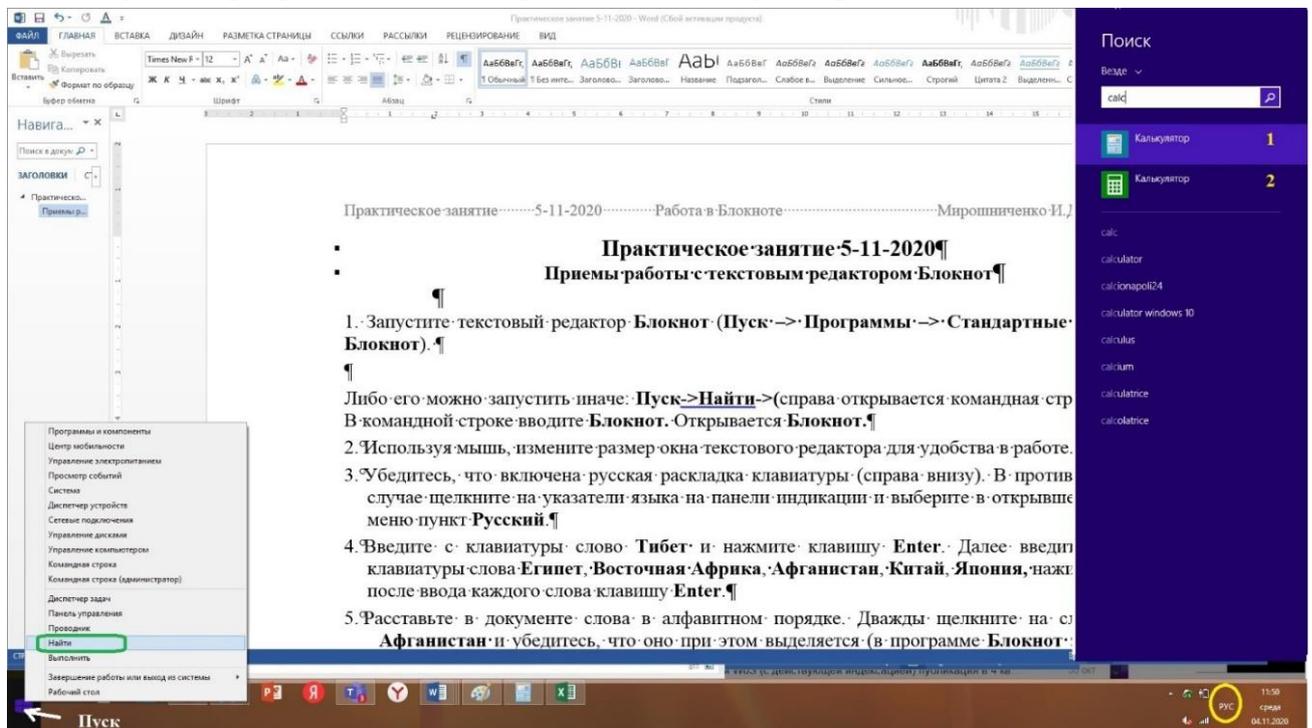


Рис. 29. Поиск приложения Калькулятор

Рассмотрим общие возможности Калькулятора под цифрой 1 и цифрой 2. Затем отметим дополнительные особенности и возможности Калькуляторов.

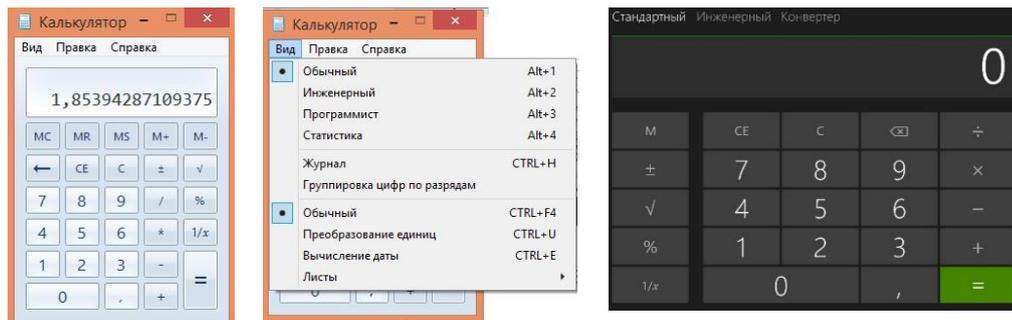


Рис. 30. Калькулятор из стандартного набора программ под цифрой 1 (слева) и его виды

Рис. 31. Калькулятор под цифрой 2 (справа) и его виды

2. Как видим, оба калькулятора имеют, прежде всего, практически одинаковый вид — *обычный* (в случае №1, рис. 30) и *стандартный* (в случае №2, рис. 31). Этот вид содержит цифры для набора чисел, арифметические операции (+, -, /(\div) и *), математические действия $1/x$, %, $\sqrt{\quad}$, \pm , десятичная точка, возврат на символ назад (\leftarrow или \Leftarrow), символ = (выполнение операции, получение результата). Эти кнопки позволяют выполнять довольно широкий диапазон обыденных операций или вычислений.

Кроме основных кнопок в обоих калькуляторах есть следующие кнопки:

C - символ обнуления (обнуление содержимого буфера или регистра) в начале вычислений.

CE - сброс последнего набираемого числа. Например, $8+4$ **CE** = Результат = **8**.

В обычном виде калькулятора №1 есть еще дополнительные кнопки:

MS (Memory save) — записывает в память число, которое на экране.

Пример: производим вычитание $76 - 43 = 23$, нажимаем **MS**, калькулятор запоминает число **23**, с которым в дальнейшем можно выполнять операции.

MR (Memory read) — выводит число из памяти на экран.

Пример: нажимаем **MS**, на экране появится число из памяти: **23**

MC (Memory clear) — удаляет число из памяти.

Пример: нажимаем **MS**, число **23** удаляется из памяти.

M+ (Memory plus) — прибавление числа на экране к числу, записанному в памяти. На некоторых калькуляторах используется вместо **MS**, так как по умолчанию память равна нулю.

Пример: в памяти записано число **15**, производим сложение $39 + 12 = 51$, нажимаем **M+**, калькулятор складывает **51** и **15** и записывает в память **66**.

M- (Memory minus) — вычитание числа на экране из числа, записанного в памяти.

Пример: в памяти записано число **62**, производим сложение $41 + 17 = 58$, нажимаем **M-**, калькулятор из **62** вычитает **58** и записывает в память **4**.

В некоторых калькуляторах встречается кнопка **MRC**

MRC (Memory read-clear) — вывод числа из памяти на экран и обнуление памяти.

Пример: в памяти записано число **36**, нажимаем **MRC**, на экране появится число из памяти - **36**, нажимаем **MRC** ещё раз - память очищается.

Кроме описанных кнопок калькулятор под №1 (рис. 30) позволяет использовать возможности копирования и вставки с помощью основного меню: **Правка->Копировать**, **Правка->Вставить**

Задания (к Практической №1)

1. Привести примеры (не менее 5) использования как основных кнопок, так и кнопок **C**, **CE**, **M+**, **M-**, **MC**, **MR**, **MS**

2. Как можно видеть из рисунков 30 (для Калькулятора 1) и 31 (для Калькулятора 2) оба Калькулятора имеют *инженерный* вид. В инженерный вид переходим с помощью команды **Вид->Инженерный**

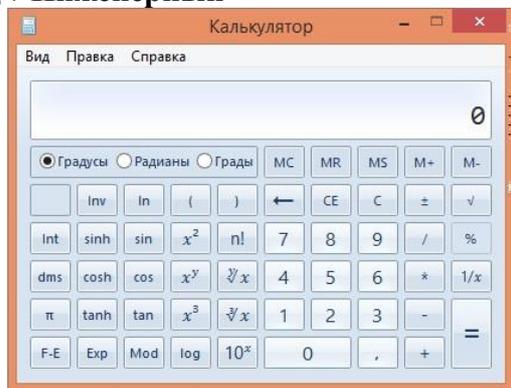


Рис. 32. Инженерный вид Калькулятора №1

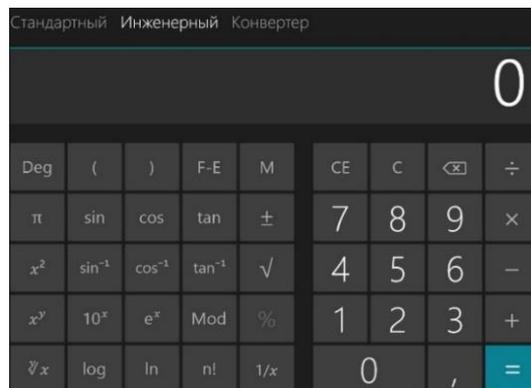


Рис. 33. Инженерный вид Калькулятора №2

В инженерном режиме можно осуществлять более сложные расчеты, чем в стандартном режиме. Как видно из рисунков 32 и 33 можно использовать такие функции, как

$\sin x$	$\log x$ (десятичный логарифм)	$x \text{ Mod } y$ (остаток от деления по модулю)	x^2	корень третьей степени из x	$\sinh x$ (гиперболический синус)
$\cos x$	$\ln x$ (натуральный логарифм)	Int (целая часть десятичного числа с удалением дробной части)	x^3	корень степени y из x	$\cosh x$ (гиперболический косинус)
$\tan x$ (тангенс)	$\text{Exp } x$ (экспонента)	значение π	x^y	10^x	\tanh (гиперболический тангенс)
dms (преобразует отображаемое число в формат градусы-минуты-секунды, предполагая, что оно выражается в градусах)		F-E представление числа в экспоненциальной форме: $40=4*e+1$		Inv – возможность использовать обратные функции	

В инженерном режиме точность калькулятора составляет 32 значащие цифры.

Чтобы выполнить любую функцию инженерного вида, нужно в окошко ввода сначала записать число, а потом нажать клавишу функции, например, последовательность ввода: **3,14; sin;** = отобразится в окне ввода как **0,0547759098534336216875256184619**.

К результату можно применить еще функцию, в этом случае нажимают на нужную кнопку, например, **log**. В окошке ввода отобразится **log(sin(3,14)) = log(0,01...)**.

Примечание.

1. Если сочетать несколько операторов (+, -, *, /), то калькулятор использует стандартную последовательность действий, то есть выполняет вычисления в следующем порядке:

- 1.1. Скобки
- 1.2. Возведение в степень и извлечение корня
- 1.3. Умножение и деление
- 1.4. Сложение и вычитание

2. Обратите внимание, что в инженерном виде есть радиокнопки: Радианы, Градусы, Грады. Мера угла может быть выражена в градусах (*degrees*), радианах (*radians*) или в градах (*grads*).

Соотношение между этими единицами таково

$$360^\circ = 2\pi = 400g.$$

Один *град* — это сотая доля прямого угла.

Один *радиан* — длина дуги, равная радиусу.

Грады — это единицы, в которых прямой угол равен 100 *град*. В целом, это всего лишь единицы измерения углов, для синусов, косинусов и т.д.

По умолчанию вычисления ведутся в радианах. Но если нужно перейти к градусам или градам, просто нажимают на соответствующую радиокнопку (рис. 32) для Калькулятора №1. Для Калькулятора №2 (рис. 33) также по умолчанию вычисления проводятся для градусов (при кнопке **Deg**). Если же необходимо перейти к радианам, требуется нажать кнопку **Deg**, при отжатии эта кнопка уже будет иметь название **Rad**.

3. Для калькулятора №1 имеется программистский вид, представленный на рисунке 34. В этот режим переходим следующим образом: **Вид->Программист**



Рис. 34. Вид Калькулятора №1 в режиме “программист”



Рис. 35. Ввод числа 84 в десятичной системе с представлением в 1 байте

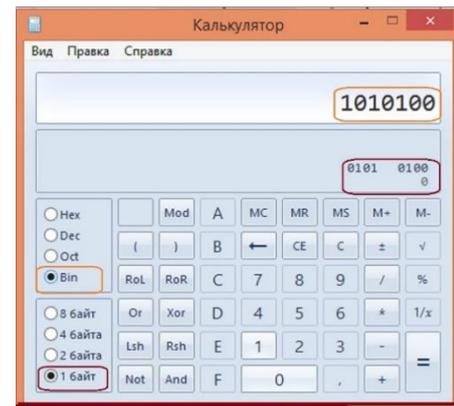


Рис. 36. Непосредственный перевод числа 84 в десятичной системе в двоичную систему в 1 байте

В режиме *программист* точность калькулятора достигает 64 разрядов в зависимости от выбранного размера машинного слова.

В режиме *программист* используются следующие блоки клавишей: размер машинного слова (1, 2, 4 и 8 байтов), блок цифр (0, 1, 2, 3, ..., 9, A, B, C, D, E, F), простейшие арифметические операции, как в обычном режиме, логические операции (*Not*, *And*, *Or*, *XOR*, *Lsh*, *Ror*, *RoL*), блок систем счисления (*Hex*, *Dec*, *Oct*, *Bin*)

По умолчанию все вычисления производятся в десятичной системе счисления (**Dec**). Используя блок систем счисления, удобно проверять правильность перевода чисел из одной системы счисления в другую. Например, можно в десятичной системе ввести некоторое число (для определенности **84**₁₀ в 1 байте – рисунок 35). Ниже строки ввода расположено представление числа **84**₁₀ в двоичном виде. Переведя теперь в двоичную систему выбором системы счисления **Bin** получим в окне ввода непосредственное представление этого числа в двоичном виде (рис. 36). Обратите внимание, что ниже окна ввода двоичное представление **01010100**₂ (рисунки 36, 37, 38) в качестве своей старшей цифры (левый 0) представляет знак числа (в нашем случае плюс)

Аналогично можно переключить в восьмеричную систему (рис. 37), выбрав кнопку радио **Oct** и получив в окне ввода числовое значение **124**₈, или в шестнадцатеричную систему (рис. 38), выбрав кнопку радио **Hex** и получив в окне ввода числовое значение **54**₁₆.

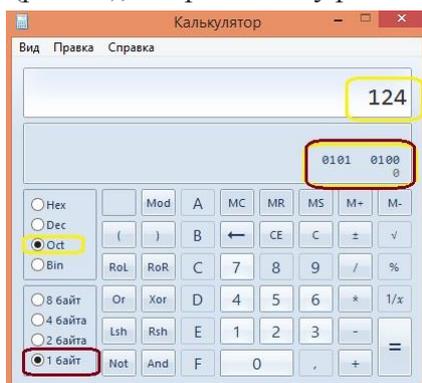


Рис. 37. Непосредственный перевод числа 84 в десятичной системе в восьмеричную систему в 1 байте

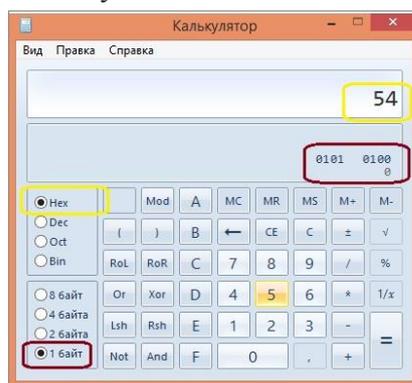


Рис. 38. Непосредственный перевод числа 84 в десятичной системе в шестнадцатеричную систему в 1 байте

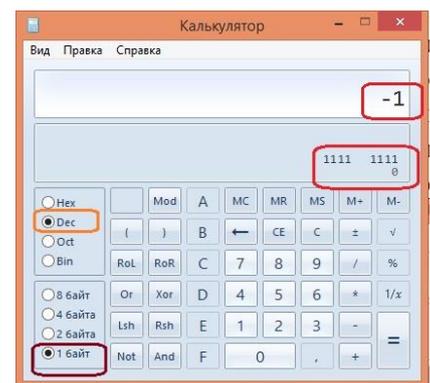


Рис. 39. Представление числа -1 в десятичной системе в двоичную систему в 1 байте

Интересный вариант представляет возможность визуализации десятичного отрицательного числа (например **-1**₁₀) в представлении в двоичной системе. Ниже окна ввода (рис. 39) изображено число **11111111**₂ — это и есть число **-1**₂ в двоичной системе. Причем, старший разряд, как и полагается представляет знак числа (**1** — это минус), а вот мантисса может показаться непривычной: **1111111**₂.

На самом деле это представление отрицательного двоичного числа в дополнительном (или обратном коде). Заметим, что в ЭВМ фактически нет операции вычитания, а есть операция

сложения числа в дополнительном (или обратном коде), что дает результат, эквивалентный операции вычитания. Так, на рисунке 39 это и отображено.

Проверка правильности представления может быть осуществлено следующим образом: нужно выполнить поразрядное сложение числа 1111111_2 и числа 0000001_2 .

1111111_2

0000001_2

0000000_2

4. Следующий вид Калькулятора №1 в режиме *Статистика* представлен на рисунке 40.



Рис. 40. Вид Калькулятора №1 в режиме «Статистика»



Рис. 41. Кнопки Калькулятора в режиме «Статистика»



Рис. 42. Ввод числа 5 (последовательное нажатие кнопок: 5, Add)



Рис. 43. Ввод нескольких чисел: нажатие кнопок: 5, Add, 2, Add, 19, Add



Рис. 44. Получение «Суммы введенных чисел»: нажатие кнопки Σx

В вид *Статистика* (рис. 41) переходим с помощью команды **Вид->Статистика**. Этот режим дает возможность выполнять базовые функции статистики (таблица ниже).

Номер кнопки (цифра или символы)	Изображение кнопки	Функция
1	\bar{x}	Среднее значение
2	$\sqrt{\frac{x^2}{n}}$	Среднеквадратичное значение
3	Σx	Сумма значений
4	Σx^2	Сумма квадратов значений
5	σ_n	Среднеквадратичное отклонение
6	σ_{n-1}	Заполнение среднеквадратичного отклонения
CAD	CAD	Удаление нескольких чисел
Add	Add	Добавление нескольких чисел

Приведем пример вычисления *суммы значений*.

Обратите внимание, что на рисунке 41 в правом верхнем углу окна визуализации нет изображения числа, а в нижнем правом углу окна визуализации изображено число 0. В нижней левой части окна видна надпись «Счет=0»: это означает, что на данный момент не введено ни одного числа.

Вводим первое число и нажимаем кнопку «Добавить» (кнопка «Add» на рис. 42). Справа вверху окна визуализации появляется число 5, изменяется и надпись слева «Счет=1» (введено одно число).

Таким же образом вводим еще два числа, последовательно нажимая кнопки: 2, Add, 19, Add (тем самым к числу 5 добавили число 2 и число 19). Справа вверху окна визуализации (рисунок 43) появляется вертикальная последовательность чисел 5, 2, 19. Изменяется и надпись слева «Счет=3» (введено три числа).

Теперь нажмем кнопку для статистического вычисления суммы введенных чисел: «Сумма значений» Σx (кнопка), которая изображена на рисунке 44 (подсвечивается

желтым цветом) и выделена на рисунке красным прямоугольником. При этом в правом нижнем углу появилась сумма введенных чисел **26: $5+2+19=26$** .

Примеры выполнения расчетов на Калькуляторе: результат заполните сами.

Описание операции	Операция	Нажимаемые клавиши	Результат
Сумма чисел 125 и 17	125 +17	1;2;5;+;1;7;=	
Частное от деления 32 на 4	32 : 4	3;2;/;4;=	
Квадратный корень из 1225	$\sqrt{1225}$	1;2;2;5;sqrt;=	
Обратная величина числа 9	1/9	9;1/x	0,11111111
30% от числа 250	250*0,3	2;5;0;+или-;3;0;%	
5!	5*4*3*2*1	5;! 	
Остаток от деления 57 на 6	(57 mod 6)	5;7;Mod;6;=	
5 в степени 7 , т.е. 5000000	5⁷	5;exp;2;7	5e+07
arcsin(0,5)	arcsin(0,5)	0.5; Inv; sin	
Вычисление корня степени y из числа x	$\sqrt[y]{x}$	23;x^y;(1/8)	1,479

Аналогично примеру вычисления суммы последовательности чисел (набора чисел) можно выполнять любые другие статистические операции в режиме Калькулятора Статистика. В этом режиме можно вводить данные для вычисления статистики, а затем выполнять вычисления:

- ввести первое значение, затем нажать кнопку **Добавить (Add)**;
- затем последовательно добавлять числа из набора данных, нажимая после каждого числа кнопку **Добавить (Add)**;
- нажать кнопку операции статистического вычисления, которое требуется выполнить.

5. Кроме возможностей работы в описанных режимах Калькулятора в любом режиме калькулятора можно вычислить время между двумя датами, а также прибавлять или вычитать дни относительно указанной даты.

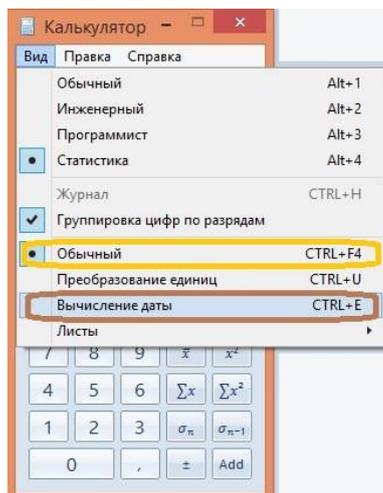


Рис. 45. Пункт меню «Вычисление даты» и пункт меню возврата в обычный режим «Обычный»

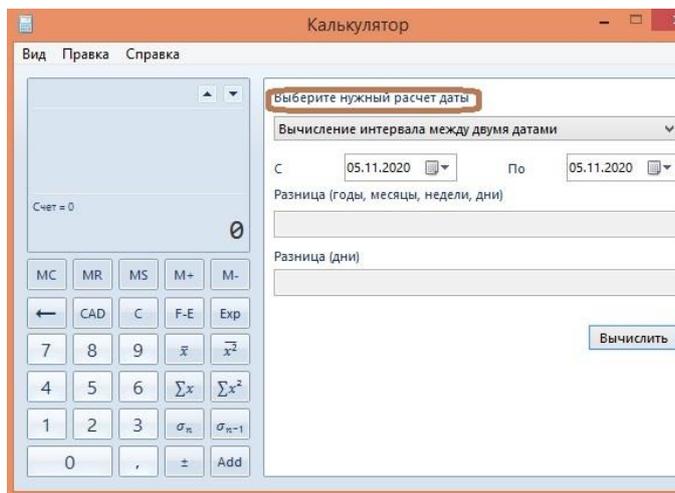


Рис. 46. Выбор пункта меню «Выберите нужный расчет даты»

Для того, чтобы воспользоваться возможностями вычисления времени между датами, следует:

- нажать пункт меню **Вид** в строке вверху, а затем выбрать пункт меню **Вычисление даты (Вид-> Вычисление даты)** (рис. 45);
- в открывшемся справа окне (рис. 46) выбрать пункт **Выберите нужный расчет даты**, затем
- выбрать тип вычисления, которое требуется выполнить:

- «Вычисление интервала между двумя датами» или
- «Добавление или вычитание дней для указанной даты» (рис 47);
- указать значения, с которыми требуется выполнить вычисления (рис. 48), а затем
- нажать кнопку **Вычислить** (рис 47).

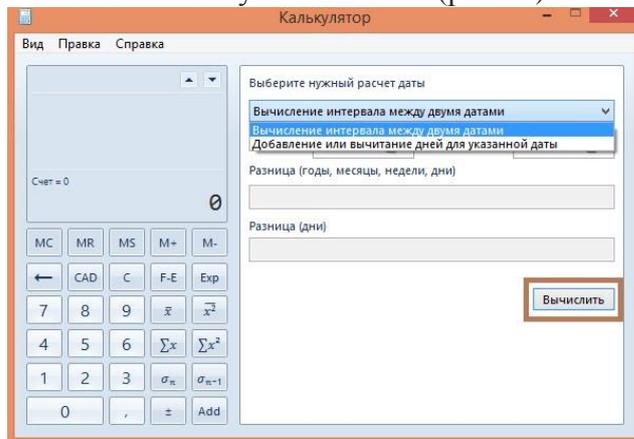


Рис. 47. Выбор пункта меню типа вычисления

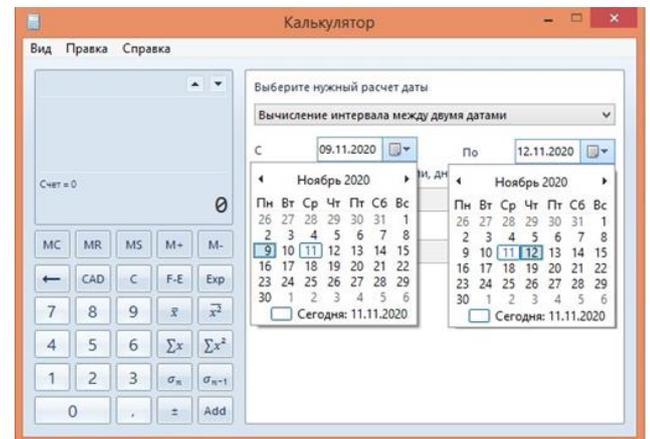


Рис. 48. Выбор значений (дат), с которыми будут производиться вычисления

6. Рассмотрим еще две возможности работы с Калькулятором.

На рисунке 45 в открывающемся окне пункта меню **Вид** можно увидеть еще два интересных пункта: **Преобразование единиц** и **Журнал**.

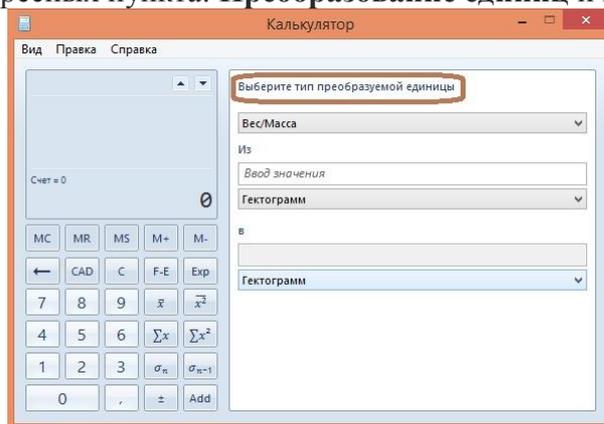


Рис. 49. Окно параметров для пункта меню «Преобразование единиц»

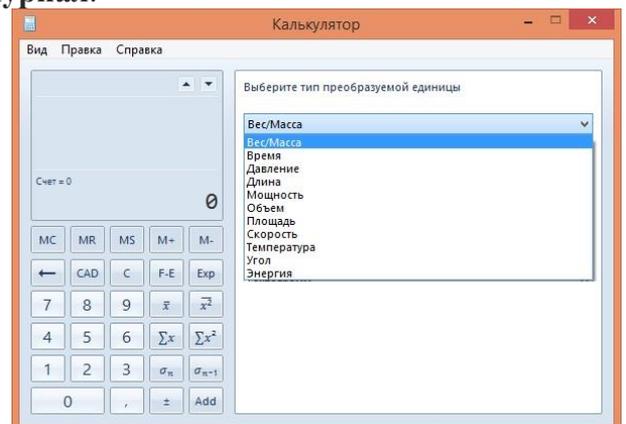


Рис. 50. Варианты типов единиц измерений

Как преобразовать значения из одних единиц измерения в другие?

Для того, чтобы воспользоваться возможностями преобразования значения из одних единиц измерения в другие, следует:

- нажать кнопку **Вид**, а затем **Преобразование единиц (Вид-> Преобразование единиц)** (рис. 45);
- в пункте **Выберите тип преобразуемой единицы** (рис. 49) нажать и выбрать
 - ♦ *тип единицы измерения*, которую необходимо преобразовать (рис. 50),
 - ♦ *исходное и конечное значения* для преобразования (рис. 51, рис. 52);
- в поле **Из** введите значение, которое вы хотите преобразовать (рис. 53).

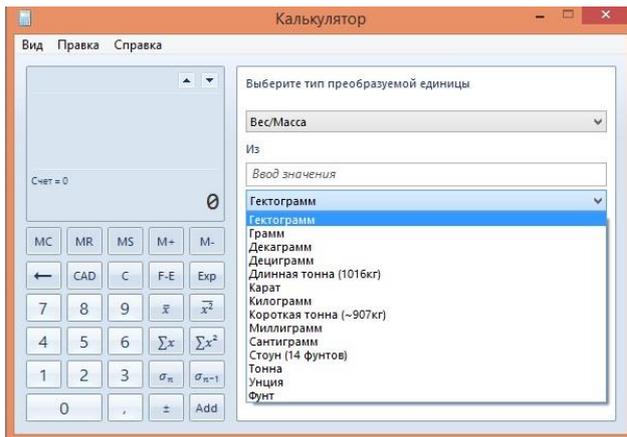


Рис. 52. Виды исходных значений для преобразований

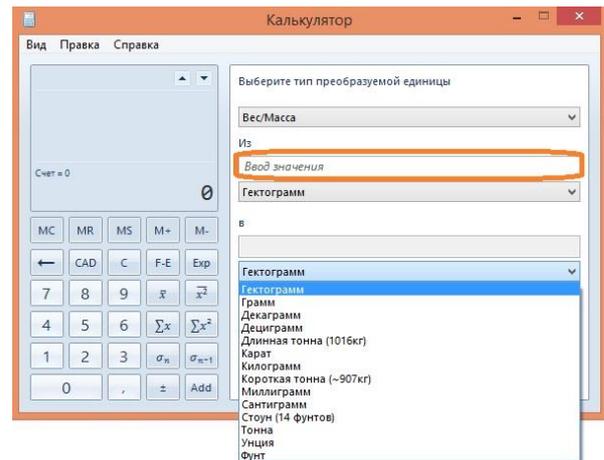


Рис. 53. Виды конечных значений для преобразований и поле **Из** для ввода конкретного преобразуемого значения

Калькулятор имеет еще одну важную возможность — возможность использования *журнала вычислений*. Журнал вычислений — это такое средство, которое сохраняет историю вычислений в продолжение сеанса, то есть со времени начала работы с калькулятором до текущего момента.

Сеанс начинается, когда открывают калькулятор, и заканчивается, когда калькулятор закрывают. Все вычисления, сделанные в течение сеанса, сохраняются в журнале (рис. 54). В правом верхнем визуальном окне демонстрируется список формируемой истории выполнения команд: **78 85/7=10,625**.

Можно изменять значения любых команд в журнале и выполнять новые вычисления (рис. 55). Меняем первую строку: к числу **78** дописываем ***4**. В результате в правом нижнем углу получаем результат **312** измененной команды, то есть, **78*4=312**.

Замечание. Важно отметить, что можно просматривать журнал как стандартного, так и инженерного режима, но не статистики и программиста.

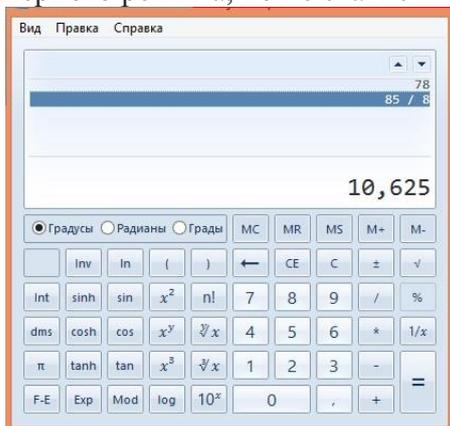


Рис. 54. Демонстрация истории команд в журнале

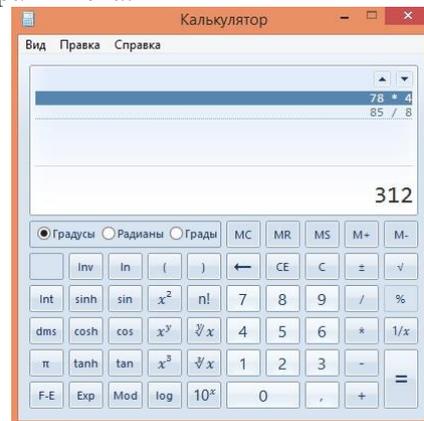


Рис. 55. Изменение произвольной команды из истории команд в журнале

Для того, чтобы воспользоваться возможностями работы с журналом, следует:

- нажать кнопку **Вид**, а затем **Журнал (Вид-> Журнал)**;
- щелкнуть мышкой вычисление, которое нужно изменить;
- ввести изменения в команде или новые значения для вычисления и
- нажать клавишу **ВВОД (Enter)**.

Примечание.

При переключении режимов *текущее* вычисление удаляется.

Журнал вычислений и числа, сохраненные в памяти, не удаляются.

Контрольные вопросы для самопроверки

- 1 Как запустить программу Калькулятор?
- 2 Режимы работы калькулятора, использование памяти
- 3 Как вычислить среднее значение ряда чисел с помощью режима **Статистика**?
- 4 Как вычислить значение корня пятой степени из 3^{12} ?
- 5 Как вычислить $\arctg 0.5$?

Задания 8 (n – номер студента по списку в журнале преподавателя).

1. Вычислить в *обычном* режиме работы калькулятора $(n + 2)\%$ от значения выражения $(3254 - 15 \cdot 25 + 9788 : 11)$.
2. В *инженерном* режиме проделать ту же операцию, соблюдая приоритет выполняемых действий.
3. Найти в *обычном* режиме значение выражения, используя память компьютера:

$$\frac{(5,9 + 3227) : (2100 - 0,5)}{3,8 + 9,1 + 2,9} * n;$$

4. Вычислить значение выражения в *инженерном* режиме:

$$\frac{155^2 + 33^3 - 2^8 + 11^{12} - 3^4}{\sqrt[3]{1/53^2 + 14^3 - 1/4^6}} * n^2$$

5. Найти среднее арифметическое от заданных чисел в *обычном* и *инженерном* режимах:

$$31550 + 2163 + 154 + 3 + n$$

6. Вычислить в *инженерном* режиме: $n \cdot \arcsin \frac{\sqrt{3}}{2}$

7. Вычислить в *инженерном* режиме $n \cdot \arctg 1$, $\arccos \frac{\sqrt{2}}{2}$

8. Найти сумму чисел из числителя задания 4 в *инженерном* и статистическом режиме.

9. Вычислить в *инженерном* режиме $1/n \cdot \log 81$; $1/n \cdot \log 0,001$; $(2+n)!$; $\sin 1873 \cdot n$.

Практическое занятие № 2. Paint

Приемы работы с графическим редактором Paint

1. Запустите графический редактор **Paint** одним из способов (**Пуск**→**Программы**→**Стандартные**→ **Paint** в Windows 8). Либо один из видов приложения **Paint** (**Пуск** → **Найти**→ **Paint**). Изображения представлены в Microsoft Office 2010 (рис. 56)

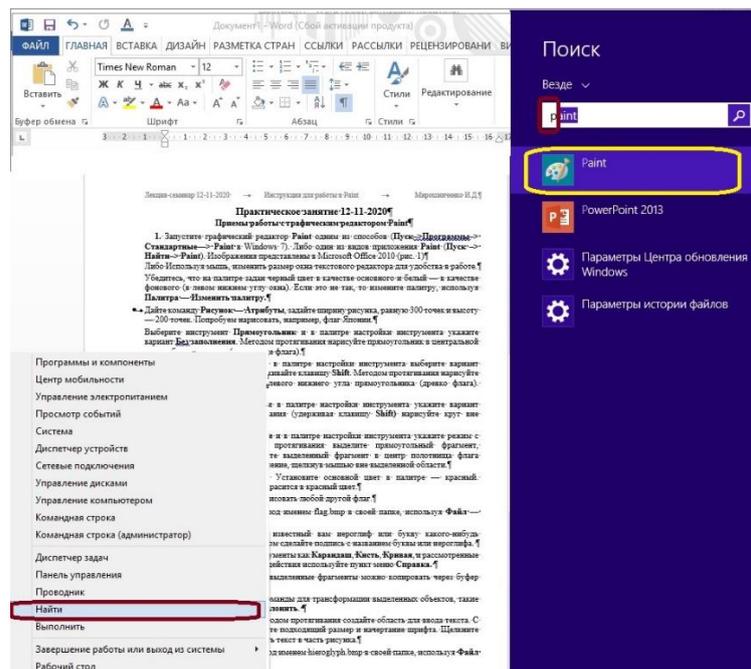


Рис. 56. Открытие приложения **Paint**

Откроется окно приложения **Paint** (рис.57).

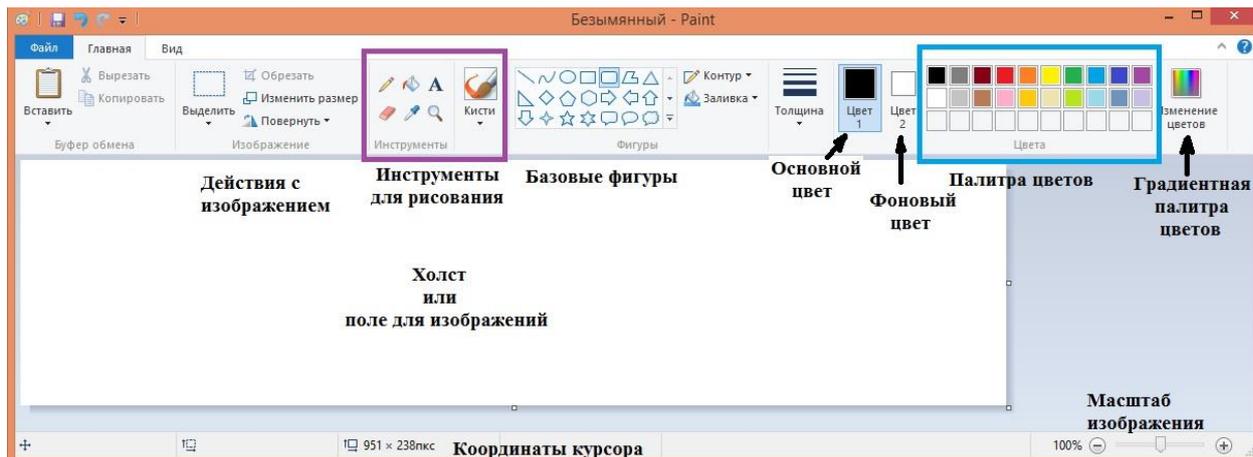


Рис. 57. Составляющие элементы окна **Paint**

2. Используя мышь, измените размер окна приложения **Paint** для удобства в работе. Убедитесь, что на палитре задан черный цвет в качестве основного и белый — в качестве фонового (на главной панели окна). Если это не так, то измените палитру основного или фонового цветов. Для этого щелкните по нужному цвету (основному или фоновому), а потом щелкните по нужному цвету палитры цветов. Например, если надо изменить основной цвет, то сначала надо щелкнуть по квадратику, который представляет **Основной цвет**, а потом щелкнуть черный квадратик из **Палитры цветов**. Квадрат Основного цвета окрасится в черный цвет.

3. Создайте новый рисунок. Для этого нужно выбрать пункт **Файл** основной панели, откроется выпадающее меню, в котором выбрать **Создать (Файл-> Создать)** (рис. 58).

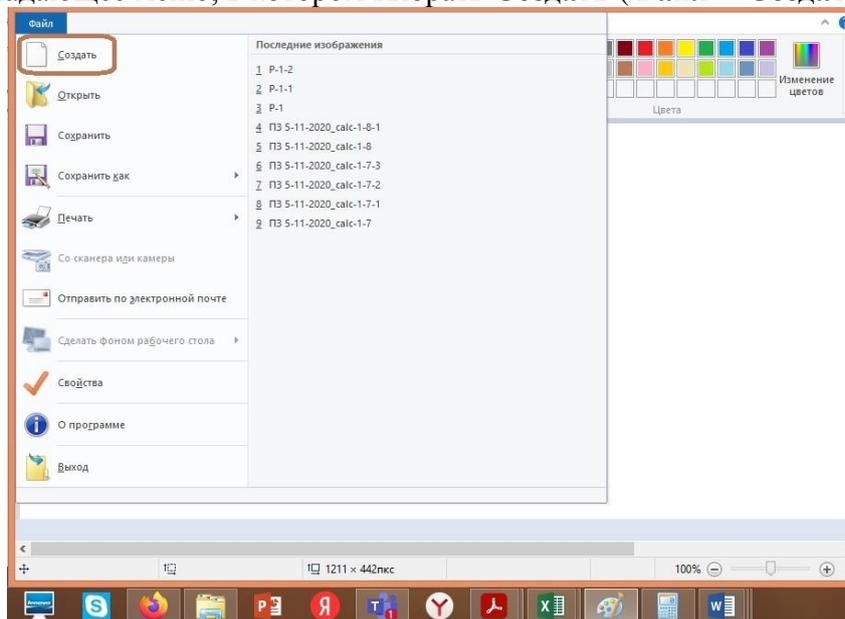


Рис. 58. Выпадающее меню пункта **Файл**

Получим окно **Paint** с чистым холстом как на рисунке 57.

4. Попробуйте нарисовать, например, флаг Японии. Выберите инструмент **Прямоугольник** из базовых фигур (см. рис. 57) и в квадратике фонового цвета оставьте белый цвет. Методом протягивания нарисуйте прямоугольник в центральной части области рисунка (полотнище флага) (рис. 59).

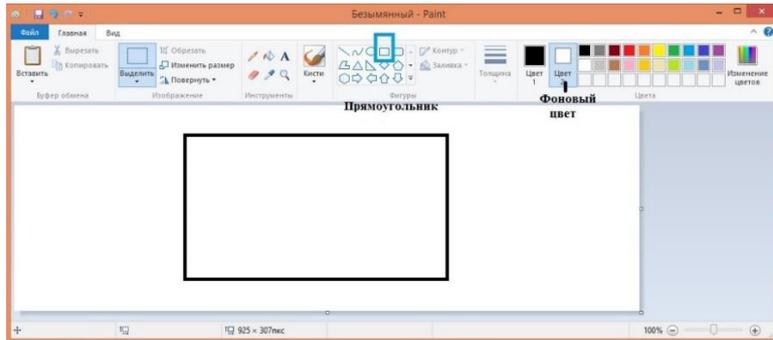


Рис. 59. Нарисовали контур полотнища флага.

5. Выберите инструмент **Линия** и в палитре настройки инструмента выберите вариант толщины линии. Нажмите и удерживайте клавишу **Shift** или нажмите и удерживайте левую часть движка. Методом протягивания нарисуйте вертикальный отрезок вниз от левого нижнего угла прямоугольника (древко флага) (рис. 60). Отпустите клавишу **Shift** или движок.

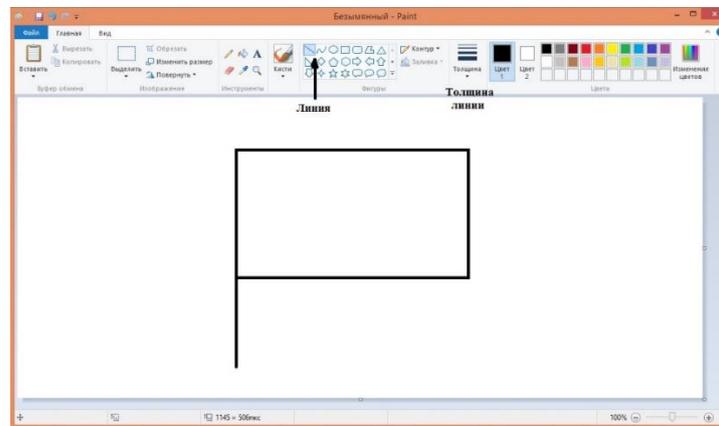


Рис. 60. Рисование древка флага

6. Выберите инструмент **Эллипс** и в палитре настройки инструмента укажите основной цвет красный. Методом протягивания (удерживая клавишу **Shift** или левую часть движка) нарисуйте окружность вне области полотнища флага и в центре полотнища флага (рис. 61).

Выберите инструмент **Заливка (ведерко)** и залейте внутреннюю часть окружности в центре полотнища флага (рис. 61). Получим красный круг в центре полотнища флага.

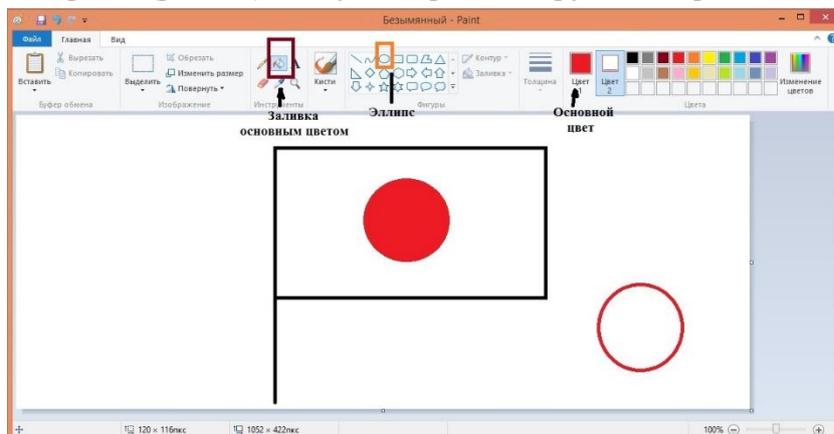


Рис 61. Рисование окружности и круга с заливкой

Уберем окружность вне флага. Выберем инструмент **Выделение** прямоугольной области и методом протягивания ограничим окружность вне флага выделением в виде пунктирного прямоугольника (рис.62).

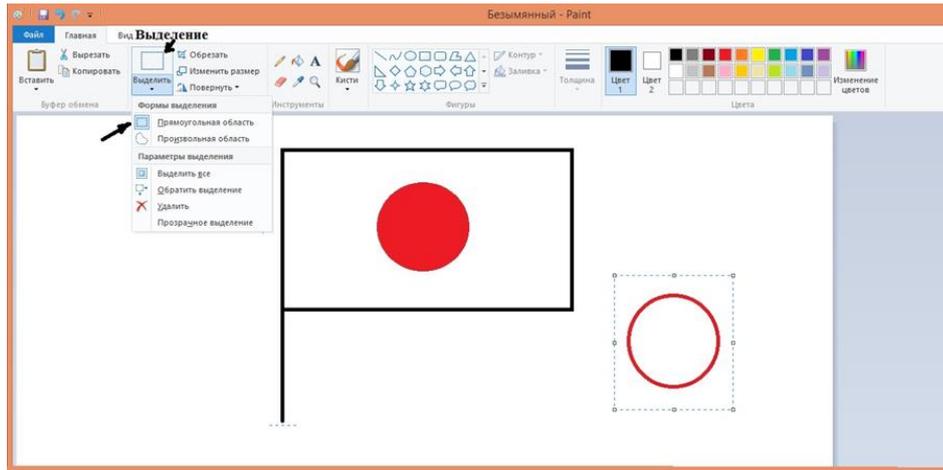


Рис. 62. Выделение окружности вне флага.

Нажмем клавишу **Удалить (Delete)**. Окружность исчезнет (рис. 63).

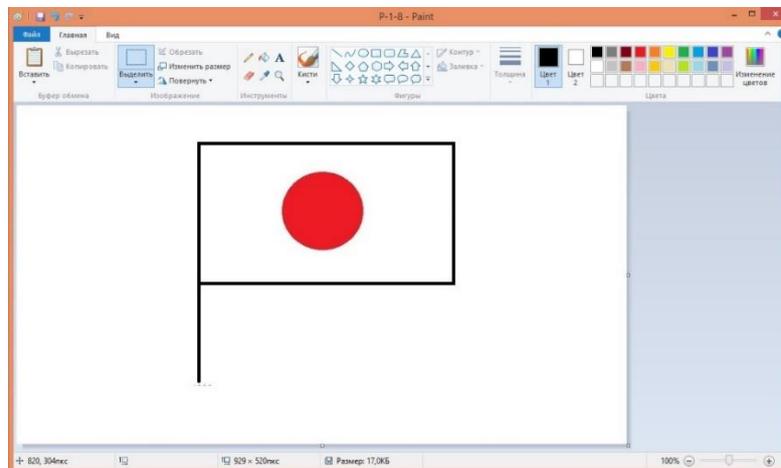


Рис. 63. Окончательный рисунок флага.

Аналогичным образом можно нарисовать любой другой флаг.

7. Сохраните созданный документ под именем **flag.jpg** в своей папке (рис. 64), используя действия **Файл** → **Сохранить как** (пункт а). Этот пункт находится в выпадающем меню пункта **Файл** основной панели (см рис. 58).

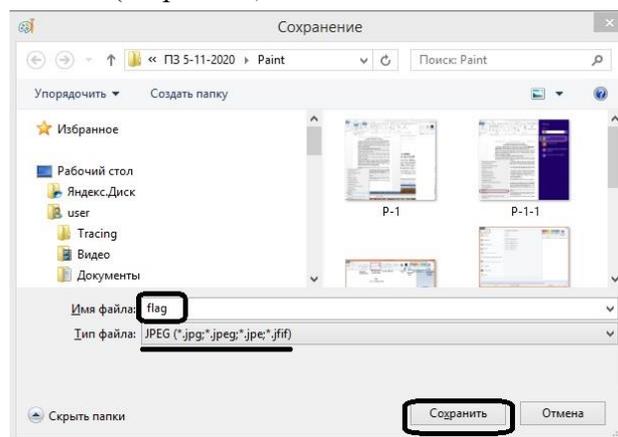


Рис. 64. Сохранение файла.

8. Сохраните созданный документ (рисунок) также под именем **flag.bmp** в своей папке, используя **Файл** → **Сохранить как** и выбрав из списка расширений расширение **bmp** нужного типа:

- Монохромный рисунок (*.bmp)
- 16-цветный рисунок (*.bmp)
- 256-цветный рисунок (*.bmp)
- 24 разрядный рисунок (*.bmp)

Задание 1. Выполните описанные действия и представьте файлы с именами *flag.jpg* и *flag.bmp*.

Задание 2. Нарисуйте любой известный вам символ или букву какого-нибудь алфавита. Под рисунком сделайте подпись с названием буквы или символа. Надпись выполняется с помощью инструмента текст **A** на главной панели (см. рис. 65).

Выберите инструмент **Текст (A)**. Методом протягивания создайте прямоугольную область для ввода текста. Как только определяется область для написания текста появляется панель **Шрифтов С** помощью панели **Шрифты** можно задать подходящий размер и начертание шрифта. Щелкните вне ввода текста, чтобы превратить текст в часть рисунка.

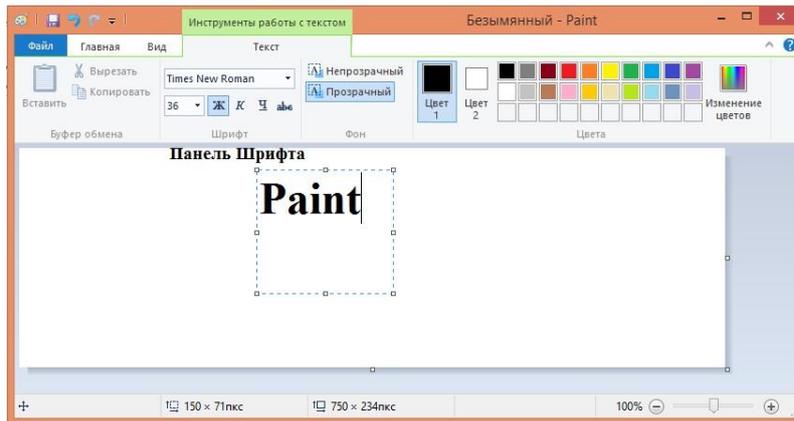


Рис. 65. Использование панели Шрифт и инструмента Текст.

Можете использовать такие инструменты как **Карандаш**, **Кисть**, **Кривая**, и рассмотренные ранее. Для уточнения способа их действия используйте пункт меню **Справка**.

Учтите, что, как обычно, выделенные фрагменты можно копировать через буфер обмена.

9. На главной панели рядом с инструментом **Выделение** можно найти кнопки для трансформации выделенных объектов, такие как **Изменить размер**, **Обрезать**, **Повернуть**. Сохраните созданный документ под именем *hieroglyph.bmp* в своей папке, используя **Файл** → **Сохранить как** (рис. 64).

Практическое занятие № 3. Блокнот

Приемы работы с текстовым редактором Блокнот

1. Запустите текстовый редактор **Блокнот** (**Пуск** → **Программы** → **Стандартные** → **Блокнот**).

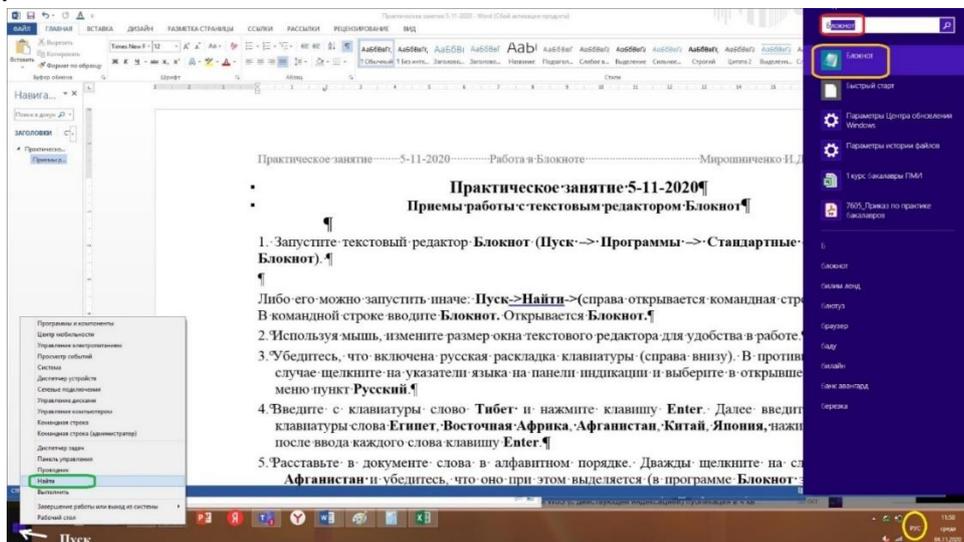


Рис. 66. Вызов приложения Блокнот

Либо его можно запустить иначе: **Пуск->Найти->**(справа открывается командная строка). В командной строке вводите **Блокнот** (или **Calc**). Открывается **Блокнот**.

2. Используя мышь, измените размер окна текстового редактора для удобства в работе (рис. 67).

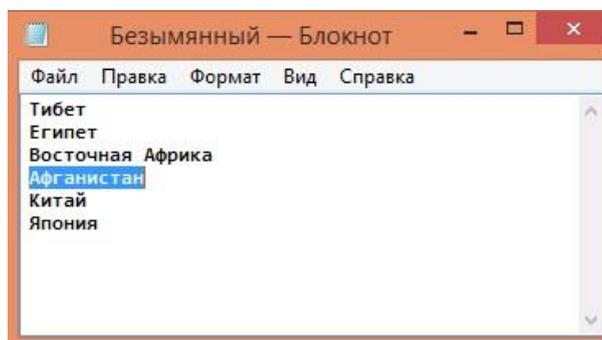


Рис. 67. Выполнение пунктов 2-3-4

3. Убедитесь, что включена русская раскладка клавиатуры  (справа внизу рисунок 66). В противном случае щелкните на указатели языка на панели индикации и выберите в открывшемся меню пункт **Русский**.

4. Введите с клавиатуры слово **Тибет** и нажмите клавишу **Enter**. Далее введите с клавиатуры слова **Египет**, **Восточная Африка**, **Афганистан**, **Китай**, **Япония**, нажимая после ввода каждого слова клавишу **Enter**.

5. Расставьте в документе слова в алфавитном порядке. Дважды щелкните на слове **Афганистан** и убедитесь, что оно при этом выделяется (в программе **Блокнот** этот способ служит для выделения отдельных слов). Нажмите комбинацию клавиш **Shift+Вправо**, чтобы включить в выделенный фрагмент невидимый символ конца строки — курсор при этом переместится в начало следующей строки. (не во всех версиях Windows)

6. а) Выполните команду **Правка — Вырезать**, чтобы забрать выделенный фрагмент в буфер обмена. Убедитесь, что он действительно удаляется из документа.

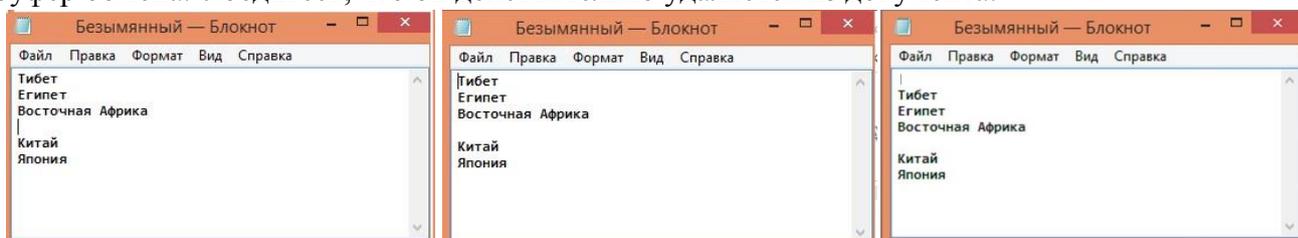


Рис.68 Выполнение пунктов 6.а-б

б) Нажмите комбинацию клавиш **Ctrl+Home**, чтобы установить курсор в начало документа. Затем нажмите **Enter**, чтобы вставить пустую строку.

в) Выполните команду **Правка — Вставить**, чтобы вставить фрагмент из буфера обмена.

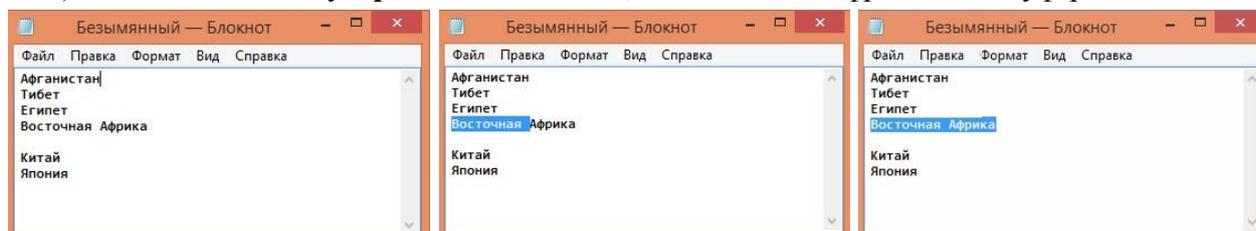


Рис. 69. Выполнение пунктов 6.в-г-д

г) Установите текстовый курсор в начало строки, содержащей слова **Восточная Африка**. Дважды нажмите комбинацию **Shift+Ctrl+Вправо** и убедитесь, что при каждом нажатии выделенный фрагмент расширяется, охватывая следующее слово.

д) Нажмите комбинацию клавиш **Shift+Вправо**. Вы выделили нужный фрагмент при помощи клавиатурных команд.

7. а) Нажмите комбинацию клавиш **Shift+Delete**, переместите текстовый курсор в начало второй строки документа.

б) Вставьте новый фрагмент из буфера обмена с помощью комбинации клавиш **Shift+Insert**.

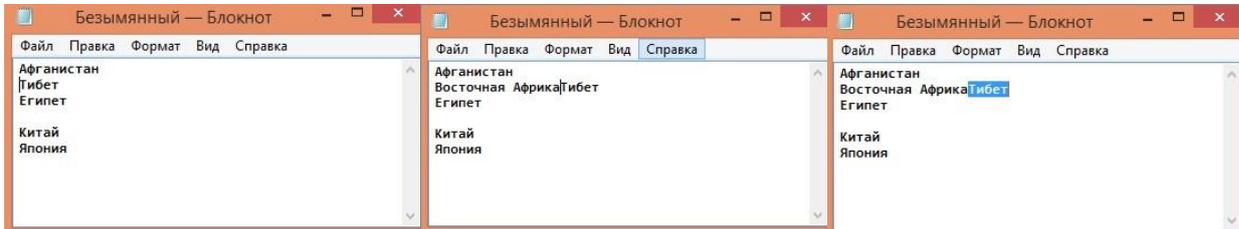


Рис. 70. Выполнение пунктов 7.а-б

б) Установите указатель мыши на начало слова **Тибет**. Нажмите левую кнопку мыши и, не отпуская ее, выделите слово методом протягивания.

в) Нажмите комбинацию клавиш **Ctrl+X**, переместите текстовый курсор в конец предпоследней строки документа и вставьте новый фрагмент из буфера обмена с помощью комбинации клавиш **Ctrl+V**. Разделите строку на две строки, используя клавишу **Enter**.

г) Установите курсор на начало пустой строки и нажмите клавишу **Backspace**.

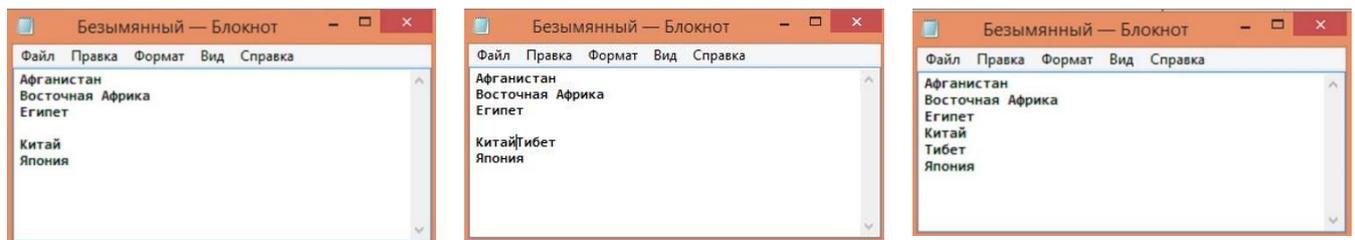


Рис. 71. Выполнение пунктов в-г

В результате должен получиться список слов или словосочетаний в алфавитном порядке, каждое слово и словосочетание (Восточная Африка) должны стоять на отдельной строке.

д) Сохраните созданный документ под именем *list.txt* в своей папке, используя команду **Файл** → **Сохранить как**.

8. Поиск определенных символов или слов (рис. 72).

В меню **Правка** выберите команду **Найти**. В поле **Образец** (или **Что**) введите любое слово текста, который требуется найти. Нажмите кнопку **Найти далее**. Убедитесь, что нужное слово найдено.

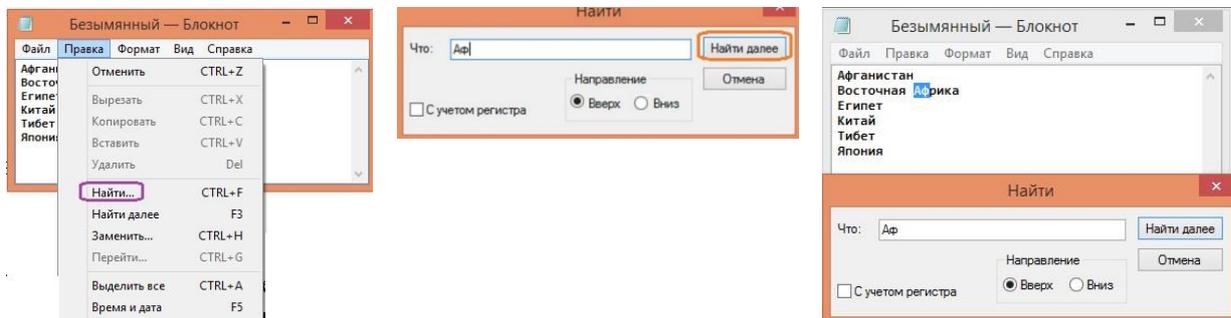


Рис. 72. Выполнение команды **Найти по образцу**

9. Вставка в документ даты и времени (рис. 73).

Поместите курсор в место вставки, например, в конец документа. В меню **Правка** выберите команду **Дата/время**. Убедитесь, что в документе появилась текущая дата и текущее время.

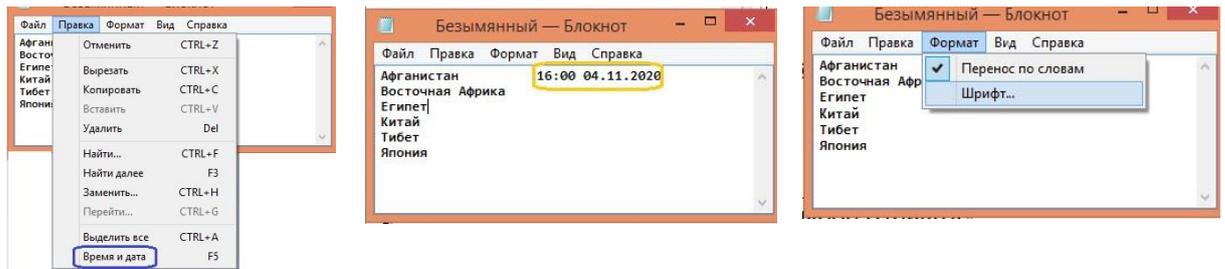


Рис. 73. Выполнение команд Вставка времени и даты, выбор команды Шрифт

10. Выбор шрифта (рис. 74).

После выполнения команд **Формат** → **Шрифт** в открывшемся диалоговом окне **Шрифт** можно выбрать шрифт и **Начертание**, а также **Размер**, используемый для всего текста документа. Выбрать, например, курсивное начертание, шрифт *Tahoma*, размер 12 pt. Посмотрите, как будет выглядеть текст в поле **Образец**.

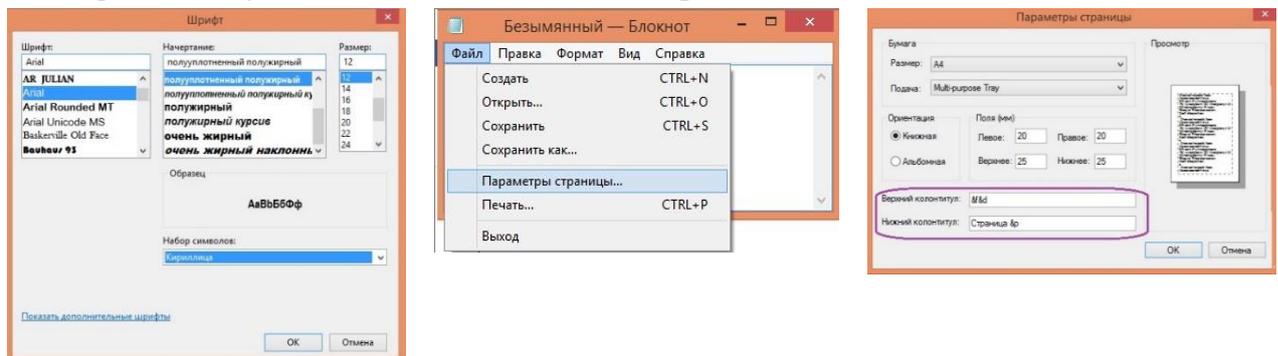


Рис. 74. Выполнение команд выбора начертания и размера Шрифта

11. Создание колонтитулов при печати документа.

В меню **Файл** выберите команду **Макет страницы** (или **Параметры страницы**). В поле **Верхний колонтитул** или **Нижний колонтитул** введите сочетание букв или символов из следующей таблицы:

Действие	Параметр
Вставка имени открываемого файла или вывод "(Безымянный)", если имя файлу не присвоено.	&f
Вставка даты.	&d
Вставка времени по часам компьютера.	&t
Вставка номеров страниц.	&p
Выровнять верхний колонтитул по левому краю, центру или правому краю	&l, &c или &r

12. Сохраните (замените) измененный документ под именем *list.txt* в своей папке, используя команду **Файл** → **Сохранить как**.

Примечания:

1. В полях **верхнего** и **нижнего** колонтитула можно вводить несколько сочетаний символов. Оставьте один или два пробела между сочетаниями, чтобы верхний и нижний колонтитулы легче читались.

2. Допускается также использование дополнительного текста в комбинации с параметрами. Например, верхний колонтитул "Страница &p" задает печать заголовков "Страница 1" и так далее.

3. Созданные колонтитулы видны только при печати на принтере.

Задание

1. Выполнить практическую работу согласно инструкции.
2. Придумать и выполнить свой вариант работы в Блокноте, добавляя изображения шагов работы.

Практическое занятие № 4. WordPad

Приемы форматирования, внедрения и связывания объектов в текстовом процессоре WordPad

1. Запустите текстовый процессор WordPad (Пуск → Программы → Стандартные → WordPad). Или Пуск->Найти -> WordPad

Используя мышь, измените размер окна текстового редактора для удобства в работе.

2. Откройте ранее созданный в Блокноте текстовый файл *list.txt* (рис. 75).

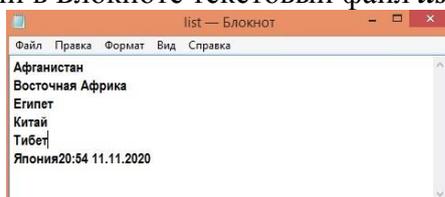


Рис. 75. Содержимое файла, созданного в Блокноте

Введите команду **Файл** → **Сохранить как**, в списке **Типов файла** выберите пункт **Word (rtf)** и сохраните файл под именем *dict.rtf* (рис. 76).

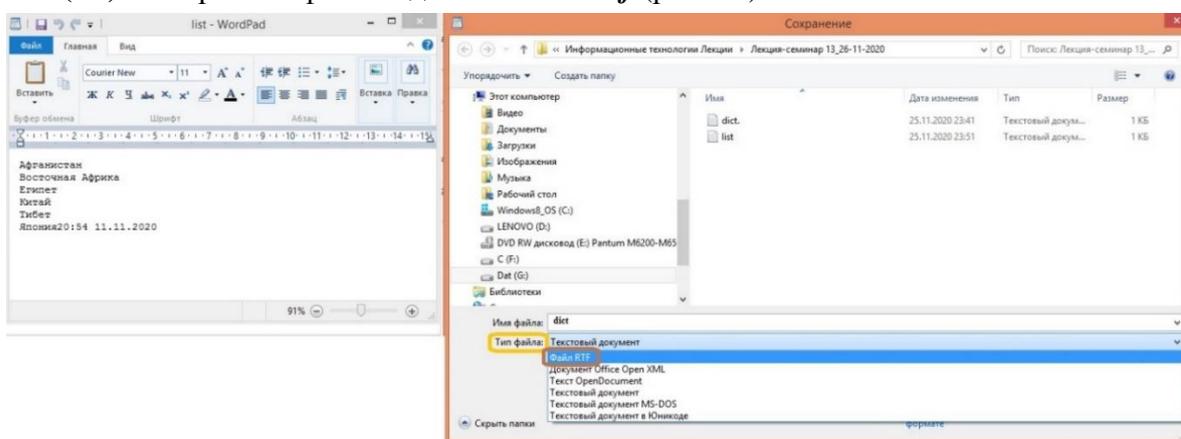


Рис. 76. Сохранение файла с расширением rtf

3. Выделите первое слово документа (Афганистан).

На панели форматирования задайте шрифт **Arial** (Кириллица), размер — 14 пунктов, выберите полужирное начертание (рис. 77).

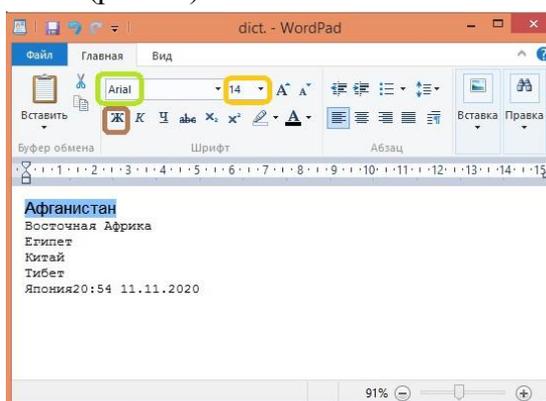


Рис. 77. Вид текста файла с расширением *rtf*

Нажмите клавишу **Enter**, чтобы убрать выделение. Затем снова нажмите **Enter**, чтобы вставить пустую строку.

4. На новой пустой строке запишите краткое описание слова, указанного в предыдущей строке, например, так: «страна в Азии со столицей — городом Кабулом». Размножьте введенный текст (**CTRL_C**, **CTRL_V**), чтобы образовался абзац размером 3–4 строки (можно вместо размножения текста дать более подробное описание)

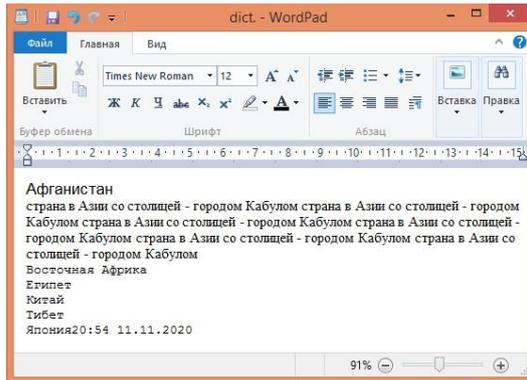


Рис. 78. Размножение строки «страна в Азии со столицей - городом Кабулом»

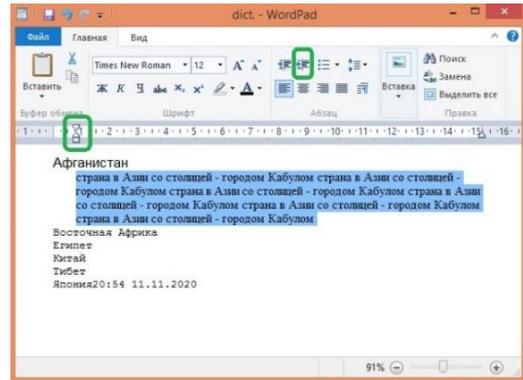


Рис. 79. Сдвиг вправо на 1 см выделенного текста: сначала на табуляцию(верхний ярлычок, затем по линейке на 1 см

5. Выделите весь только что введенный абзац (**Shift** ↓, **Shift** →, **Shift** ←). На панели форматирования задайте шрифт **Times New Roman** (Кириллица), размер — 12 пунктов, выберите обычное начертание.

На линейке, расположенной ниже панели форматирования, перетащите маркер в виде ярлычка на расстояние 1 см (по линейке) вправо (или влево в зависимости от отступа табуляции). Убедитесь, что весь абзац теперь отображается с отступом от левого края.

6. Снимите выделение, нажав **Enter** в конце выделенного абзаца, установите курсор в начало первой строки того же абзаца. Нажмите клавишу **Tab** для создания абзацного отступа (способ выделения красной строки).

7. Введите аналогичные описания для последующих слов и отформатируйте все так, как описано выше.

8. Установите курсор в конце описания слова **Япония** и нажмите клавишу **Enter**. Введите команду **Вставка** → **Объект**.

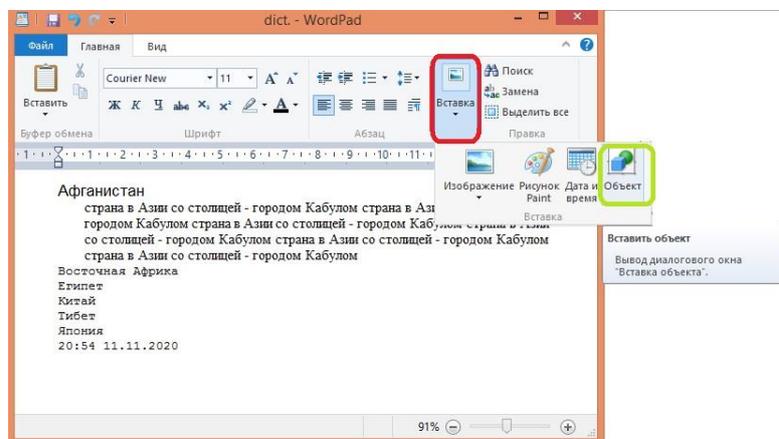


Рис. 80. Вставка текста и даты

9. В диалоговом окне **Вставка объекта** установите переключатель **Создать из файла**.

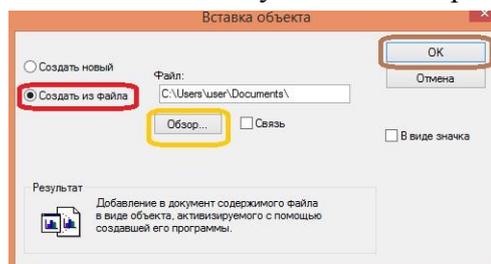


Рис. 81. Вставить объект в файл в WordPad

10. Щелкните на кнопке **Обзор**, разыщите в файловой структуре ранее созданный документ **flag.jpg**, щелкните на кнопке **Ok**. Убедитесь, что созданное изображение флага

Японии вставлено в документ в качестве иллюстрации. Для этого щелкните по вставленной стилизации изображения (появится изображение флага в размере окна просмотра рисунков).

11. Если был создан флаг другой страны, то внесите название страны и ее краткое описание в файл, следуя алфавитному порядку. Вставьте флаг в соответствующее место.

12. Измените масштаб отображения рисунка в документе путем перетаскивания маркеров изменения размера, расположенных на границах объекта.

13. Сохраните созданный документ *dict.doc* в своей папке, используя **Файл** → **Сохранить как**.

14. **Настройка параметров печатной страницы.** Параметры страницы задают в диалоговом окне **Параметры страницы** (**Файл** → **Параметры страницы**). Задайте ориентацию и поля.

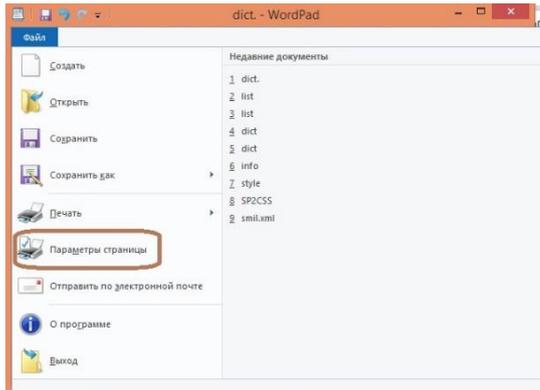


Рис. 82. Настройка параметров страницы



Рис. 83. Ориентация поля

15. **Настройка параметров абзаца.** Выполняется в диалоговом окне **Абзац**, открываемом командой **Формат** → **Абзац**.

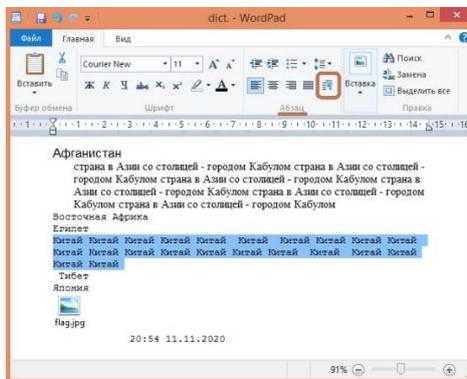


Рис. 84. Вид текста после вставки объекта (рисунка)

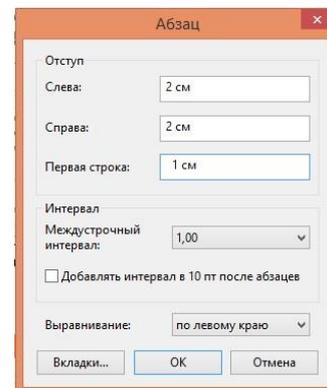


Рис. 85. Настройка параметров абзаца

Поэкспериментируйте с форматированием абзацев.

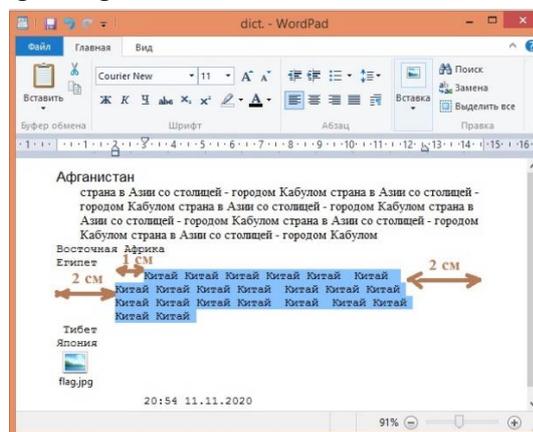


Рис. 86. Разметка текста после форматирования абзаца

16. **Создание маркированных списков.** Создайте небольшой маркированный список, состоящий из столиц восточных стран. Первая строка маркированного списка создается командой (**Формат** → **Маркер**). Последующие строки автоматически получают маркер после нажатия клавиши **Enter**. Для прекращения маркировки надо просто повторить команду еще раз.

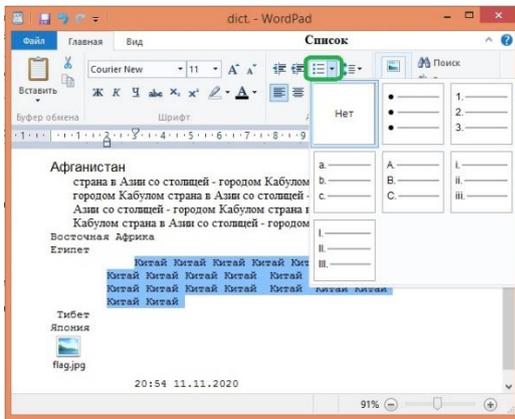


Рис. 87. Создание маркированного списка



Рис. 88. Поиск и замена образца в документе

17. **Поиск и замена текстовых фрагментов.** Используя команду **Правка** → **Заменить**, можно найти и заменить слова в тексте по образцу. Поэкспериментируйте с поиском и заменой слов.

18. **Настройка печати документа перед выводом на печать.** В меню **Файл** выберите команду **Печать**. Чтобы вернуться в обычный режим из окна **Печать**, нажмите кнопку **Отмена**.

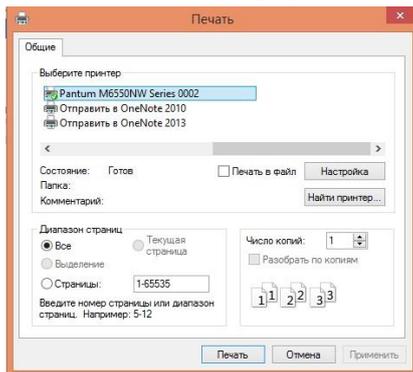


Рис. 89. Настройка принтера для печати

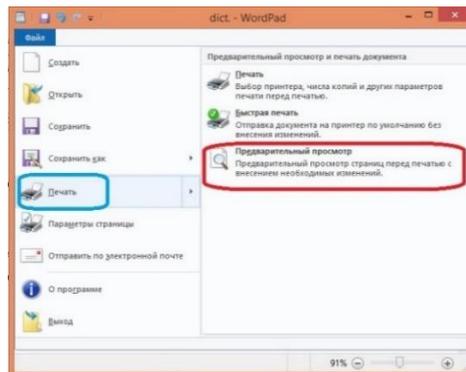


Рис. 90. Настройка предварительного просмотра

19. **Предварительный просмотр документа перед выводом на печать.** В меню **Файл** выберите команду **Предварительный просмотр**. Чтобы вернуться в обычный режим из окна **Предварительный просмотр** (рис. ниже), нажмите кнопку **Закреть**.

20. Сохранить документ с именем *dict* и расширением *rtf*.

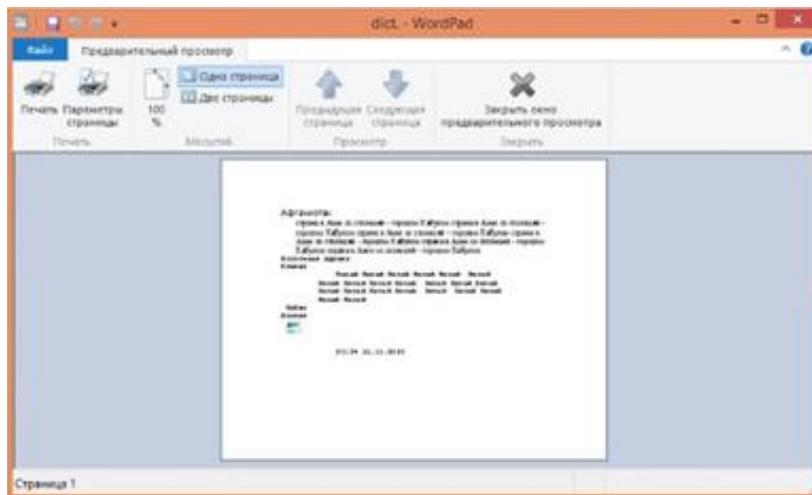


Рис. 91. Вид документа при предварительном просмотре

21. **Удаление и замена внедренного рисунка.** Если файл *dict.rtf* закрыт, откройте его заново. Сохраните файл *dict.rtf* под именем *dict1.rtf*.

В файле *dict1.rtf* удалите внедренный рисунок. Для этого выделите его изображение щелчком мыши и нажмите клавишу **Delete**.

Для вставки другого рисунка выполните команду **Вставка — Объект**. В диалоговом окне **Вставка** объекта установите переключатель **Создать из файла**.

22. Щелкните на кнопке **Обзор**, разыщите в файловой структуре ранее созданный рисунок *flag.bmp*, щелкните на кнопке **Вставить**. Установите флажок **Связь**, чтобы установить связь с рисунком (операция связывания) на странице. Щелкните на кнопке **Ок**. Сохраните документ под именем *dict1.rtf*, выполнив команду **Файл —> Сохранить (Файл —> Сохранить как**, а затем подтвердить замену).

23. Закройте программу **WordPad**.

24. **Изменение рисунка.** Сохраните файл *dict1.rtf* под именем *dict2.rtf*.

25. Откройте файл *flag.bmp* и измените его, например, сотрите древко флага с помощью инструмента **Ластик** или измените вид полотнища (сделайте его в виде елочного флажка с угловым вырезом справа). Сохраните рисунок под тем же именем *flag.bmp*.

26. Запустите текстовый процессор **WordPad (Пуск —> Программы —> Стандартные —> WordPad)**. Откройте файл *dict2.rtf*. Проверьте, изменился ли его вид. Закройте документ *dict2.rtf*. Откройте документ *dict1.rtf*. Обратите внимание на вспомогательную операцию, выполняемую после загрузки документа. Проверьте, изменился ли вид этого документа.

Задание. Выполнить все предложенные пункты практической работы, получить и предоставить для проверки все указанные в работе файлы: *dict.rtf*, *dict1.rtf*, *dict2.rtf*, *flag.jpg*, *flag.bmp* (до изменения), *flag.bmp* (после изменения), необходимые скриншоты.

Практическое занятие № 5. MS Word

Создание и сохранение документов в текстовом процессоре MS Word

1. Запустите текстовый процессор **MS Word (Пуск —> Программы —> Microsoft Word)**. Или **Пуск-> Найти-> Word**. Используя мышь, изменить размер окна текстового редактора для удобства в работе.

2. Наберите текст любой загадки. Сохраните набранный текст в файле в папке *Zagadki* под именем *zagadka1.doc*.

Для этого последовательно перейдите по пунктам (**Файл —> Сохранить как**). В появившемся диалоговом окне выберите Ваш рабочий диск и выберите свою рабочую папку. Затем нажмите кнопку **Создать новую папку**, дайте ей имя *Zagadki*, откройте ее, дайте файлу имя *zagadka1* с расширением *doc* (*zagadka1.doc*), после чего нажмите **Enter** или кнопку **Сохранить**.

3. Создайте в этой же папке новый документ, не закрывая предыдущего (**Файл —> Создать —> Новый документ**).

4. Наберите текст другой загадки. Сохраните текст в этой же папке в файле под именем *zagadka2.doc*.

5. Создайте аналогичным образом еще два файла с отгадками для подготовленных загадок под названиями *otgadka1.doc* и *otgadka2.doc*.

6. Выбрав пункт меню **Вид->Новое окно**, перейдите к документу *zagadka1.doc* и наберите после текста загадки на следующей строке вопрос: «*Отгадайте, что это?*».

7. Выполните аналогичные действия для документа *zagadka2.doc*. Сохраните (**Сохранить**) и закройте все документы.

8. Откройте файл *otgadka1.doc*. Допишите в первую строку (перед существующим текстом): «Верный ответ». Сохраните документ под *новым* именем *otgadka.doc* в папке *Zagadki* (**Файл** → **Сохранить как**).

9. Создайте файл, содержащий текст какого-нибудь стихотворения или песни. Файл начните с *названия* данного произведения.

При наборе текста полезны две пользовательские кнопки **Отменить** и **Вернуть**. Найдите их и посмотрите принцип их действия.

Сделайте заголовок полужирным шрифтом размером 20 пунктов, гарнитуру выберите Comic Sans MS. Основной текст — размер (кегель) 14, начертание - курсив, шрифт (гарнитура) *Arial*.

10. Настройте параметры страницы (**Файл** → **Параметры страницы**). На вкладке **Поля** установите величину верхнего и нижнего поля — 1,5 см; левого — 2,5 см; правого — 1 см. На вкладке **Размер бумаги** посмотрите формат страницы (по умолчанию А4) и ориентацию страницы (книжная).

11. Выполните **Файл** — **Предварительный просмотр**, посмотрите, что получилось. Выйдите из режима предварительного просмотра, нажав кнопку **Заккрыть**.

12. Сохраните файл под именем *poem.doc* в своей основной папке.

13. Создайте новый файл, содержащий текст заявления.

Например:

Преподавателю информатики
Мирошниченко Ирине Дмитриевне
от студента Иванова Ивана

Заявление

Я, Иванов Иван, успешно осваиваю компьютерную грамотность. Хочу узнать еще больше нового и интересного.

22 октября 2022 года.

При наборе текста используйте шрифт различного начертания и размера.

Используйте кнопки абзацного выравнивания **По левому краю**, **По центру**, **По правому краю** и **По ширине**. Найдите их на панели инструментов и посмотрите принцип их действия.

Выполните **Предварительный просмотр**, нажав на **Enter**. Посмотрите, что получилось, а затем вернитесь в исходное состояние, если не понравился результат. Попробуйте разные варианты, остановитесь на понравившемся.

14. Сохраните файл под именем **Zayavlenie** (с расширением *doc*) в своей папке (**Сохранить как**).

15. Закройте все открытые файлы и закройте текстовый процессор MS Word.

Задание. Выполнить все предложенные пункты практической работы, получить и предоставить для проверки все указанные в работе файлы: в папке *Zagadki* должны находиться файлы *zagadka1.doc*, *zagadka2.doc*, *otgadka1.doc*, *otgadka2.doc*, *otgadka.doc*, — файлы *poem.doc* и *Zayavlenie.doc*.

Практическое занятие № 6. Форматирование в MS Word

Шрифтовое и абзацное форматирование в текстовом процессоре MS Word

1. Запустите текстовый процессор MS Word (Пуск – Программы – Microsoft Word). Или Пуск->Найти -> Word. Используя мышь, изменить размер окна текстового редактора для удобства в работе.

2. Откройте файл «**poem**». Задайте следующее шрифтовое форматирование.

Название: размер шрифта (кегель) 20 пт, полужирный, шрифт Comic Sans.

Текст: размер шрифта (кегель) 14, курсив, шрифт Arial.

Межстрочное расстояние: 1,5 строки.

Пример применения параметров форматирования шрифта представлено на рисунке 88.

Выделите **написание** любого слова в тексте стихотворения цветом. Также возможно изменить его *начертание* и сделать подчеркивание узорной линией или двумя чертами (по выбору). Например, ёлочка.

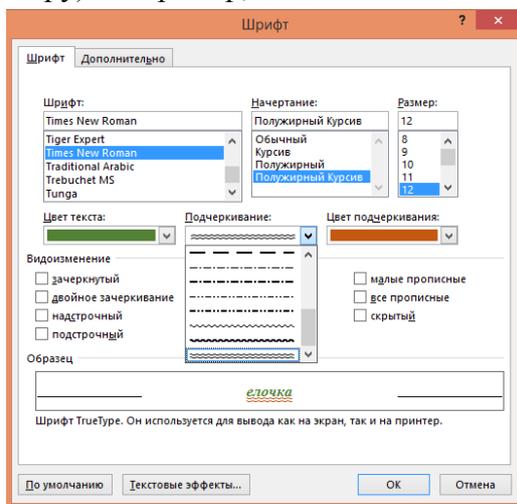


Рис. 92. Выбор параметров шрифта

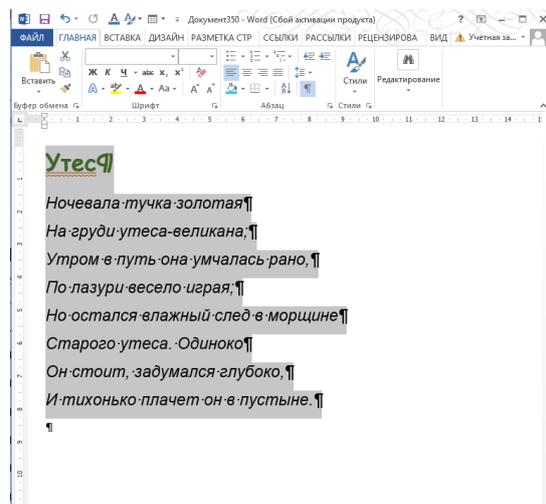


Рис. 93. Исходный вид стихотворения

Сохраните файл.

3. Откройте (продолжаем работать, если файл открыт) файл **poem.doc**.

Уменьшите соответственно левую и правую границы этого абзаца с помощью движков (рис. 94) на линейке (или используя отступы слева и справа в пункте меню **Абзац**). Выделите название и текст стихотворения и разместите по центру строки (по горизонтали) в рамках выделенного диапазона (рис. 95).

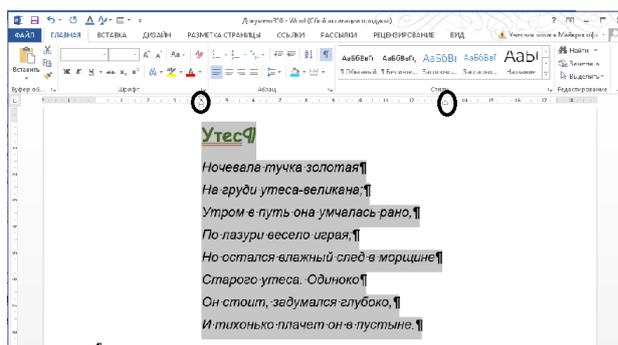


Рис. 95. Размещение стихотворения в рамках диапазона

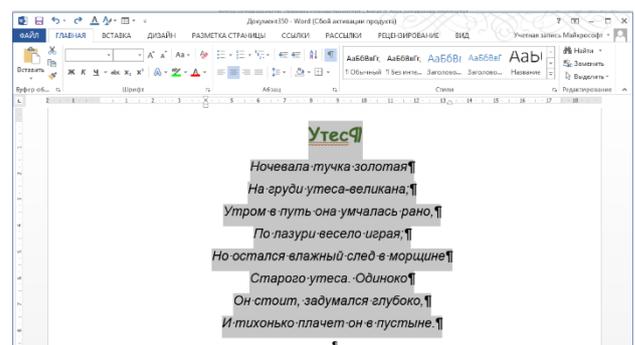


Рис. 96. Размещение стихотворения по центру в рамках диапазона

4. Задайте абзацу оформление какой-нибудь рамкой.

Для этого на основной панели вкладки Главная найдите значок Границы и раскройте выпадающее меню, чтобы настроить вид и цвет, а также толщину линий границы.

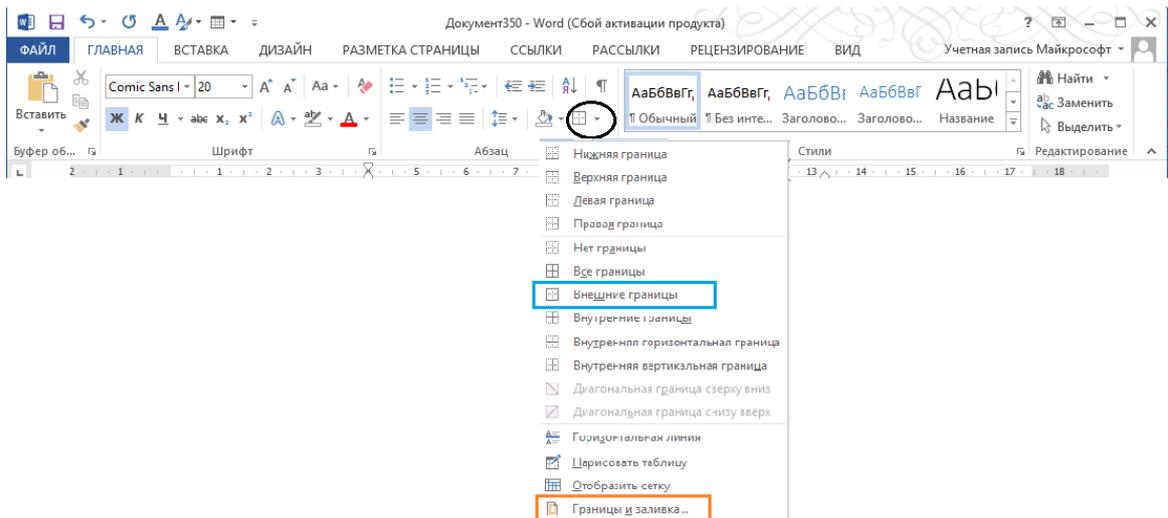


Рис. 97. Выпадающее меню выбора типа границ оформления текста

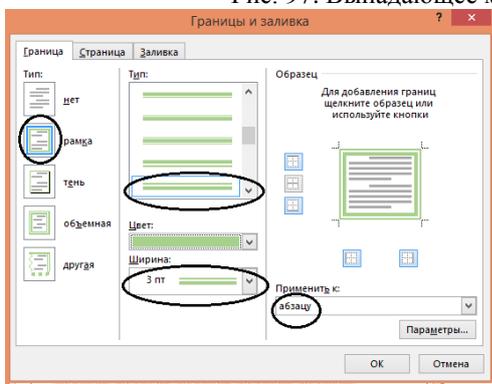


Рис. 98. Настройка границ оформления абзаца

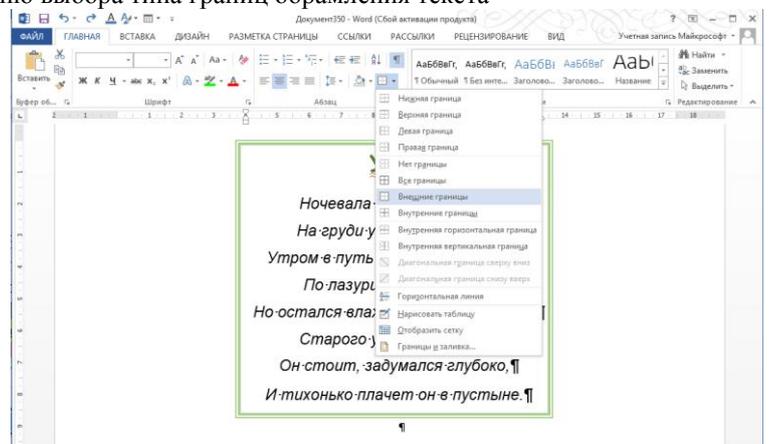


Рис. 99. Вид текста с оформлением

5. Сделайте заливку какого-либо слова стихотворения цветом. Для этого сначала выделите слово, а затем установив маркер на выделенном слове, найдите на вкладке **Главная** значок заливки шрифта  или заливки абзаца ,

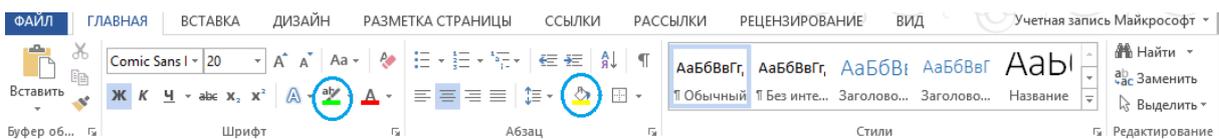


Рис. 100. Положение значков заливки шрифта и заливки абзаца на панели **Главная**

раскройте выпадающее меню значка, нажав на стрелочку справа от значка, и в выпадающем меню выберите нужный цвет.

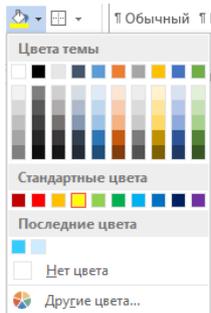


Рис. 101. Меню цветов для заливки абзаца

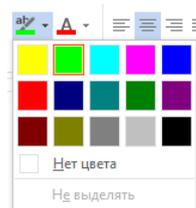


Рис. 102. Меню цветов для заливки шрифта

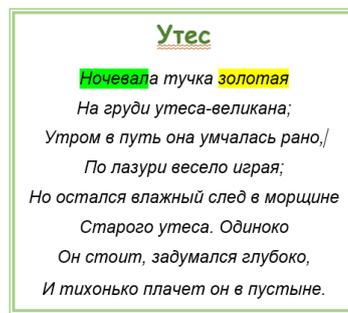


Рис. 103. Пример заливки слова



Рис. 104. Сдвиг всего текста

6. Сдвиньте текст стихотворения с рамкой вниз вставляя пустые строки

7. Задайте странице обрамление границей и заливку страницы узором.

Для обрамления страницы границей выберите вкладку **Дизайн** на основной панели и затем пункт **Границы страницы**, где затем выбираем настройки границы (которые аналогичны настройкам границ абзаца) и применяем их к странице.

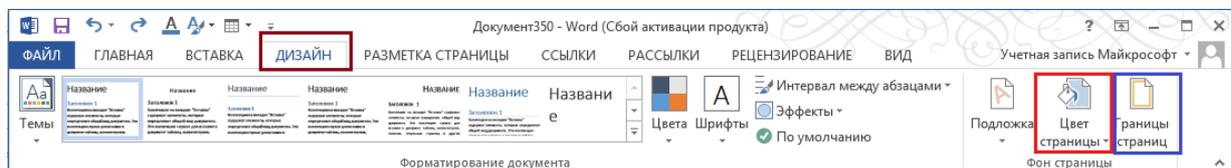


Рис. 105. Панель вкладки **Дизайн**

Затем выберите настройки пункта **Цвет страницы** и примените к странице (причем, здесь можно выбрать и сложный цвет, и сложный вид узора).

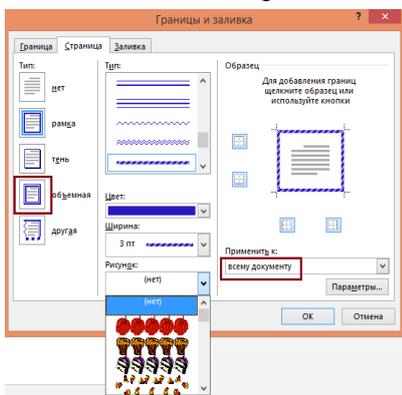


Рис. 106. Выбор рисунка обрамления страницы

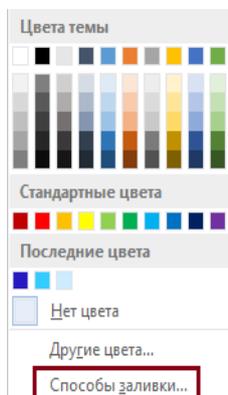


Рис. 107. Способы заливки страницы

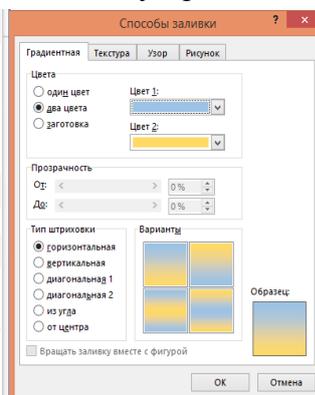


Рис. 108. Градиентный способ заливки

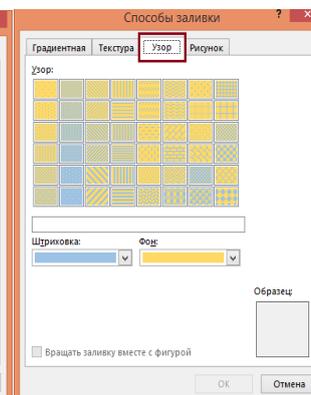


Рис. 109. Заливка узором из 2-х цветов

Можно получить, например, следующий результат применения рамки и заливки к странице (рисунок справа 110).

Сохраните файл под именем в своей папке *poem1.doc*.

8. Создайте файл, содержащий демонстрацию дополнительных возможностей шрифтов. Наберите математические, химические формулы с помощью пунктов **Надстрочный знак (x²)**, **Подстрочный знак (x₂)** в элементах шрифта:

Квадратное уравнение: $ax^2 + bx + c = 0$.

Азотная кислота: HNO_3 .

Сохраните файл под именем *math.doc* в своей папке.

9. Откройте файл *Zayavlenie.doc* и выполните абзацное форматирование и расстановку переносов в файле *«Zayavlenie»*.

Для абзацного форматирования используйте либо вкладку **Разметка страницы**->**Поля**, выбирая далее нужную разметку полей страницы: **Обычные**, **Узкие**, **Средние**, **Последнее пользовательское использование** и прочие или **Настраиваемые поля** (самостоятельно настраивая все отступы страницы: слева, справа, сверху, снизу). То же самое можно сделать, предварительно выделив весь текст на странице,

- выбрав пункт **Абзац** на основной панели либо
- щелкнув правой кнопки мыши, перейти на пункт **Абзац**,

далее использовать вкладку **Положение на странице**, пункт **запретить автоматический перенос слов** (поставить или снять галочку рис. 111).



Рис. 110. Вид страницы с заливкой узором и рамкой из рисунков

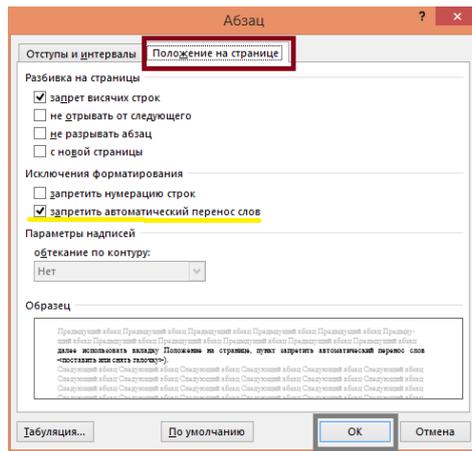


Рис. 111. Активированная вкладка **Положение на странице** пункта **Абзац**

10. Выполните расстановку переносов (**Абзац**—>**Положение на странице**->снять галочку «запретить авт. перенос слов»).

11. На вкладке **Вид** поставьте флажок **Линейка**.

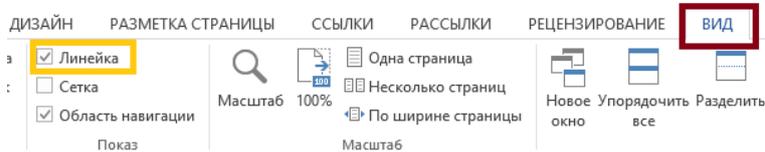


Рис. 112. Установка флажка в пункте **Линейка** вкладки **Вид**

12. Правую границу всего текста установите на 15 см (на горизонтальной линейке вверху текста установите правый маркер **отступ справа** в соответствующее положение).

13. Сформируйте из первых трех строк документа абзац шириной, равной самой продолжительной из них (на горизонтальной линейке вверху текста установите левый квадратный маркер **отступ слева** в соответствующее положение). Измените в полученном абзаце межстрочный интервал на **полуторный** (**Формат** —>**Абзац**).

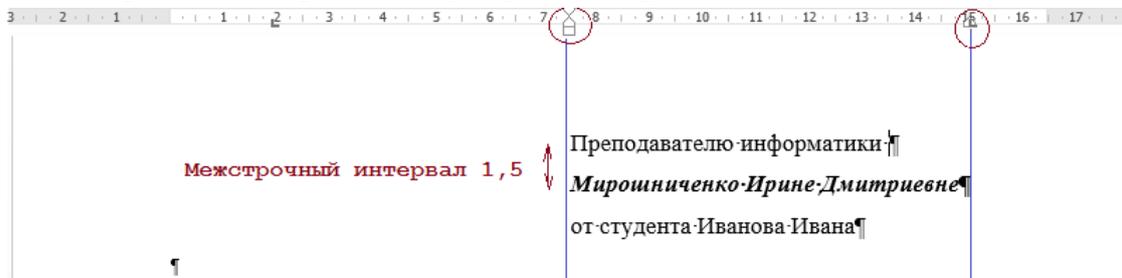


Рис. 113. Форматирование абзаца с помощью маркеров **Линейки**

14. В тексте заявления установите **красную строку 1 см** (на линейке установите левый треугольный маркер **отступ первой строки** в соответствующее положение), а левую границу текста заявления сдвиньте на отметку 2 см (используйте соответствующие маркеры или **Формат** —> **Абзац**, или маркер **отступ слева**). Задайте **двойной межстрочный** интервал.

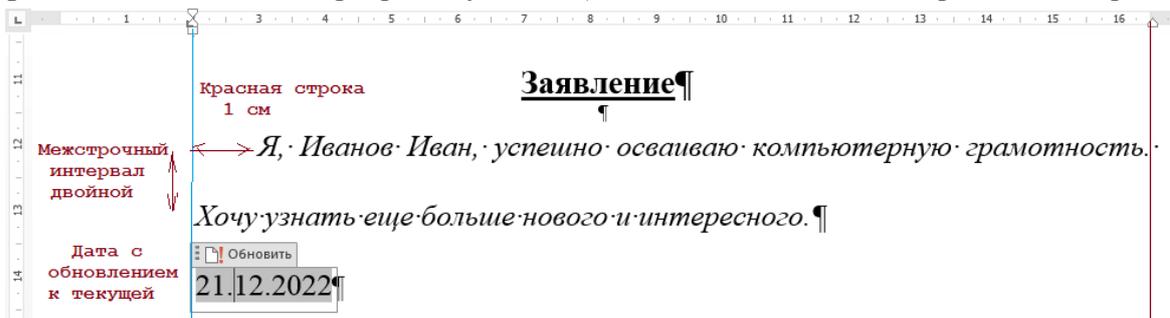


Рис. 114. Форматирование текста Заявления и вставка текущей даты (с обновлением)

Преподавателю информатики
Мирошниченко Ирине Дмитриевне
от студента Иванова Ивана

Заявление

Я, Иванов Иван, успешно осваиваю компьютерную грамотность.

Хочу узнать еще больше нового и интересного.

27.02.2023

15. Измените дату на текущую дату, используя комбинацию клавиш **Alt + Shift + D**. Сдвиньте дату вправо на отметку 2 см.

16. Сохраните файл под тем же именем **Zayavlenie1.doc** в своей папке.

Практическое занятие № 7. Буквица, колонки **Дополнительные возможности форматирования и набора текста**

1. Запустите текстовый процессор **MS Word**.
2. Откройте файл **poem.doc** или любой другой файл с большим объёмом текста.
3. Выведите на страницу **Линейку** разметки страницы (**Вид->Линейка-><поставить галочку>**)
4. Выделите текст и установите левый движок на горизонтальной линейке на 4 см, а правый движок — на 13 см (либо диапазон можно сделать шире в случае более длинных строк в тексте)
5. Настройте текст следующим образом: шрифт **Time New Roman**, размер **12**, *курсив*, размещение — по центру. Выполните обрамление первого абзаца текста справа и слева двойными линиями (**Главная->в блоке Абзац** выберите значок **Границы** ->**Граница и заливка**). Настройте тип линии (двойная линия), ширину линии и цвет. Затем примените к абзацу сначала левую границу, а потом — правую (рис. 115).

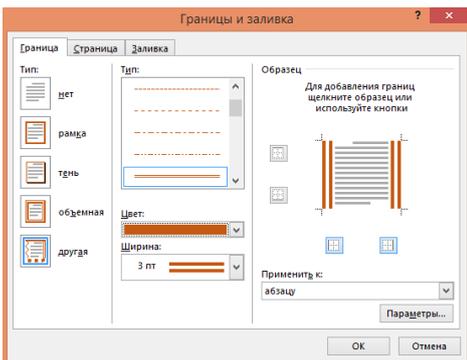


Рис. 115. Выбор ширины и вида границы (тип - двойная)

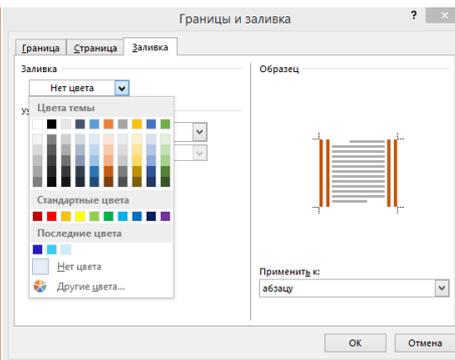


Рис. 116. Выбор цвета границы

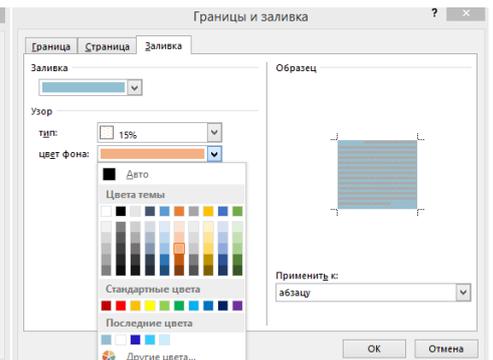


Рис. 117. Выбор цвета заливки выделенного текста (примененного к абзацу)

Залейте абзац цветным узором с прозрачностью 15%. Сначала устанавливаем курсор на абзаце и на Главной вкладке выбираете значок **Границы** ->**Граница и заливка** (рис. 116). В раскрывшемся меню выбираем основной цвет любым способом (цвет из таблицы, из набора RGB или из спектра), затем выбираем прозрачность (тип 15%) и, наконец, дополнительный цвет (рис. 117). Применяем к абзацу (рис. 118).

6. Настройте по этому абзацу остальные, используя кнопку **Формат по абзацу** на панели инструментов. Чтобы настроить форматирование для нескольких выделенных

настраиваемых абзацев, сделайте двойной щелчок на кнопке , затем щелкните в любом месте нескольких настраиваемых абзацев (рис. 119). Отключить кнопку щелчком кнопкой мыши по значку **Формат по абзацу**.

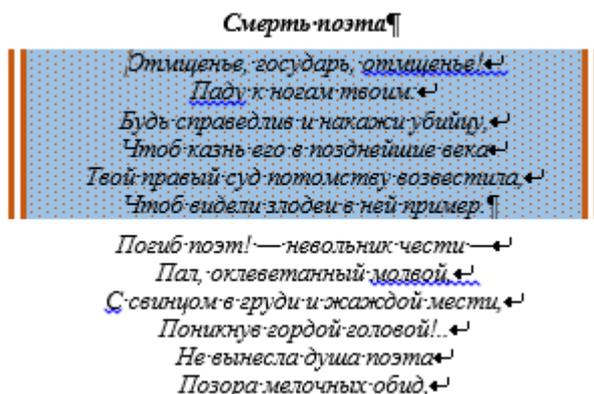


Рис. 118. Применение заливки с рисунком к абзацу

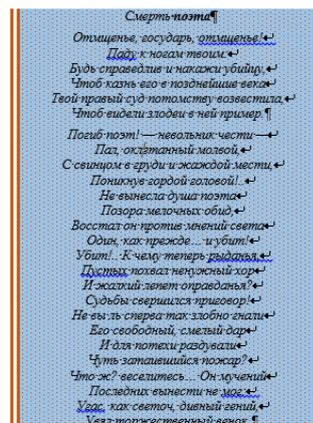


Рис. 119. Применение заливки ко всему тексту

Чтобы настроить по некоторому абзацу только один абзац, сделайте один щелчок по кнопке **Формат по абзацу**, затем щелкните в любом месте требуемого настраиваемого абзаца. После этого действие кнопки **Формат по абзацу** автоматически прекращается.

7. Сохраните документ в своей папке под именем *poem2.doc*.
 8. Откройте последовательно файлы *zagadka1.doc*, *zagadka2.doc*, *otgadka1.doc* и *otgadka2.doc*.
 9. Создайте новый документ. Сохраните документ в своей папке под именем *zagadki.doc*. Скопируйте в него тексты из открытых файлов следующим образом: первая загадка, отгадка, вторая загадка, отгадка. Перед каждой загадкой напишите Загадка, перед отгадкой — Отгадка. Используйте команды **Правка** → **Копировать** (или **Вырезать**) и **Правка** → **Вставить** или соответствующие кнопки на панели инструментов.
- Переход из одного окна другое осуществляется с помощью меню **Окно**. Откройте на мониторе 2 окна (**Вид** → **Новое окно**) и настройте удобную работу с двумя окнами на экране монитора. Отправьте преподавателю вид Вашего экрана с двумя окнами.
10. Научитесь выполнять **автозамену** части слов или фраз (фрагмента текста) с помощью вкладки **Главная** основного меню и пункта **Заменить** на главной панели: для этого задайте для автозамены следующие фрагменты слов: *заг* — от Загадка, *отг* — от Отгадка. В открывшейся вкладке **Найти и Заменить** выполните **автозамену**: заполните следующие поля вкладки **Найти и Заменить** соответствующими фрагментами.

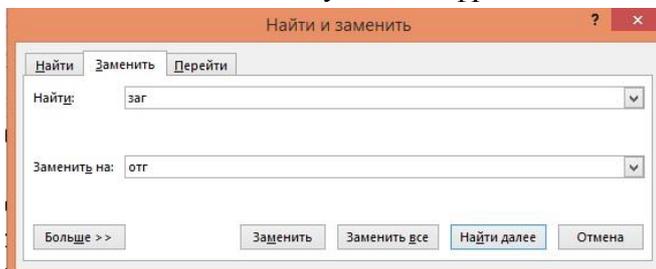


Рис. 120. Заполнение полей для автозамены

11. Удалите созданные автозамены либо возвратом действий, либо применив частично обратную автозамену.

12. Примените известные Вам приемы форматирования к рабочему документу *zagadki.doc*. Например, выберите шрифт для заголовков большего размера, чем текст, и другой гарнитуры, возможно применение подчеркивания. Тексту каждой загадки задать

обрамление, произвольно расположить на странице и применить буквицу (Формат → Буквица).

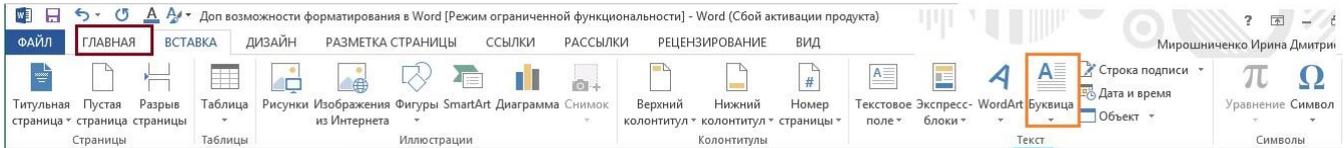


Рис. 121. Расположение пункта меню Буквица на основной панели

13. Выполните заливку абзацев разными цветами. Выполните автоматическую расстановку переносов.

Проявите свой оформительский талант. Используйте возможности инструмента Шрифт и Абзац.

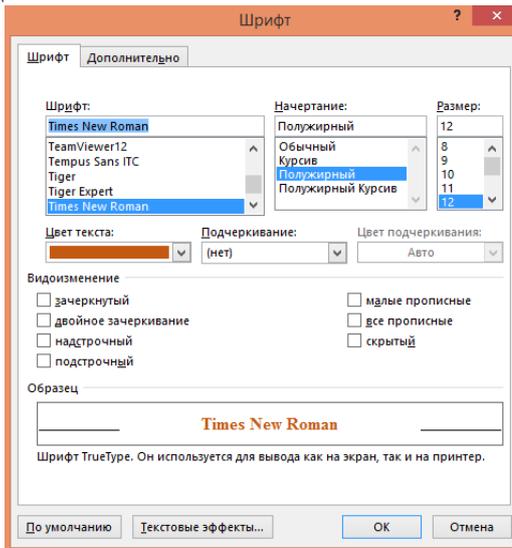


Рис. 122. Возможности инструмента Шрифт: выбор кегля, написания, начертания, цвета и других представлений шрифта.

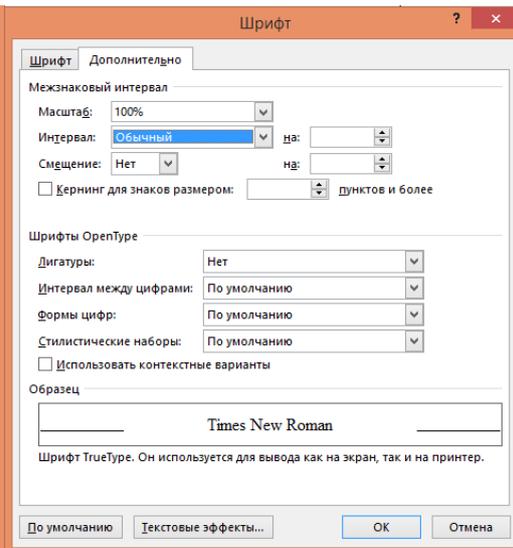


Рис. 123. Возможности инструмента Шрифт: уплотнение и растяжение выбранного фрагмента текста и других форматов символов

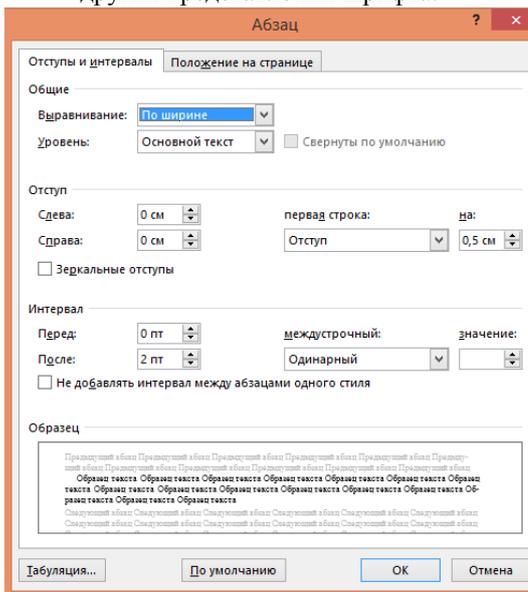


Рис. 124. Возможности инструмента Абзац: выравнивание абзаца по горизонтали и вертикали

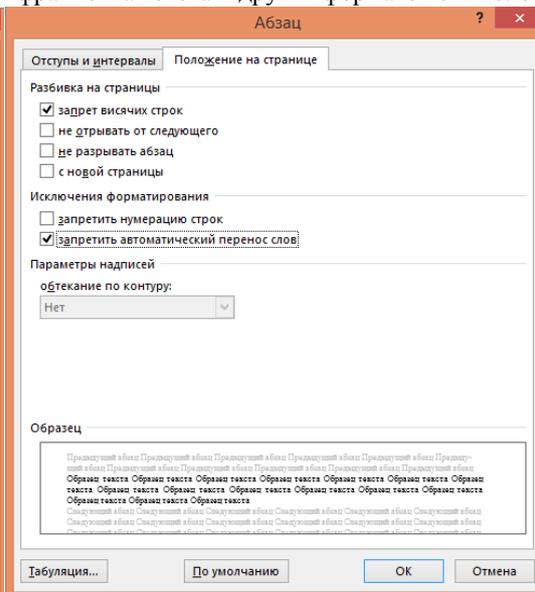


Рис. 125. Возможности инструмента Абзац: форматирование абзацев относительно друг друга

14. Сохраните документ в своей папке под именем *zagadki.doc*.

15. Откройте файл *poem2.doc*.

16. Под текстом наберите автора данного произведения. Задайте размер шрифта **12**, выравнивание по правому краю.

17. Раздвиньте левый и правый движки настолько широко, чтобы в две колонки текст помещался без изменений

18. Расположите текст в несколько колонок (**Разметка страницы** —> **Колонки**), выбрав нужное число колонок (рис. 126).



Рис. 126. Вид панели вкладки **РАЗМЕТКА СТРАНИЦЫ** и выпадающий список для выбора количества колонок.

Варьируя размеры колонок, добейтесь красивого набора. Возможна разбивка на колонки разной ширины и добавление разделителя, поэкспериментируйте с разными вариантами (рис. 127).

17. Вставьте рисунок в текст с колонками.



Рис. 127. Вид текста в две колонки

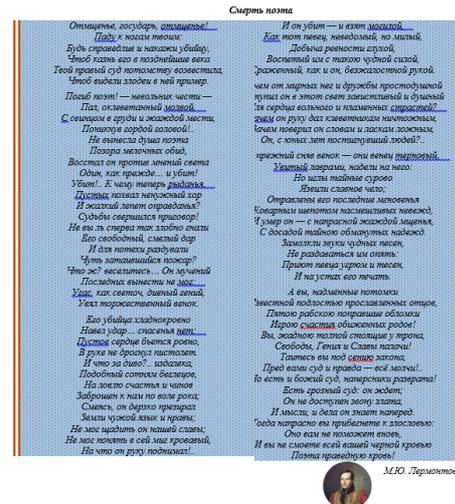


Рис. 128. Вид текста в две колонки с рисунком

18. Сохраните документ в своей папке под именем **kolonki.doc**.

Практическое занятие № 8. Списки Word. Оформление списков

Это практическое занятие посвящено форматированию текста в виде списков. Различают три вида списков:

- маркированный,
- нумерованный и
- многоуровневый.

Рассмотрим создание этих списков на примере следующих данных.

1. Наберите следующий перечень словосочетаний:

Компоненты компьютера
 Системный блок
 Монитор
 Клавиатура
 Мышь
 Программное обеспечение
 Системные программы
 Прикладные программы
 Компьютерные сети
 Локальные сети
 Глобальные сети

2. Оформление маркированным списком.

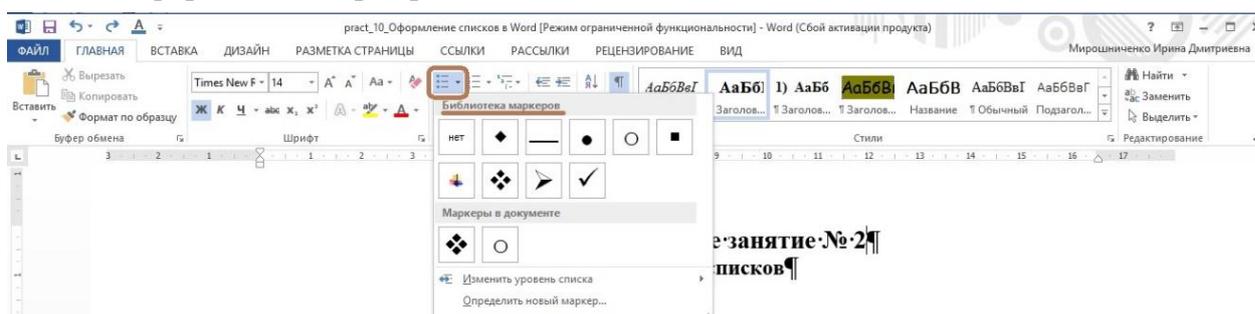


Рис. 129. Расположение маркированного маркера на основной панели и выпадающий список видов значков

Выделите элементы главного (или первого) подсписка. Их можно выделить, например, следующим образом: выделить первый элемент (Компоненты компьютера),

- трижды щелкнув по нужной строке или,
- зажав левую кнопку мыши, провести слева направо, выделяя нужную строку) будущего списка,

затем нажмите кнопку **CTRL** и выделите второй элемент будущего списка (Программное обеспечение) и так далее. В нашем списке выделим (при нажатой кнопке **CTRL**) третий элемент будущего списка (Компьютерные сети), рисунок 130.

По умолчанию это будут элементы первого уровня (об уровнях чуть позже).

Выполните команду **Выбрать маркер из Библиотеки маркеров** (нажмите на значок ) и выберите вариант маркера. На рисунке выбран значок маркера . Нажмите **ENTER**.

Получите результат, представленный на рисунке 131.

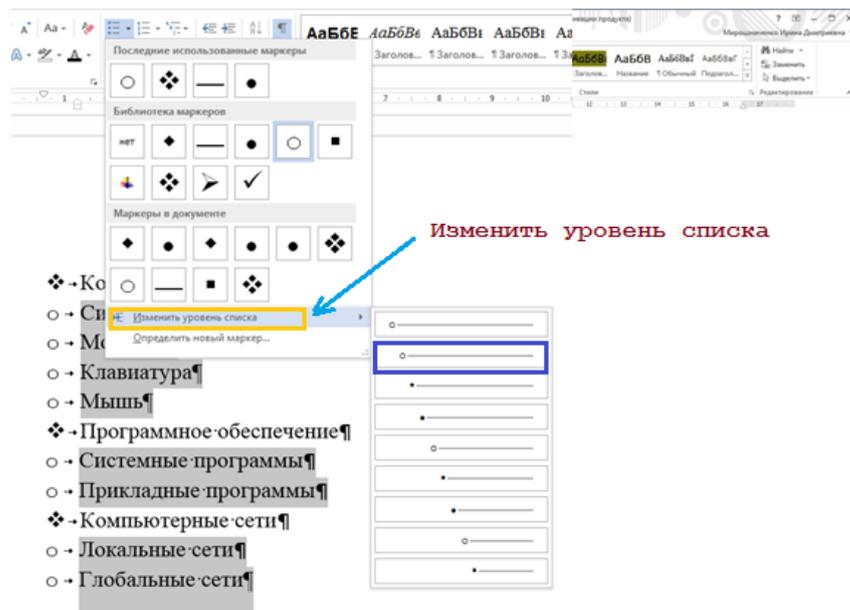
Компоненты компьютера
 Системный блок
 Монитор
 Клавиатура
 Мышь
 Программное обеспечение
 Системные программы
 Прикладные программы
 Компьютерные сети
 Локальные сети
 Глобальные сети

Рис. 130. Выделение элементов главного подсписка

• Компоненты компьютера
 Системный блок
 Монитор
 Клавиатура
 Мышь
 • Программное обеспечение
 Системные программы
 Прикладные программы
 • Компьютерные сети
 Локальные сети
 Глобальные сети

Рис. 131. Результат применения маркера  к элементам главного подсписка

Теперь выделите элементы вторых подсписков (рис. 133). Выберите маркер для идентификации элементов второго подсписка – значок . Нажмите **ENTER**, получите результат, представленный на рисунке 134.



Изменить уровень списка

Рис. 132. Выбор требуемого уровня для элементов подсписков

- ❖ → Компоненты компьютера
 - Системный блок
 - Монитор
 - Клавиатура
 - Мышь
- ❖ → Программное обеспечение
 - Системные программы
 - Прикладные программы
- ❖ → Компьютерные сети
 - Локальные сети
 - Глобальные сети

Рис. 133. Подготовка элементов вторых подсписков

- ❖ → Компоненты компьютера
 - → Системный блок
 - → Монитор
 - → Клавиатура
 - → Мышь
- ❖ → Программное обеспечение
 - → Системные программы
 - → Прикладные программы
- ❖ → Компьютерные сети
 - → Локальные сети
 - → Глобальные сети

Рис. 134. Результат применения маркера ○ к элементам второго подсписка

Осталось привести наш список к удобном для восприятия виду, то есть сместить вправо элементы подсписков второго уровня. Для этого снова обратитесь к значку маркированного списка, выберите строку «Изменить уровень списка», расположенную ниже Библиотеки маркеров и затем выберите нужный вам уровень списка (рис. 132).

В результате получите окончательный результат, представленный на рисунках 135, 136.

- ❖ → Компоненты компьютера
 - → Системный блок
 - → Монитор
 - → Клавиатура
 - → Мышь
- ❖ → Программное обеспечение
 - → Системные программы
 - → Прикладные программы
- ❖ → Компьютерные сети
 - → Локальные сети
 - → Глобальные сети

Рис. 135. Вид списка после выбора второго уровня для второго подсписка

- ❖ → Компоненты компьютера
 - → Системный блок
 - → Монитор
 - → Клавиатура
 - → Мышь
- ❖ → Программное обеспечение
 - → Системные программы
 - → Прикладные программы
- ❖ → Компьютерные сети
 - → Локальные сети
 - → Глобальные сети

Рис. 136. Окончательный результат получения маркированного списка

3. Оформление нумерованным списком.

Выполняется аналогично оформлению маркированным списком, используя кнопку маркера нумерованных списков . Возьмем тот же набор фраз для нумерованного списка.

Компоненты компьютера
 Системный блок
 Монитор
 Клавиатура
 Мышь
 Программное обеспечение
 Системные программы
 Прикладные программы
 Компьютерные сети
 Локальные сети
 Глобальные сети

При нажатии на кнопку маркера нумерованных списков  раскрывается **Библиотека нумерованных списков**. Ниже Библиотеки нумерованных списков находится пункт **Изменить уровень списка**

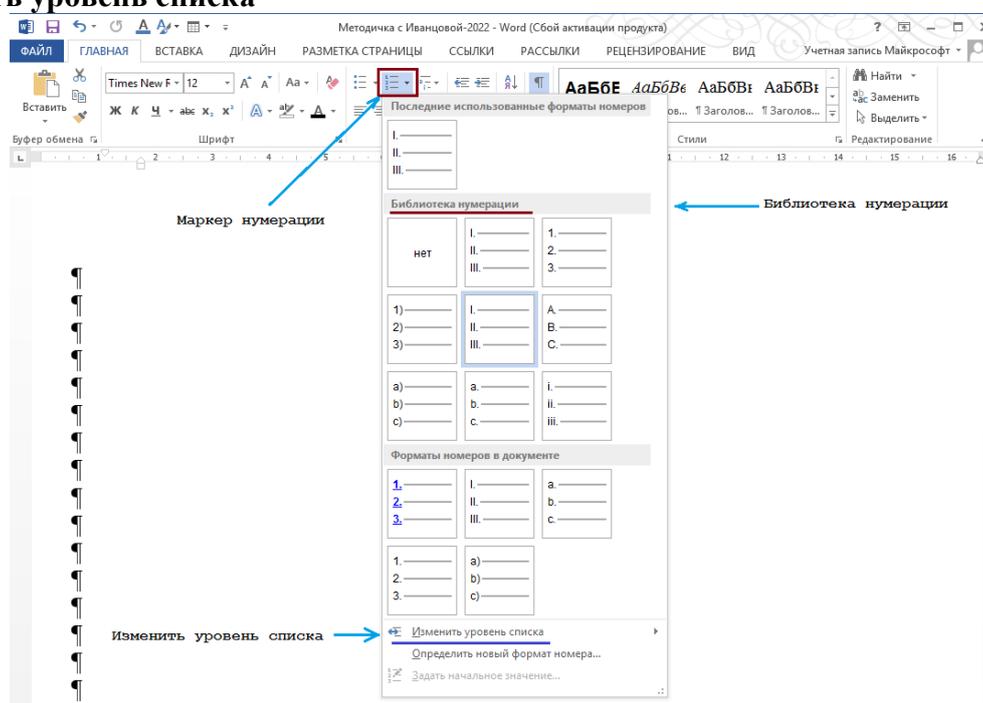


Рис. 137. Выбор маркера Нумерации и список маркеров нумерации Библиотеки нумерации

Выполните нумерацию списка **Римскими цифрами**.

Выделите все строки, которые собираетесь сделать заголовками первого, второго и третьего подсписков, удерживая клавишу **Ctrl** (рис. 138). Нажмите кнопку маркера Нумерации и выберите вид нумерации (I, II, III., выделено на рисунке 137).

Возможный результат выполнения операции представлен на рисунке 139.

Компоненты компьютера
 Системный блок
 Монитор
 Клавиатура
 Мышь
 Программное обеспечение
 Системные программы
 Прикладные программы
 Компьютерные сети
 Локальные сети
 Глобальные сети

Рис. 138. Выделение элементов для нумерованного списка

I. Компоненты компьютера
 Системный блок
 Монитор
 Клавиатура
 Мышь
 II. Программное обеспечение
 Системные программы
 Прикладные программы
 III. Компьютерные сети
 Локальные сети
 Глобальные сети

Рис. 139. Возможный вид списка после нажатия кнопки нумерованный список

Но иногда полезно после выполнения операции нумерации с выделенными элементами списка, не отжимая кнопки **CTRL** (или заново нажав на кнопку маркера нумерации ) , выбрать в выпадающем списке **Нумерации** пункт **Изменить уровень списка** и далее перейти в меню уровней и нажать *первый уровень*. Получим вид списка, представленный на рисунке 139.

Теперь для лучшего восприятия списка надо сдвинуть вправо элементы, представляющие конкретное содержание каждого из элементов главного списка. Для этого выделите эти элементы как на рисунке 140 и выполните сдвиг вправо

- либо используя кнопку сдвига выделенного текста вправо  на основной панели,
- либо двигая левый движок вправо на **Линейке** (рис. 141).

I.Компоненты компьютера¶
 Системный блок¶
 Монитор¶
 Клавиатура¶
 Мышь¶
 II.Программное обеспечение¶
 Системные программы¶
 Прикладные программы¶
 III.Компьютерные сети¶
 Локальные сети¶
 Глобальные сети¶

Рис. 140. Подготовка к форматированию (сдвигу) элементов не главного подсписка

Буфер обмена Шрифт Абзац
 I. → Компоненты компьютера¶
 Системный блок¶
 Монитор¶
 Клавиатура¶
 Мышь¶
 II. → Программное обеспечение¶
 Системные программы¶
 Прикладные программы¶
 III. → Компьютерные сети¶
 Локальные сети¶
 Глобальные сети¶

Рис. 141. Использование левого движка для сдвига текста

После сдвига получим результат (рис. 142) и, окончательно, получим вид списка как на рисунке 143.

I.Компоненты компьютера¶
 Системный блок¶
 Монитор¶
 Клавиатура¶
 Мышь¶
 II.Программное обеспечение¶
 Системные программы¶
 Прикладные программы¶
 III.Компьютерные сети¶
 Локальные сети¶
 Глобальные сети¶

Рис. 142 Вид списка после сдвига

I.Компоненты компьютера¶
 Системный блок¶
 Монитор¶
 Клавиатура¶
 Мышь¶
 II.Программное обеспечение¶
 Системные программы¶
 Прикладные программы¶
 III.Компьютерные сети¶
 Локальные сети¶
 Глобальные сети¶

Рис. 143. Окончательный вид списка

4. Оформление многоуровневым списком (создание многоуровневого списка).

Постройте многоуровневый список из следующего набора строк. Обозначьте элементы уровней списка:

- первого уровня - зеленым цветом,
- второго уровня – красным цветом,
- третьего уровня – синим цветом.

Исходный набор строк. Запишем его для компактности последовательно в столбики

Компоненты компьютера
 Системный блок
 Диагональ матрицы (экрана)
 Разрешение экрана
 Частота обновления экрана монитора
 Клавиатура
 Интерфейс
 Раскладка кириллицы
 Мышь
 Гироскопическая мышь
 Лазерная оптическая мышь

Программное обеспечение
 Системные программы
 Операционные системы
 Встроенные программы
 Утилиты
 Прикладные программы
 Текстовые редакторы.
 Графические редакторы.
 Пакеты программ

Компьютерные сети
 Локальные сети
 Компьютеры разных типов
 Сетевые адаптеры;
 Концентраторы;
 Программное обеспечение;
 Коммутаторы;
 Маршрутизаторы;
 Глобальные сети
 Распределенные сети
 Централизованные сети
 Децентрализованные сети

Выберите Стиль обычный и выделите весь список как текст. Затем нажмите значок маркера **Многоуровневый список** (рис. 144).

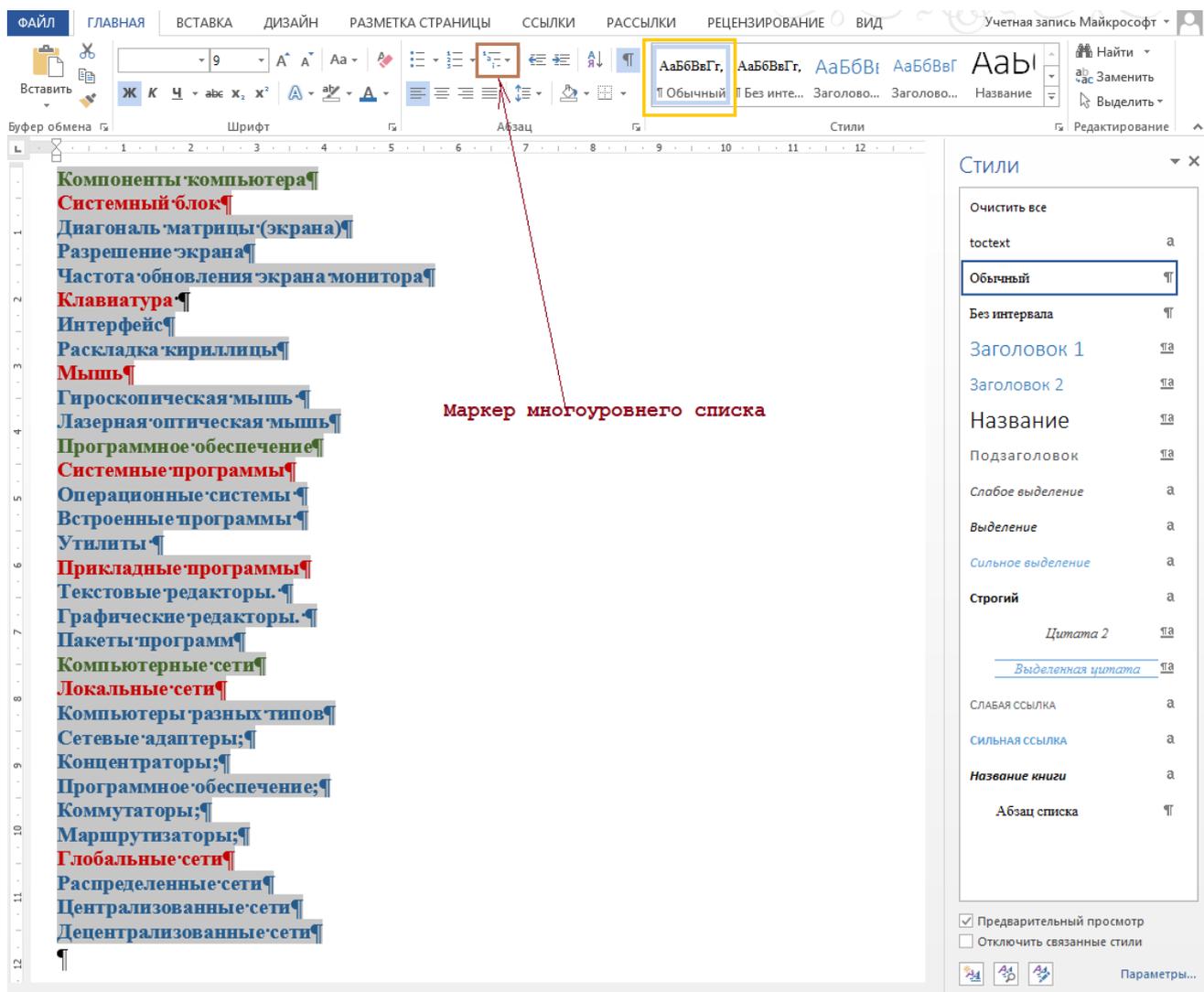


Рис. 144. Выделение исходного списка и применение маркера Многоуровневый список. Появится рабочее окно с названием **Все** (рисунок 145). В нём три раздела (подчёркнуты):

- Текущий список
- Библиотека списков
- Списки в текущих документах

На данный момент по состоянию этих разделов видно, что нет ни одного активного раздела.

Воспользуйтесь разделом **Библиотека списков**, который предоставляет различные способы форматирования многоуровневых списков (рис. 145).

Выберите подходящий вариант представления многоуровневого списка (рис. 146).

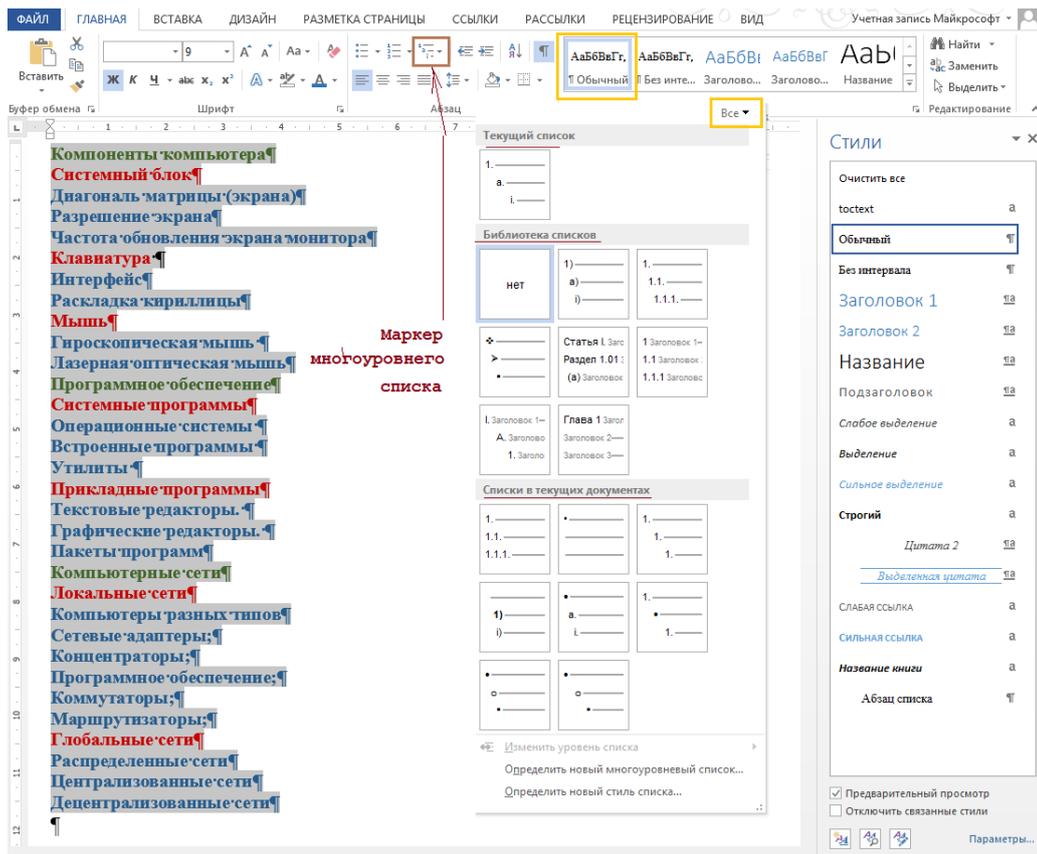


Рис. 145. Выделенный список и рабочее окно **Все** с тремя разделами и шаблонами многоуровневых списков в **Библиотеке списков**

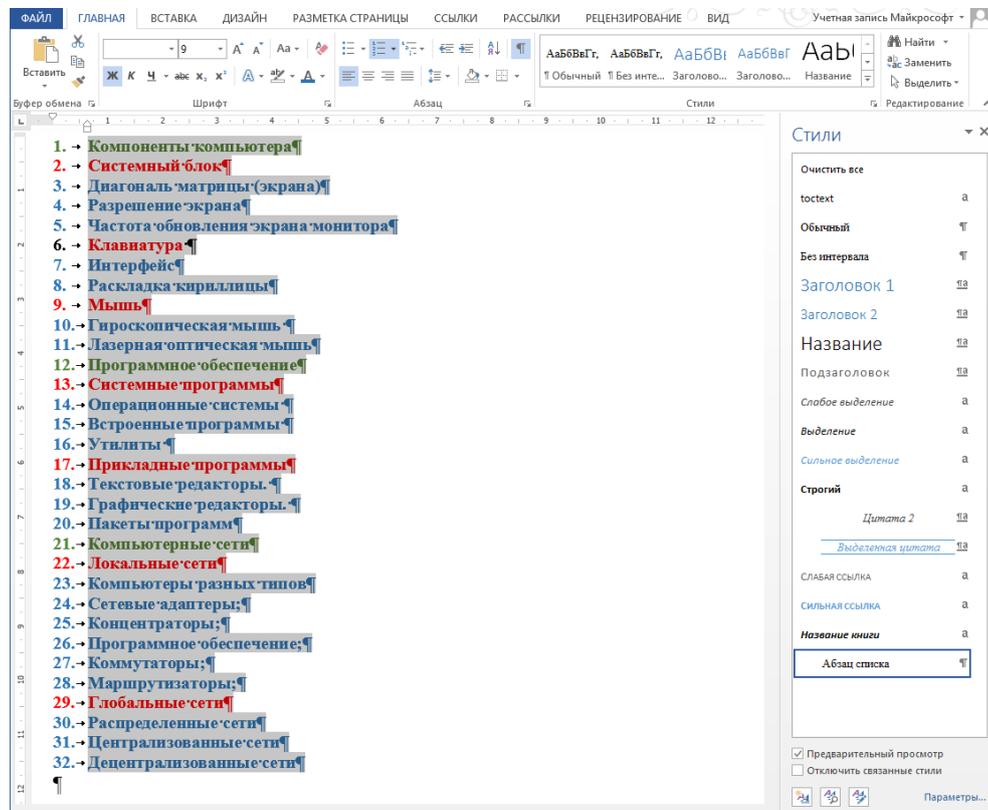


Рис. 146. Вид списка после применения выбранного шаблона многоуровневого списка (1.->1.1.->1.1.1.)

Выделите элементы исходного списка, которые должны стать элементами второго уровня (выделены шрифтом красного цвета), держа нажатой клавишу **CTRL** (рис. 147).

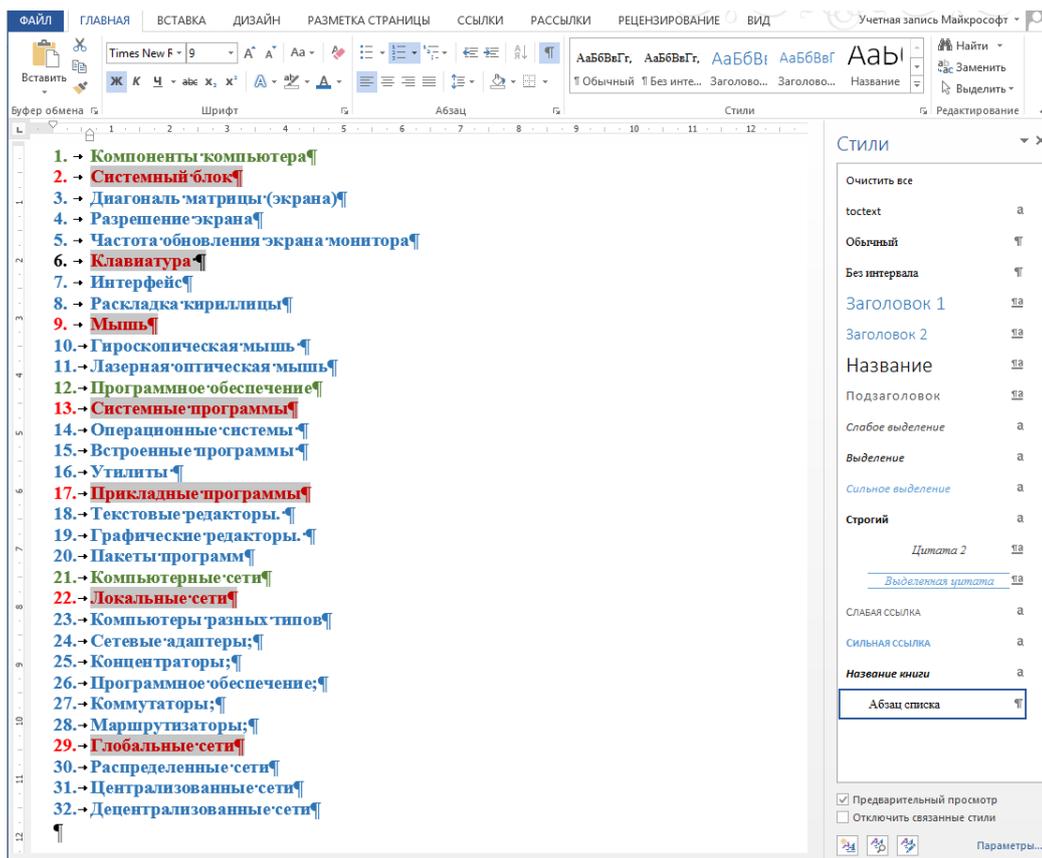


Рис. 147. Подготовка фрагментов к выделению подписка второго уровня

Выделенные элементы (шрифта красного цвета) сдвинем вправо с помощью значка сдвига текста вправо

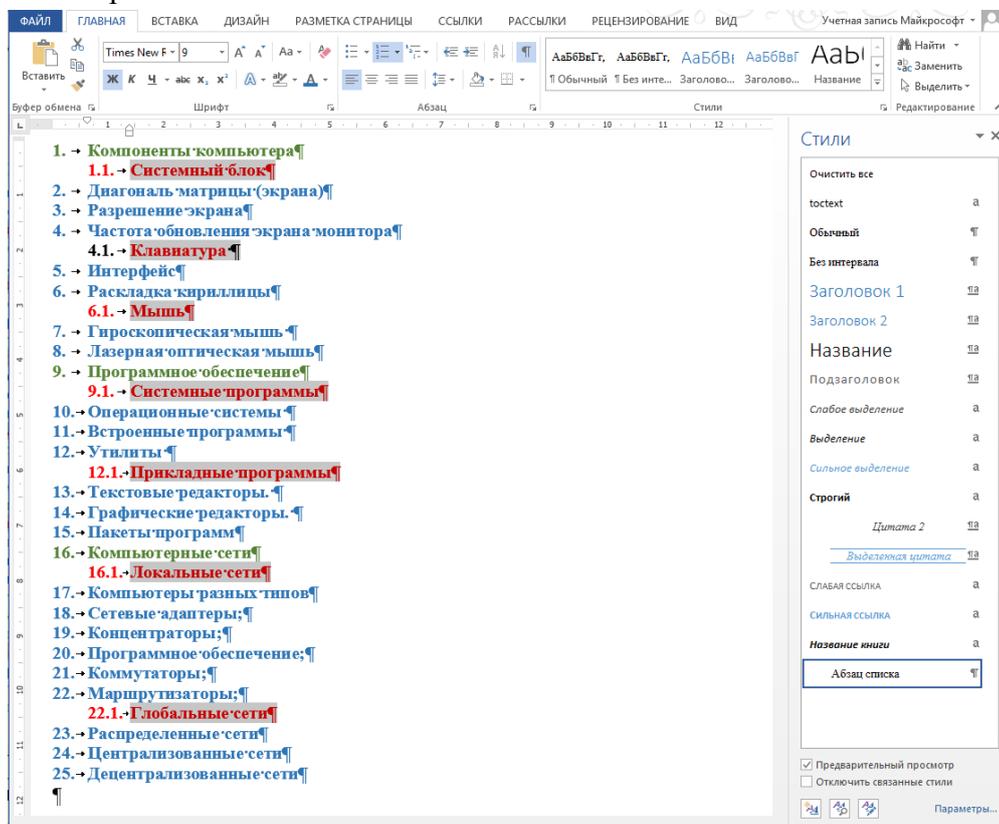


Рис. 148. Вид списка со смещенными вправо элементами подписка второго уровня

Сдвинув элементы подписка второго уровня (рис. 148), продолжите аналогичным образом работу с элементами подписка третьего уровня. Выделим элементы третьего уровня (со шрифтом синего цвета рис. 149).

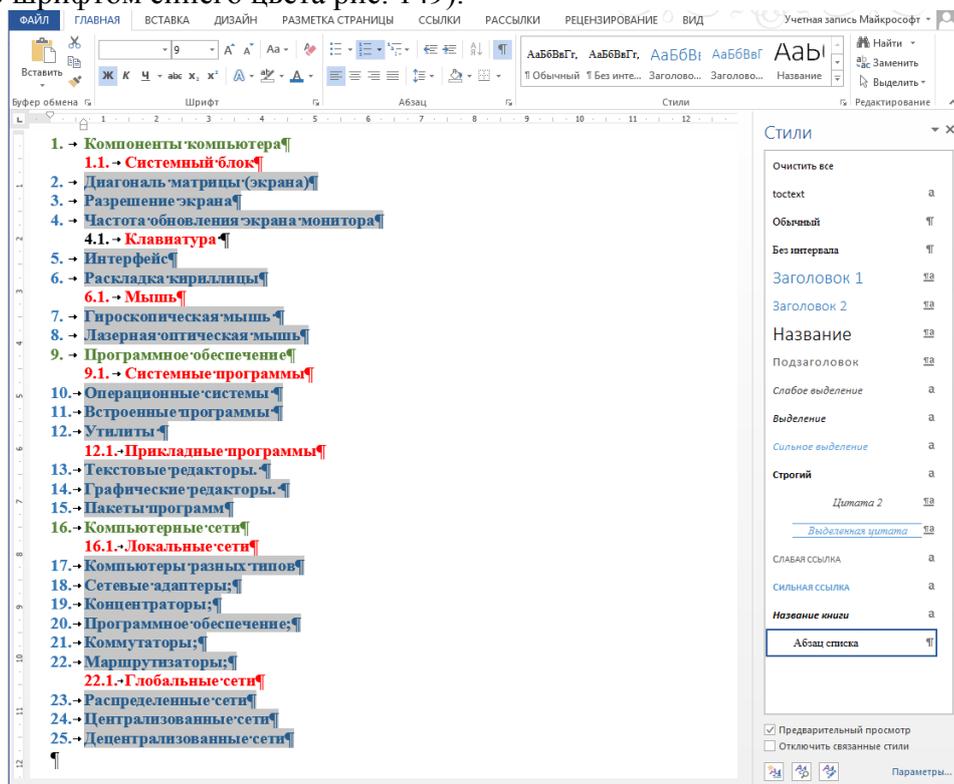


Рис. 149. Выделение элементов подписка третьего уровня (шрифт синего цвета)

Воспользуйтесь значком сдвига текста вправо, нажав его теперь уже два раза

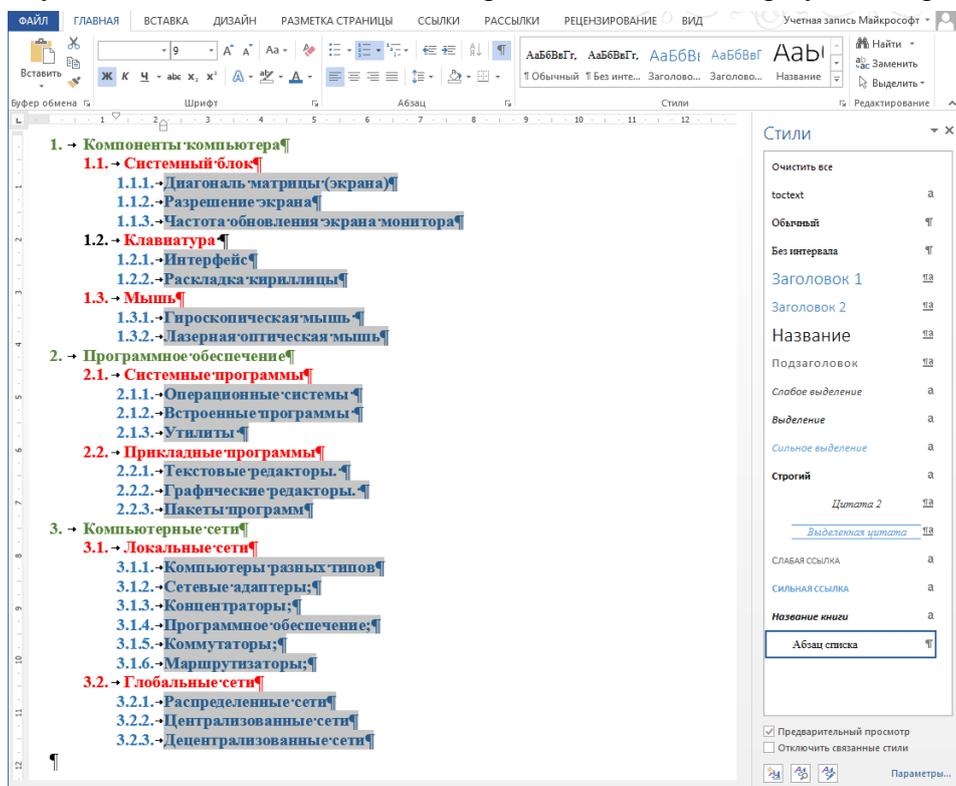


Рис. 150. Правильно сформированный многоуровневый список

В результате получите правильно сформированный многоуровневый список (рис. 150).

Практическое занятие № 9. Стили

1. Создайте файл, содержащий достаточный объем текста (скопируйте из какого-либо документа или Интернета).

2. Создание стиля абзаца.

Установите курсор в текст абзаца (в любом месте). На вкладке **Главная** найдите **окно Стили**, нажмите кнопку **Меню стилей** (рис. 151).

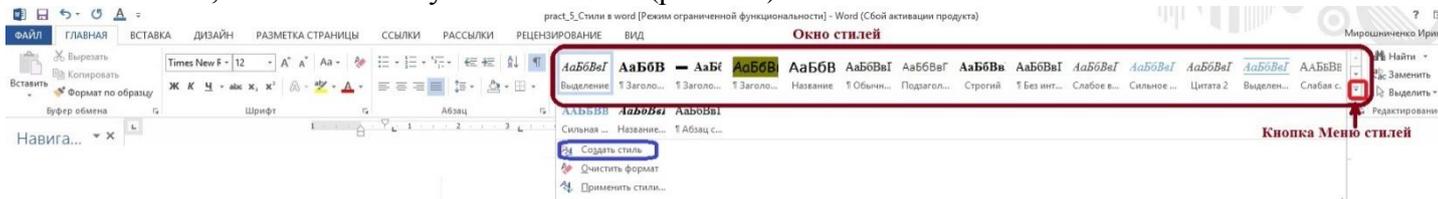


Рис. 151. Вкладка **Главная** с открытым **окном Стили**

В нижней части **окна Стили** расположено меню действий со стилями, в котором три пункта (рис. 151, 152):

- Создать стиль
- Очистить формат
- Применить стили...

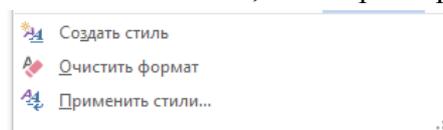


Рис. 152. Меню действий со стилями

Нажмите кнопку **Создать стиль**. Для создания нового стиля абзаца во вкладке **Создание стиля** (рис. 153) в поле **Имя:** Ваш_стиль задайте имя Вашего стиля.

В образце будет показан **Стиль абзаца** (на рисунке 153 это **Стиль1**).

Если же надо настроить именно Ваш стиль, нажмите кнопку **Изменить** (рис. 153).

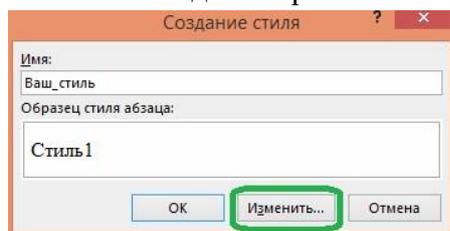


Рис. 153. Вкладка **Создание стиля**

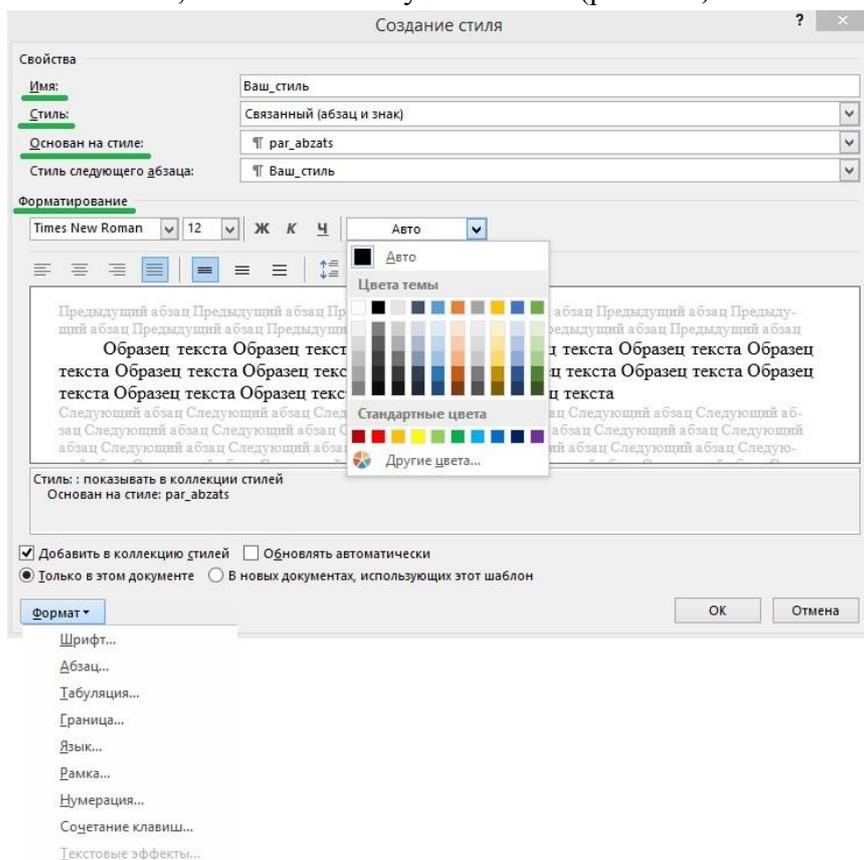


Рис. 154. Окно **создание стиля**

В открывшемся окне **Создание стиля** задайте в разделе **Свойства** следующие поля: **Стиль:** Абзаца, **Основан на стиле:** выбрать стиль, например, **Обычный** или **par abzats**,

(как на рисунке 154). В разделе **Форматирование**, если необходимо, измените **Шрифт**, **Размер** шрифта, **Начертание**, **Цвет** шрифта.

А именно, если нужен более детальный выбор параметров,

- перейдите к блоку **Форматирование** (рис. 154)
- или к кнопке **Формат** (раскрывая пункты меню) в левом нижем углу (рис. 154), выберите пункт меню **Шрифт** и задайте конкретный шрифт, например: **Times New Roman**, **размер 12, обычный**.

Кроме того, перейдя в блок **Форматирование** (или нажав кнопку **Формат**), можно выбрать пункт меню **Абзац** и задать: выравнивание **по ширине**, отступ первой строки **2 см**, межстрочный **интервал: 1,7 (множитель)**, интервал перед: **6 пт**, после **12 пт**.

После того, как выбрали все пункты оформления, нажмите кнопку **ОК** внизу вкладки **Создание стиля**. Созданный стиль появится в окне **Стилей**. Обратите внимание, что абзац отформатирован в соответствии с созданным стилем. Выделите оставшиеся абзацы исходного текста и примените к ним созданный **Стиль абзаца** из списка стилей.

Пример создания стиля и применения его к абзацам текста.

Что это было?¶
 Я шел зимою¶
 Вдоль болота¶
 В галошах,¶
 В шляпе¶
 И в очках.¶
 Вдруг по реке пронесся кто-то¶
 На металлических¶
 крючках.¶
 ¶
 Я побежал скорее к речке,¶
 А он бегом пустился в лес,¶
 К ногам приделал две дощечки,¶
 Присел,¶
 Подпрыгнул¶
 И исчез.¶
 ¶
 И долго я стоял у речки,¶
 И долго думал, сняв очки:¶
 «Какие странные¶
 Дощечки¶
 И непонятные¶
 Крючки!»¶
 Даниил Хармс¶

Рис. 155. Исходный текст

Что это было?¶
 Я шел зимою¶
 Вдоль болота¶
 В галошах,¶
 В шляпе¶
 И в очках.¶
 Вдруг по реке пронесся кто-то¶
 На металлических¶
 крючках.¶
 ¶
 Я побежал скорее к речке,¶
 А он бегом пустился в лес,¶
 К ногам приделал две дощечки,¶
 Присел,¶
 Подпрыгнул¶
 И исчез.¶
 ¶
 И долго я стоял у речки,¶
 И долго думал, сняв очки:¶
 «Какие странные¶
 Дощечки¶
 И непонятные¶
 Крючки!»¶
 Даниил Хармс¶

Рис. 156. Выделили абзац

Что это было?¶
 Я шел зимою¶
 Вдоль болота¶
 В галошах,¶
 В шляпе¶
 И в очках.¶
 Вдруг по реке пронесся кто-то¶
 На металлических¶
 крючках.¶
 ¶
 Я побежал скорее к речке,¶
 А он бегом пустился в лес,¶
 К ногам приделал две дощечки,¶
 Присел,¶
 Подпрыгнул¶
 И исчез.¶
 ¶
 И долго я стоял у речки,¶
 И долго думал, сняв очки:¶
 «Какие странные¶
 Дощечки¶
 И непонятные¶
 Крючки!»¶
 Даниил Хармс¶

Рис. 157. Применение стиля к абзацу

Этапы создания индивидуального стиля Хармс.

а). Ввод имени стиля в окно **Создание стиля** (рис. 154, 155).

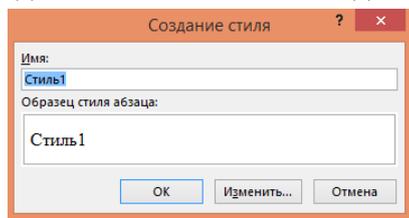


Рис. 158. Начальный вид окна **Создание стиля**

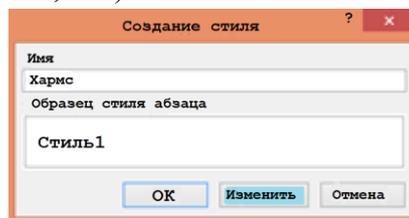


Рис. 159. Ввод названия стиля в окно **Создание стиля**

б). При нажатии кнопки **Изменить** осуществляется переход к выбору индивидуальных свойств стиля в блоке **Свойства** окна **Создание стиля** (рис. 160).

В поле **Имя** уже присутствует имя (рис. 160), которое было задано на предыдущем шаге (Хармс).

В поле **Стиль** требуется выбрать вид стиля (в зависимости от того, к чему он применяется). В примере из предложенного списка (рис. 160) выбрано значение **Абзац**.

Значение поля **Основан на стиле** также выбирается из предлагаемого списка (рис. 162). В примере выбрано значение **Основной текст** (рис. 161).

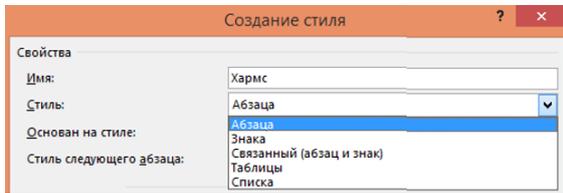


Рис. 160. Выбор стиля из стандартных шаблонов

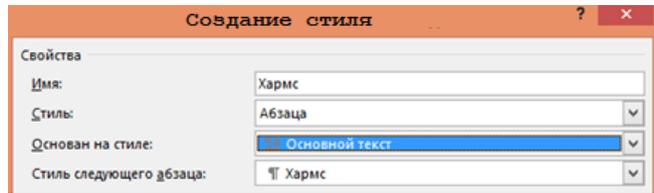


Рис. 161. Заполнение полей блока Свойств

Теперь можно перейти к блоку **Форматирование** (рис. 163) или к кнопке **Формат**, в которых можно изменить параметры шрифта: кегль, начертание, написание, подчеркивание и цвет,

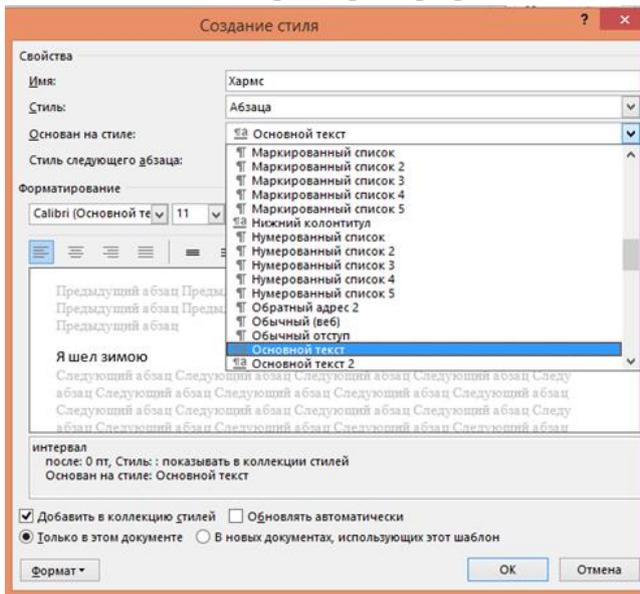


Рис. 162. Список шаблонных стилей для выбора

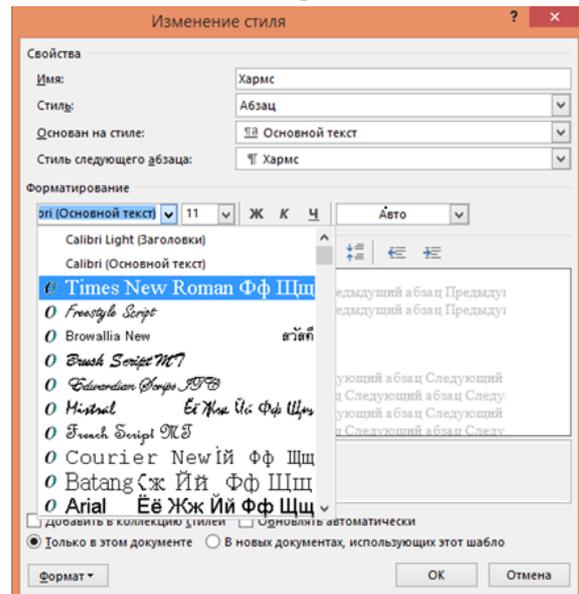


Рис. 163. Выбор шрифта из блока **Форматирование**

а также размещение текста по горизонтали, выбор интервала между строками абзаца (рис. 164), отступ текста слева и справа и отступы между абзацами.

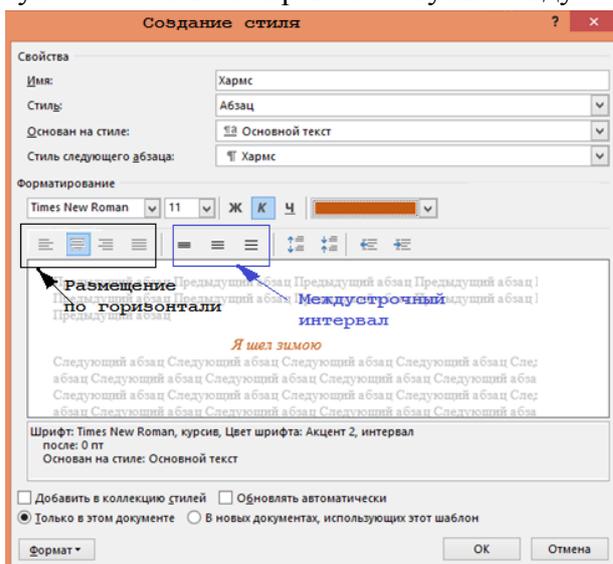


Рис. 164. Элементы размещения текста в блоке **Форматирование**

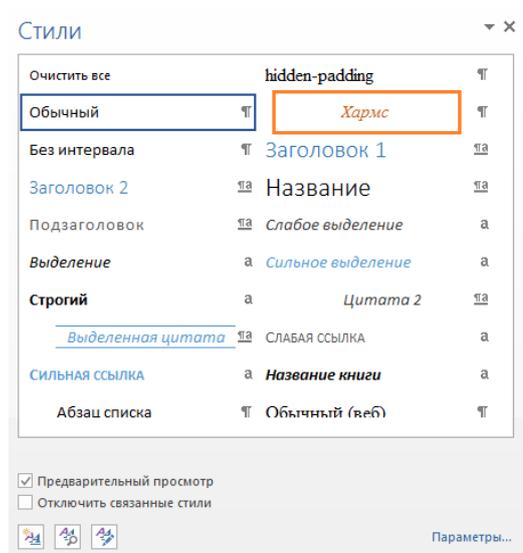


Рис. 165. В списке стилей получили стиль **Хармс**

После настройки всех параметров индивидуального стиля (рис.164), нажмите кнопку ОК и стиль Хармс создан! Его наличие можно увидеть в списке Стилей (Стили рис.165).

В примере выделили первый абзац текста (рис. 156), выбрали стиль Хармс и получили результат (рис. 166).

Теперь распространим этот стиль на весь текст стихотворения (без названия и автора). Для этого выделим оставшиеся два абзаца стихотворения (рис. 167), выберем из списка стилей стиль Хармс и нажмем ОК. Получим результат, показанный на рисунке 168.

Что это было?¶
Я шел зимою¶
Вдоль болота¶
В галошах,¶
В шляпе¶
И в очках.¶
Вдруг по реке пронесся кто-то¶
На металлических¶
крючках.¶
 ¶
 Я побежал скорее к речке,¶
 А он бегом пустился в лес,¶
 К ногам приделал две
 дощечки,¶
 Присел,¶
 Подпрыгнул¶
 И исчез.¶
 ¶
 И долго я стоял у речки,¶
 И долго думал, сняв очки.¶
 «Какие странные¶
 Дощечки¶
 И непонятные¶
 Крючки!»¶
 Даниил Хармс¶

Рис. 166. Результат применения
 стиля Хармс к абзацу

Что это было?¶
Я шел зимою¶
Вдоль болота¶
В галошах,¶
В шляпе¶
И в очках.¶
Вдруг по реке пронесся кто-то¶
На металлических¶
крючках.¶
 ¶
 Я побежал скорее к речке,¶
 А он бегом пустился в лес,¶
 К ногам приделал две
 дощечки,¶
 Присел,¶
 Подпрыгнул¶
 И исчез.¶
 ¶
 И долго я стоял у речки,¶
 И долго думал, сняв очки.¶
 «Какие странные¶
 Дощечки¶
 И непонятные¶
 Крючки!»¶
 Даниил Хармс¶

Рис. 167. Выделение оставшихся
 абзацев стихотворения

Что это было?¶
Я шел зимою¶
Вдоль болота¶
В галошах,¶
В шляпе¶
И в очках.¶
Вдруг по реке пронесся кто-то¶
На металлических¶
крючках.¶
 ¶
Я побежал скорее к речке,¶
А он бегом пустился в лес,¶
К ногам приделал две дощечки,¶
Присел,¶
Подпрыгнул¶
И исчез.¶
 ¶
И долго я стоял у речки,¶
И долго думал, сняв очки.¶
«Какие странные¶
Дощечки¶
И непонятные¶
Крючки!»¶
 Даниил Хармс¶

Рис. 168. Результат применения
 стиля Хармс к выделенному тексту

3. Аналогичным образом создайте стиль для **заголовка**. Выделите текст заголовка. Нажмите кнопку **Создать стиль**. Для создания нового стиля заголовка поступите аналогично созданию стиля для текста. Для Вашего задания задайте:

Имя: Стиль заголовка Ваш_заголовок, **Стиль:** Абзаца, **Основан на стиле:** Обычный.

В блоке **Форматирование** (или с помощью кнопки **Формат**) выберите пункты:

- **Шрифт:** шрифт: **Arial Black**, размер **14**, полужирный.
- **Абзац:** выравнивание по центру, отступ слева 2 см, справа 2 см.
- **Цвет текста:** синий

Нажмите кнопку **ОК** и удостоверьтесь, что заголовок отформатирован в соответствии с созданным Вами стилем заголовка **Ваш_заголовок**.

Выделите оставшиеся заголовки и примените к ним **Стиль заголовка Ваш_заголовок** из списка стилей.

В примере стихотворение Хармса для заголовка создадим **Стиль Моё_название**, а для оформления шрифта автора — **Стиль Автор**. Характеристики этих стилей приведены ниже.

Стиль Моё_название (рис. 169) применяется ко всем выделенным названиям:

Шрифт: наименование *Time New Roman*, кегль: 14, начертание: *полужирный курсив*, выравнивание: *по центру*, цвет шрифта: *синий*, межстрочный интервал: *обычный*.

Стиль Автор (рис. 166) применяется ко всем выделенным подписям:

Шрифт: наименование *Time New Roman*, кегль: 12, начертание: *полужирный*, выравнивание: *по правому краю*, цвет шрифта: *оранжевый*, межстрочный интервал: *обычный*.

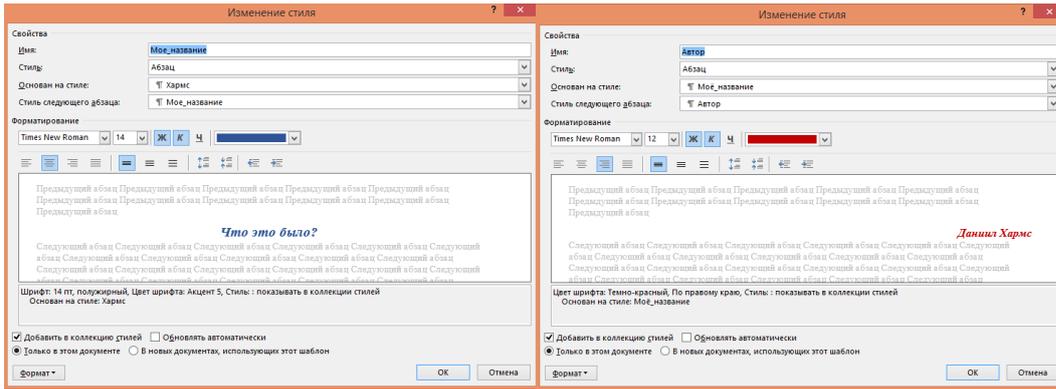


Рис. 169. Стиль для заголовка *Моё_название*

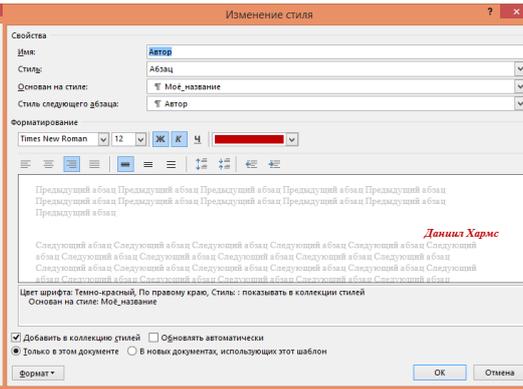


Рис. 170. Стиль *Автор*

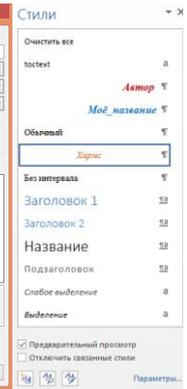


Рис. 171. Список стилей

На рисунке 171 видно, что в списке стилей появились стили *Автор*, *Моё_название*, *Хармс*. Для того, чтобы оформить выделенный текст (не в характеристиках стиля) рамкой или

границами выделите весь текст и нажмите кнопку **Границы и Заливка** . Откроется меню (рис. 172) для выбора границ (отдельно для каждой стороны, для внешних или внутренних границ) стандартным цветом. Выбрав пункт меню **Границы и заливка**, можно подобрать нужный вид, толщину и цвет границы (рис. 173) или вид, цвет, способ заливки (рис. 174), в том числе с узором с процентным отношением дополнительного цвета (рис. 175).

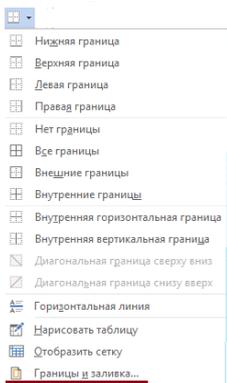


Рис. 172. Меню Границы

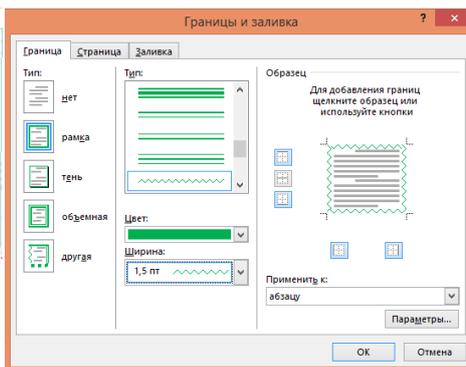


Рис. 173. Выбор характеристик границы абзаца

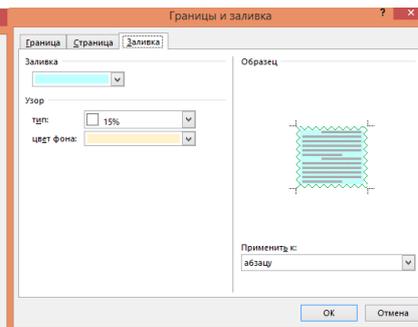


Рис. 174. Выбор характеристик заливки (узором)

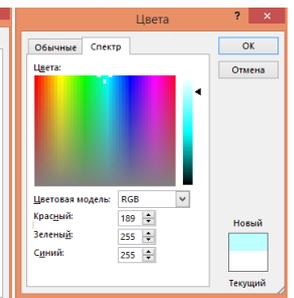


Рис. 175. Выбор цвета (спектром)

Ниже на рисунках приведен окончательный вид текста (стихотворения Даниила Хармса) без оформления границ и заливки (рис. 176) и с оформлением (рис. 177).

Что-это-было!
*Я шел зимою вдоль болота,
 В валюшках,
 В шляпе,
 И в бочках.*
*Вдруг по реке пронесся кто-то
 На металлических
 крючках.*
*Я побежал скорее к ручке,
 А он бегом пустился в лес,
 Присел,
 Подпрыгнул,
 И исчез.*
*И долго я стоял у ручки,
 И долго думал, сняв очки,
 «Какие странные»
 Дощечки,
 И непонятные
 Крючки!»*
Даниил Хармс

Рис. 176. Текст без оформления

Что-это-было!
*Я шел зимою вдоль болота,
 В валюшках,
 В шляпе,
 И в бочках.*
*Вдруг по реке пронесся кто-то
 На металлических
 крючках.*
*Я побежал скорее к ручке,
 А он бегом пустился в лес,
 Присел,
 Подпрыгнул,
 И исчез.*
*И долго я стоял у ручки,
 И долго думал, сняв очки,
 «Какие странные»
 Дощечки,
 И непонятные
 Крючки!»*
Даниил Хармс

Рис. 177. Текст с оформлением

Оформление абзаца можно ввести также при создании или изменении **Стиля**, указав в нем характеристики границ и заливки. Для этого нажмите кнопку **Формат** в левом нижнем

углу окна **Создать стиль** или **Изменить стиль** и примените пункт **Граница**. Далее, как в примере с **Форматом** на основной панели выберите требуемые

- линии границ, их цвет, вид форму
- заливку цветом или узором.

4. Создание знакового стиля.

Создание знакового стиля представляет возможность быстрого формирования списка определений и текста к ним как фрагментов текста.

Сначала создайте **Стиль определения**. **Стиль** создается аналогично выше описанному. Установите курсор в тексте на любом слове. Нажмите кнопку **Создать стиль**. Задайте следующие параметры в блоке **Свойства**.

Имя: Стиль *определения* **Ваше_определение** (рис. 178),

Стиль: Знака,

Основан на стиле: Основной шрифт абзаца.

Далее, нажав кнопку **Формат** (или используя блок **Форматирование**), выберите **Шрифт...** (рис. 179): написание **Courier New**, кегль **15**, начертание **полужирный курсив**, цвет шрифта **лиловый**, видоизменение: *малые прописные* (галочка в квадратике), интервал: *разреженный* на 1 пункт (рис. 180).

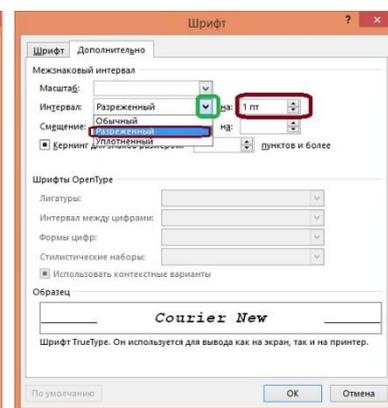
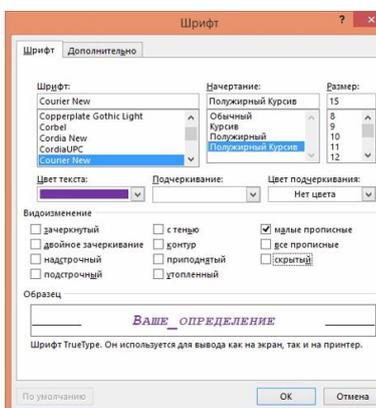
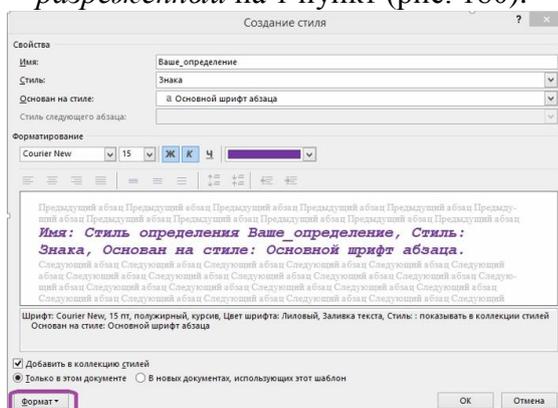


Рис. 178. Создание Стиля *Ваше_определение*

Рис. 179. Выбор характеристик шрифта

Рис. 180. Выбор интервала между символами

Теперь созданный стиль можно применить к другим словам-определениям. Для этого выделяем их и выбираем **Стиль определения: Ваше_определение**.

4. Создание оглавления.

Простейший способ быстро сделать оглавление/содержание на нужной странице. Для этого не обязательно нумеровать заголовки.

Необходимо выполнить 3 простых действия:

- ♦ выделить заголовки в нужных местах документа,
- ♦ применить к ним стили и
- ♦ скомпоновать их в оглавление.

4.1. Выделите заголовки в тексте. Возьмем случайный текст для примера (рис. 181):

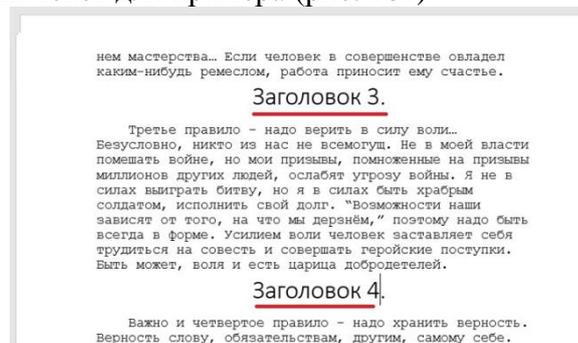
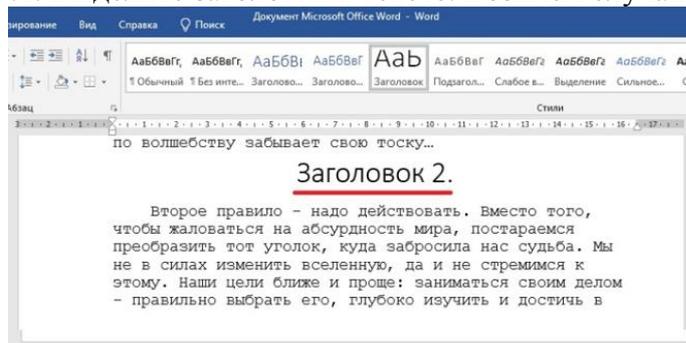


Рис. 181. Две страницы текста

4.2. Для каждого заголовка необходимо указать его **Стиль**.

Для этого выделите каждый заголовок по очереди и, обратившись к Главной панели (в верхней части окна), найдите вкладку **Стили** и выберите нужный стиль, например, стиль с названием "**Заголовок**" (рис.182).

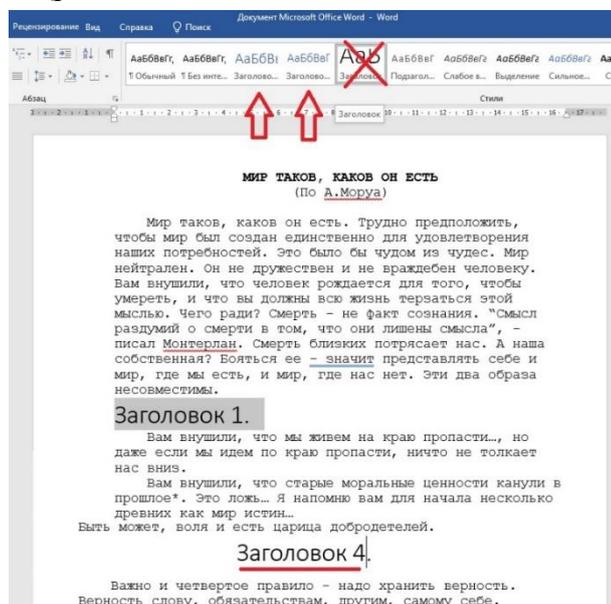


Рис. 182. Выбор и применение Стиля **Заголовок** к первому заголовку текста

Замечание 4.1. Нельзя использовать перечеркнутый стиль на рисунке выше (почему — об этом чуть позже в замечании 3).

4.3. Выбираем место, где будет располагаться наше **Оглавление** и отмечаем щелчком левой кнопкой мыши начало строки.

На Главной панели (рис. 183) находим вкладку **Ссылки**, выбираем пункт меню слева — **Оглавление** и открываем выпадающее меню этого пункта, в котором, в свою очередь, выбираем первый вариант **Автособираемое оглавление 1**.

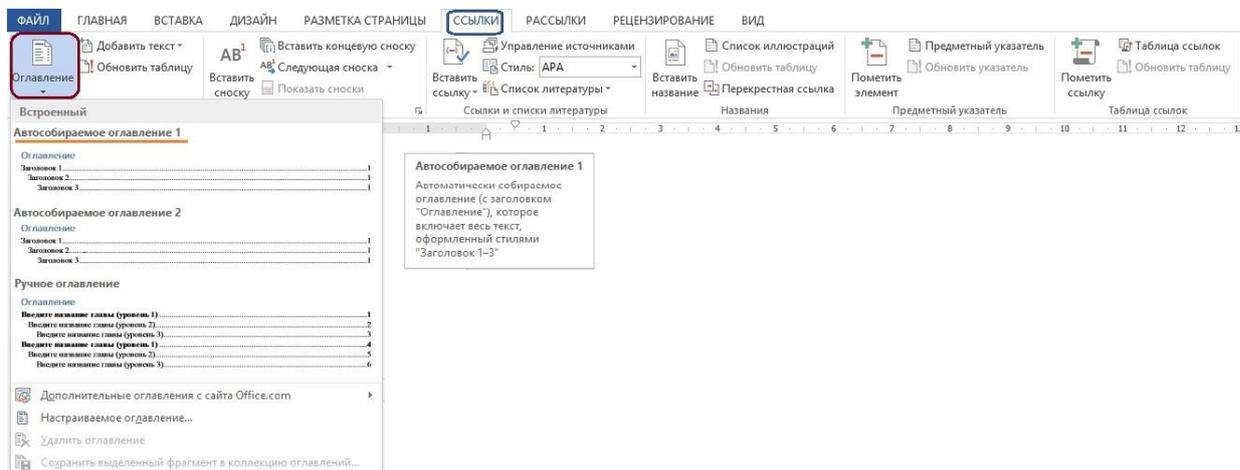


Рис. 183.Размещение вкладки Ссылки, пункта Оглавление на панели Ссылки и меню Оглавления

Замечание 4.2. Выбирать можно любой тип **Оглавления** на свой вкус. Первый пункт был выбран для упрощения.

В нашем примере получился следующий результат, который показан на рисунке 184. На нем стрелкой выделен **Стиль Заголовка**, который использовался (третий слева в окне **Стилей**, но первый из стилей **Заголовка**).

Замечание 4.3. Если выбрать 3-й стиль заголовка (указанный стрелкой), то Word выдаст следующее предупреждение: «*Элементы оглавления не найдены*» (рис. 185).

Чтобы этого избежать, выбирайте либо первый, либо второй стиль заголовка.

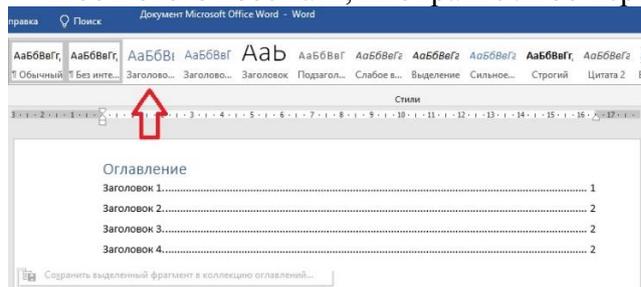


Рис. 185. Результат применения выделенного (первого по порядку) стиля Заголовка (крупно) к исходному тексту

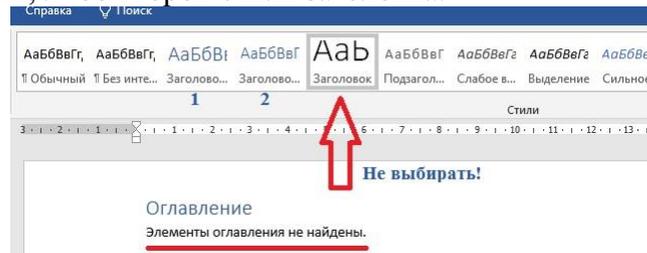


Рис. 186. Результат применения выделенного (третьего по порядку) стиля Заголовка (крупно) к исходному тексту

4.4. Познакомьтесь с алгоритмом создания **Оглавления со своим Стилем**.

Установите курсор в конец или начало документа (то есть в месте, где собираетесь разместить оглавление).

На Главной панели выберите вкладку **Ссылки**, найдите пункт (самый левый) **Оглавления**, раскройте выпадающее меню, нажав на плоскую стрелочку вниз и выберите пункт **Настраиваемое оглавление** (рис. 186).

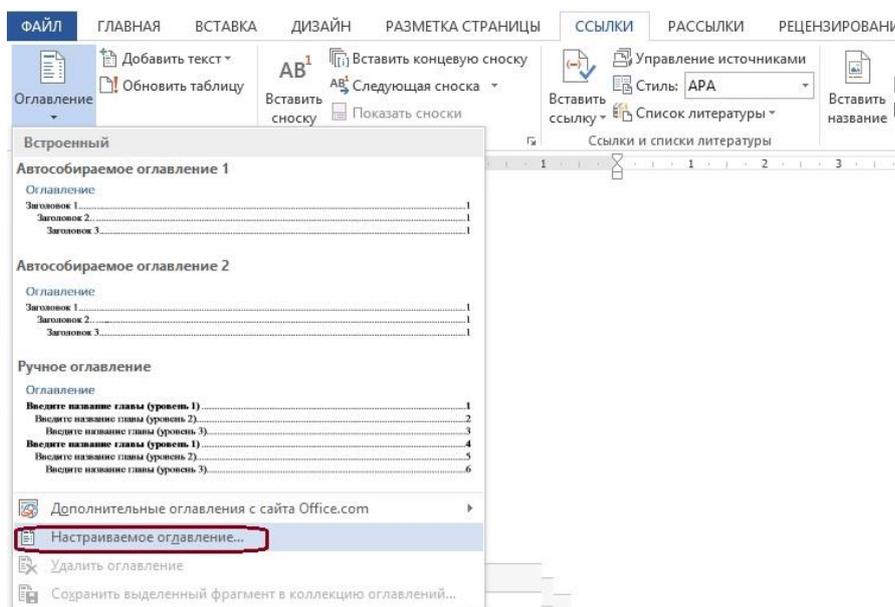


Рис. 186. Выбор **Настраиваемого оглавления** для оформления оглавления

Откроется окно **Оглавление**, в котором представлены различные характеристики представления оглавления документа (рис. 187).

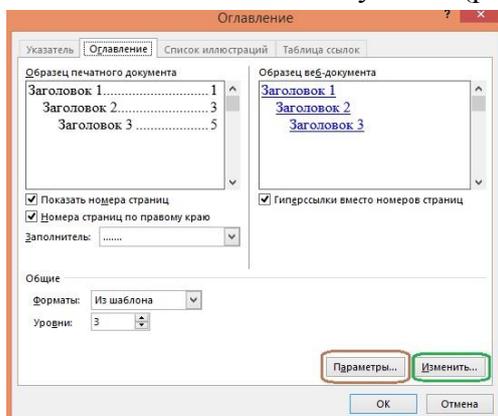


Рис. 187. Характеристики оформления оглавления

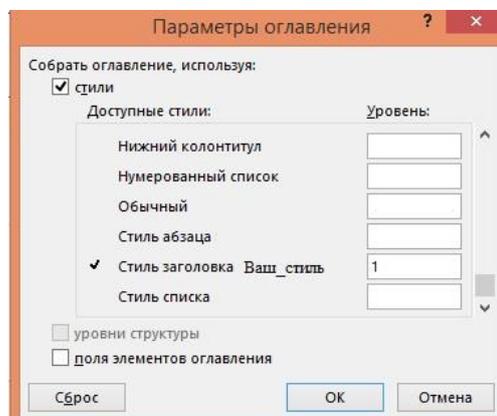


Рис. 188. Параметры сбора элементов оглавления

Нажав кнопку **Параметры** в окне **Оглавление** (рис. 187), установите уровень 1 только для **Стиль заголовка Ваш_стиль** (рис. 188), уровни для остальных стилей удалите.

5. Нумерация страниц.

Для вставки номеров страниц в документ, необходимо выполнить следующие действия **Вставка** → **Номера страниц**, затем выбрать положение номера из предлагаемых вариантов, например, внизу страницы справа (рис. 189).

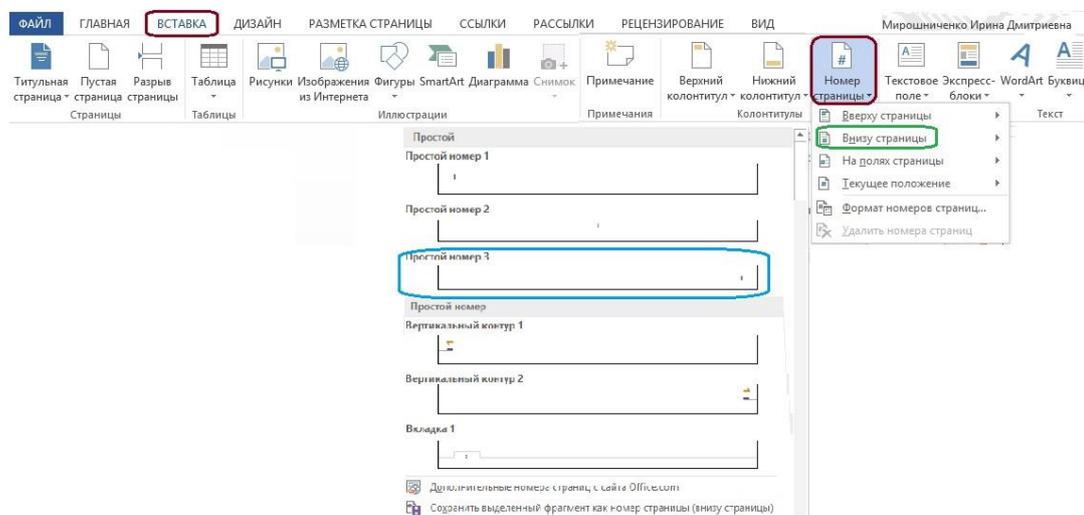


Рис. 189. Выбор вкладки **Вставка** на главной панели, пункта номер страницы и варианта расположения нумерации

6. Создание многоуровневого списка для оглавления.

Выполните создание стиля для первого уровня списка согласно ранее описанному алгоритму. Выберите элемент из понравившегося Вам окна **Стилей**, выберите пункт меню **Создать стиль**.

Определите параметры Вашего стиля.

Имя: Стиль списка **Ваш_стиль**,

Стиль: Абзаца,

Основан на стиле: Обычный.

Для задания параметров шрифта в блоке **Форматирование** выберите пункт **Шрифт** и задайте шрифт: **Times New Roman**, размер **12**, курсив.

Примените **Стиль списка Ваш_стиль** для первого уровня списка. Для второго уровня списка используйте стиль **Заголовок 2**.

При создании многоуровневого списка **обязательно** свяжите первый уровень со стилем **Стиль списка Ваш_стиль**, а второй уровень со стилем **Заголовок 2**.

Сохраните документ под именем «стили».

Пример (рис. 190 - 192).

Конечно-нестационарный·обобщенный·автомат¶

В·определении·конечно-нестационарных·обобщенных·**АВТОМАТОВ**·типа·(2)·форма·их· задания·сформулирована·в·самом·общем·виде.·Вместе·с·тем·в·целях·построения·теории·таких· автоматов· и· использования· их· при· решении· задач· математического· моделирования· целесообразно·выработать·некоторую·единую·оптимальную·форму·задания·этих·**АВТОМАТОВ**,· к·которой·с·помощью·специально·разработанной·процедуры·можно·было·бы·свести·любой· **АВТОМАТ**·типа·(2),·произвольно·заданный·в·соответствии·с·приведенным·определением.¶

2

Рис. 192. Первая страница документа с номером страницы 2

Обобщенное отображение ¶

Задание конечно-нестационарного обобщенного **АВТОМАТА** A (нижний индекс - gv) (2) может, в частности, служить автоматным способом задания соответствующих спектров обобщенных отображений (5). ¶

¶

Понятие об автомате ¶

Конечно-нестационарным обобщенным **АВТОМАТОМ**, заданным над частично упорядоченным полукольцом R , назовем систему A (нижний индекс - gv) где G есть структурный граф автомата (конечный, ориентированный, нагруженный граф), имеющий: ¶

1. конечное множество вершин S с выделенными: ¶

- -- начальной вершиной C (верхний индекс - 0), для которой заданы алфавит состояний A (нижний индекс - 0) и начальное распределение r (вектор-строка с элементами из R) весов этих состояний, подмножеством конечных вершин; ¶

3

Рис. 191. Вторая страница документа с номером страницы 3

2. конечное множество G направленных ребер, соединяющих вершины графа, ребра являются: ¶

• -- направленными ¶

• -- нагруженными; ¶

3. однозначную функцию f , приписывающую каждому ребру одну из элементарных автоматных структур; ¶

4. однозначную функцию g , приписывающую каждой конечной вершине графа одно из финальных распределений весов состояний. ¶

Оглавление

Конечно-нестационарный обобщенный автомат.....	2 ¶
Обобщенное отображение.....	3 ¶
Понятие об автомате.....	3 ¶

4

Рис. 192. Третья страница документа с номером страницы 4

Если текст в дальнейшем будет редактироваться, и нумерация страниц изменится, нужно изменить **Оглавление**. Левой кнопкой мыши щелкните по слову **Оглавление**, над ним появится квадратик с фразой **Обновить таблицу** (рис. 192). Выберите вариант обновления (рис. 193).

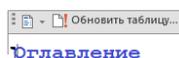


Рис. 193. Изменение Оглавления

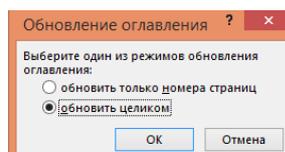


Рис. 194. Выбор варианта обновления

Практическое занятие № 10. Шаблон

Стили и шаблоны (комплексная работа – проверка навыков)

1. Запустите текстовый процессор MS Word.

2. Загрузите файл, содержащий достаточный объем текста, например, курсовую работу, назовите её (*kurs.doc*).

3. Создание стиля абзаца.

Для создания нового стиля абзаца отформатируйте его (практическая №9) должным образом (задайте основной шрифт абзаца, отступ первой строки, межстрочный интервал и т.д.). Выделите отформатированный абзац. Щелкните в поле списка **Стиль** панели форматирования и введите имя нового стиля, например, *abzats*.

Теперь созданный стиль можно применить к другим абзацам. Для этого выделяем абзац и выбираем имя нового стиля (*abzats*) в списке **Стиль** панели инструментов форматирования. Можно использовать форматирование по образцу.

Для установки дополнительных атрибутов стилей абзацев выполняют **Формат** → **Стиль**, выбирают имя стиля и выполняют **Изменить**. Далее можно установить стиль *следующего* абзаца и другие параметры, выполняя **Шрифт**, **Абзац** и т.д.

Аналогичным образом создайте стили для заголовков, списков и т.д.

4. Создание **знакового** стиля. Создайте специальный знаковый стиль для записи *определений*, встречающихся в тексте. Отредактируйте одно из определений, например, установите другой шрифт. Выделите определение, щелкните в поле списка **Стиль** панели форматирования и введите имя нового стиля, например, *opredel*.

Теперь созданный стиль можно применить к другим определениям. Для этого выделяем определение и выбираем имя нового стиля (*opredel*) в списке **Стиль** панели инструментов форматирования. Можно использовать форматирование по образцу.

Сохраните документ под именем *kurs.doc*.

5. Создание **оглавления**.

Для автоматического создания *оглавления* соответствующие заголовки должны иметь **Уровень 1**, **Уровень 2** и т.д. (в меню **Абзац**).

Поместите курсор в начало предполагаемого *оглавления*. Выполните **Вставка** → **Оглавление и указатели**, выберите вариант *оглавления*, формата, установите флажок **Показать номера страниц**, выберите заполнитель (можно переопределить стили для заголовков, используя **Параметры**).

6. Проверка правописания и словари.

Если в курсовой встречаются слова, не содержащиеся в основном словаре, то полезно создать пользовательский словарь. Чтобы его открыть, нужно:

6.1. Открыть **Файл** → **Параметры** → **Правописание** → **Настраиваемые словари**

При нажатии вкладки **Файл**, открывается окно в левом выделенном столбце которого находятся пункты действия с файлом (**Создать**, **Сохранить** и прочее). Самым последним стоит пункт **Параметры** (рис. 195).

Нажмите на пункт **Параметры**, появится окно **Параметры Word**, в котором требуется выбрать пункт **Правописание** (рис. 196).

При нажатии на пункт **Правописание** появится новое окно, которое называется тоже **Параметры Word**, но содержит характеристики параметра **Правописание**, где нужно выбрать пункт **Настраиваемый словарь** (рис. 197).

При нажатии на пункт **Настраиваемый словарь** откроется окно со словарями, в том числе со словарем **Custom.dic** (рис. 198), в которое можно вносить новые пользовательские слова или редактировать существующие (рис. 199).

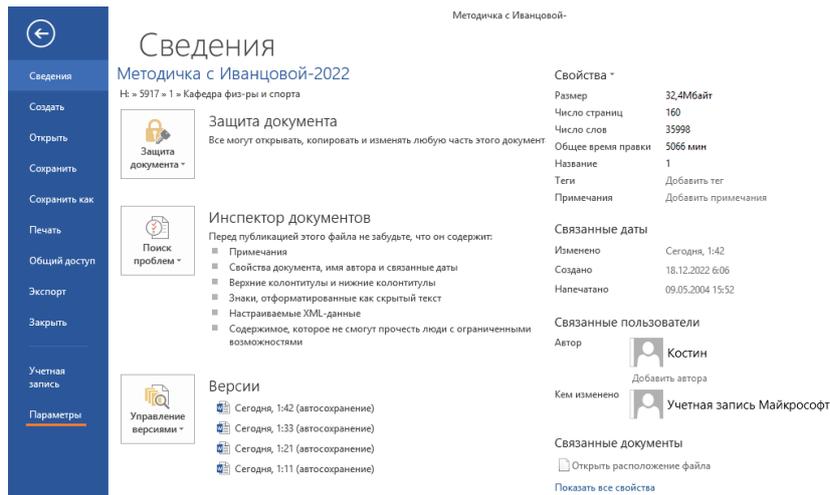


Рис. 195. Вид вкладки после нажатия пункта **Файл**, слева пункт **Параметры**.
Переход по нажатию кнопки **Параметры**.

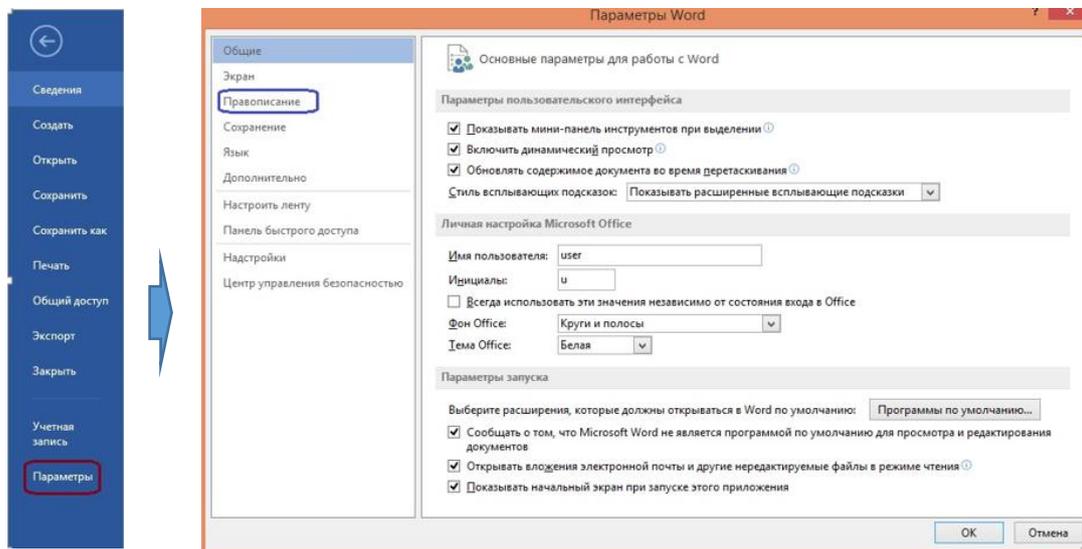


Рис. 196. Вкладка **Параметры Word** (после нажатия пункта **Параметры** в цепочке **Файл**→**Параметры**). Местонахождение пункта **Правописание**.
Пункт **Правописание** открывает новое окно, где нужно выбрать пункт **Настраиваемый словарь**

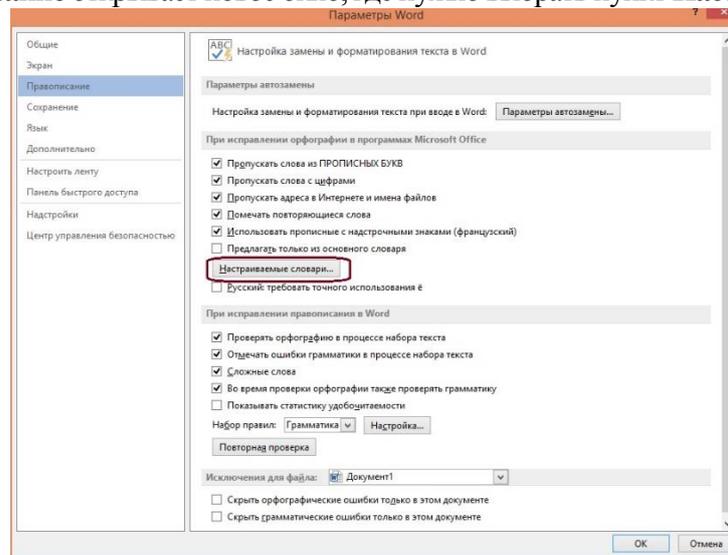


Рис. 197. Окно **Параметры Word** с выделенным пунктом **Настраиваемый Словарь**

6.2. При нажатии на пункт **Настраиваемые словари** открывается окно **Настраиваемые словари**, в котором в блоке **Список словарей** под названием **Все языки** по умолчанию присутствуют два словаря, один из которых называется **Custom.dic** (рис. 198). В этом окне надо выбрать словарь **Custom.dic**.

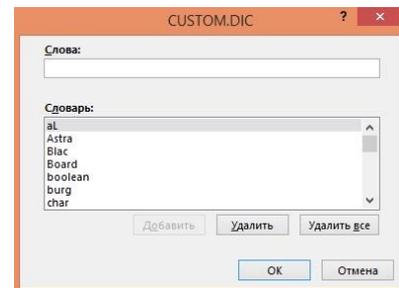
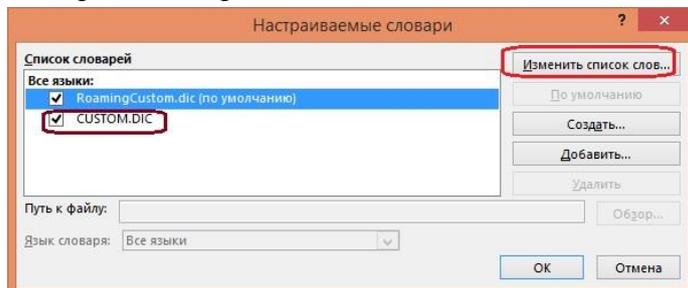


Рис. 198. Выбор словаря **Custom.dic** в окне **Настраиваемые словари** Рис. 199. Добавление слов в словарь

6.3. Для того, чтобы добавить слова в словарь **Custom.dic** или редактировать этот словарь требуется нажать кнопку **Изменить список слов** в окне **Настраиваемые словари** (рис. 199).

В открывшемся окне **Custom.dic** будут содержаться все слова, которые пользователь ранее туда вносил в процессе редактирования документов.

Для создания пользовательского словаря выполните следующие действия: **Открыть Файл** → **Параметры** → **Правописание** → **Настраиваемые словари**. В окне **Настраиваемые словари** нажмите **Создать** (рис. 198). Появится окно **Создание вспомогательного словаря** (рис. 200) введите имя словаря (имя файла) и нажмите **Сохранить**.

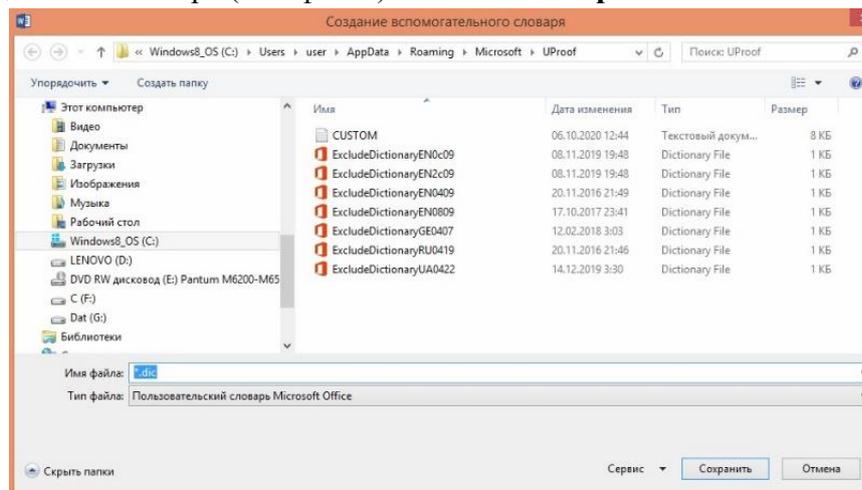


Рис. 200. Создание (сохранение) пользовательского словаря

Чтобы настроить проверку правописания нужно пройти по следующим вкладкам: **Файл** → **Открыть Файл** → **Параметры** → **Правописание** (рис 197). В открывшемся окне **Параметры Word** в списке **Настраиваемые словари** (рис. 198) выберите созданный словарь или в том же окне **Параметры Word** выполните необходимые настройки.

6.4. Для редактирования или проверке словаря пополняйте словами, нажимая на **Добавить**.

Созданные словари можно править или удалять. Выполняется это следующим образом: **Файл** → **Открыть Файл** → **Параметры** → **Правописание** → **Настраиваемые словари** → **Изменить список слов/Удалить** (рис. 198). Исправьте, если необходимо, свой словарь.

6.5. Возобновите автоматическую проверку правописания, выполнив следующие действия: **Файл** → **Открыть Файл** → **Параметры** → **Правописание** (рис. 197), и, установив в разделе **При исправлении орфографии в программах Microsoft Office** флажки в необходимые пункты.

7. Нумерация страниц.

Выполните **Вставка**—>**Номера страниц** и выберите положение, выравнивание, если необходимо, выполните форматирование номера страницы.

Сохраните документ в *kurs.doc*.

8. Создание нового шаблона на базе документа.

Чтобы сохранить все стили и настройки, сделанные в документе, целесообразно создать шаблон на основе данного документа.

Оставьте в документе текст, который будет необходим в качестве шаблона для написания других работ, остальное удалите. Выполните **Файл** —> **Сохранить как**.

В списке **Тип файла** выберите пункт **Шаблон документа**. Выберите папку для хранения нового шаблона (*Templates*), введите имя нового шаблона (*kurs.dot*).

9. Использование шаблона для создания документа.

Выполните **Файл** —> **Создать**, выберите созданный шаблон. Проверьте, что все настройки действуют, набрав небольшой кусок новой курсовой работы. Сохраните документ под именем *kurs1.doc*.

Пример.

Создайте служебную записку, используя встроенный шаблон для записок.

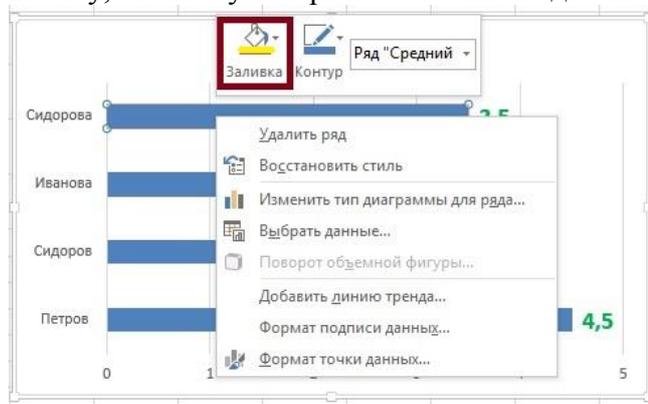


Рис. 201. Параметры форматирования шаблона для записок

Практическое занятие № 11. Графика в Word

Графические возможности

1. Откройте файл «*поет*».
2. Уменьшите левую и правую границы абзаца с названием стихотворения с помощью
 - маркеров на горизонтальной линейке,
 - кнопок сдвига текста  или (рис. 202)
 - параметров отступа слева и справа в блоке **Абзац** на главной панели (рис. 202) и сделайте **Заливку цветом** абзаца (рис. 203).

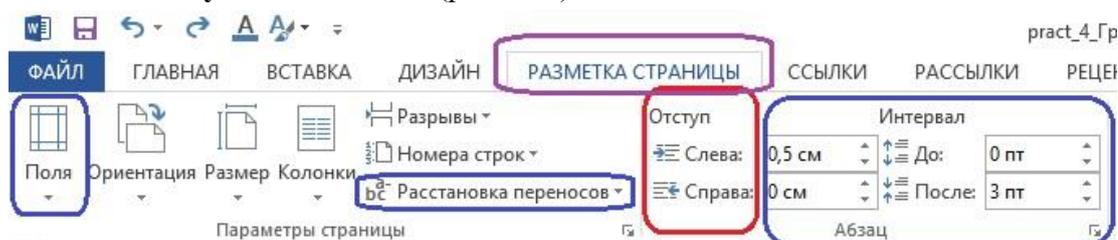


Рис. 202. Часть блоков основной панели, касающаяся размещения текста на странице

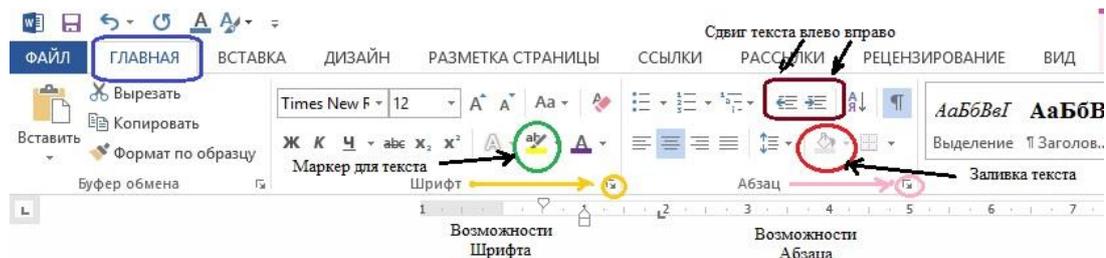


Рис. 203. Часть основной панели, касающаяся размещения и заливки цветом элементов текста

Выделите любое слово цветом, используя **Выделение цветом** (рис. 203).

3. Поэкспериментируйте с разными типами фона и подложки (вкладка **Разметка страницы** рис.202, 203).

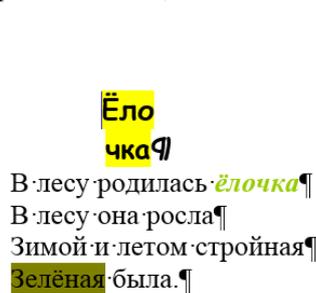


Рис. 204. Вид текста на странице



Рис. 205. Применение различных элементов для изменения цвета

4. Оформление текста готовыми графическими объектами.

Вставьте графический объект, используя библиотеку изображений (вкладка **Вставка**->**Рисунок**—>**Картинки** или **Из файла**). Поместите графический объект в тексте стиха.

Используйте *обтекание текстом*. Для этого щелкните левой кнопкой мыши по графическому объекту, над главной панелью появится пункт **Работа с рисунками** (рис. 206).

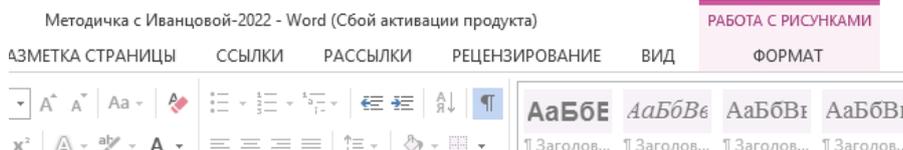


Рис. 206. Часть основной панели с пунктом **Работа с рисунками** над кнопкой **Формат**

Щелкните по пункту **Работа с рисунками**, появится панель (рис. 207), касающаяся различных возможностей работы с рисунками.

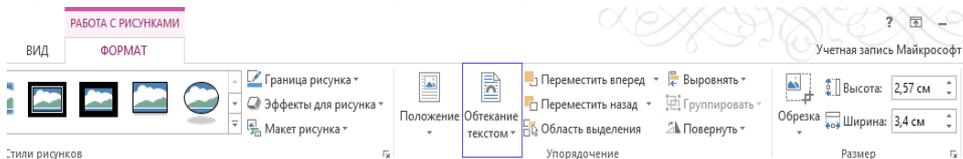


Рис. 207. Панель работы с рисунками с выделенным пунктом **Обтекание текстом**

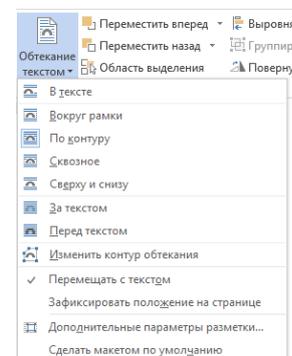


Рис. 208. Параметры обтекания

Выберите пункт **Обтекание текстом** (рис. 208) и выполните действие **Обтекание текстом**—> **По контуру**.

Измените размеры и местоположение рисунка с помощью мыши.

Сохраните документ в своей папке под именем *«poem-image.doc»*.

5. Фигурный текст. Создайте файл «*Table.doc*» (создать таблицу из 3 ячеек в строке (Вкладка **Вставка**—>**Таблица**, рис. 209).

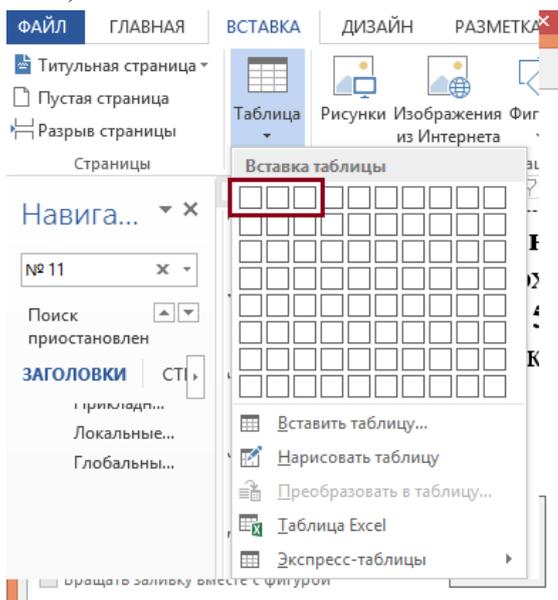


Рис. 209. Создание таблицы с выбором 3 ячеек в строке

В первую ячейку запишите слово **Студент**

Во вторую ячейку вставьте **Объект WordArt** (рис. 2010), имитируя подпись (вкладка **Вставка** ->блок **Текст** значок ).

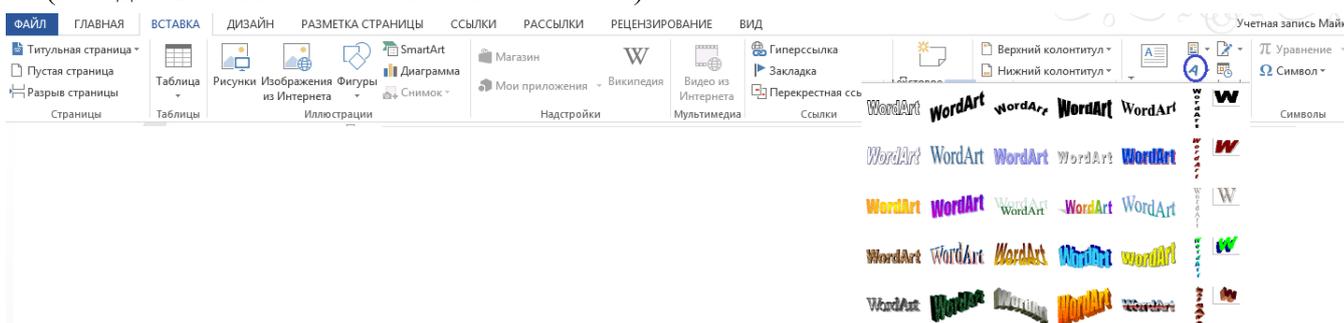


Рис. 206. Панель вкладки **Вставка** с раскрытым пунктом **WordArt** (вариантами фигурной надписи) блока **Текст**

В третью ячейку вставьте ФИО и дату, а также фигуру, имитирующую печать (рис. 211).
Задайте высоту строки 3 см (точно), используя **Свойства таблицы** (**Свойства таблицы** ->**Строка** ->**высота**->**3 см**).

Студент		ФИО 
---------	---	--

Рис. 211. Заполнение таблицы различными объектами

Для этого переведите указатель на таблицу, в левом верхнем углу таблицы появится значок маленького крестика (идентификатора таблицы). Установите курсор на этот крестик, — курсор также превратится в крестик, — нажмите на него левой кнопкой мыши, а потом правой кнопкой мыши.

Появится меню действий с таблицей, выберите в нём пункт **Свойства таблицы**. Появится окно **Свойства таблицы** (рис. 211), в котором выделены кнопки **Таблица**, **Строка**, **Ячейка** **Замещающий текст**.

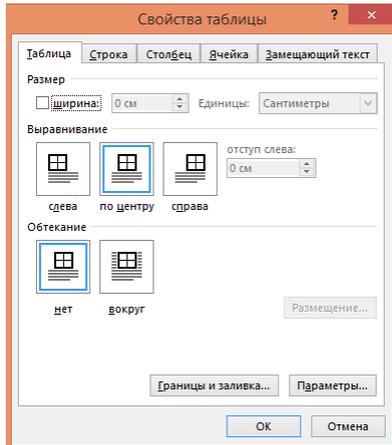


Рис. 212. Окно **Свойства таблицы**

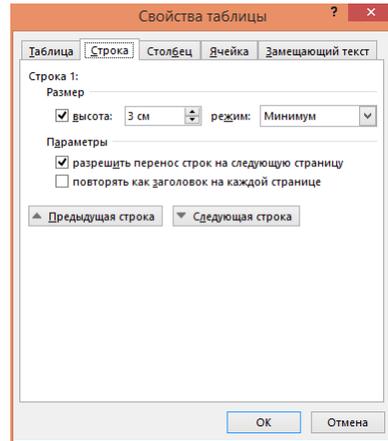


Рис. 213. Окно свойства **Строки**

Выберите пункт **Строка**, нажмите как кнопку, получите свойства **Строки** (рис. 212), в которой измените высоту строки — 3 см

Выворачивайте ширину ячеек. Для этого щелкните левой кнопкой по таблице, появится значок (крестик) выделения таблицы. Установите курсор на этот значок и правой кнопкой мыши щелкните по этому значку, получите выпадающий список **Свойства таблицы**, из которого выберите

- пункт **Выровнять ширину столбцов** либо
- выберите пункт **Столбец** и поочередно установите одинаковую ширину столбца во всех ячейках таблицы.

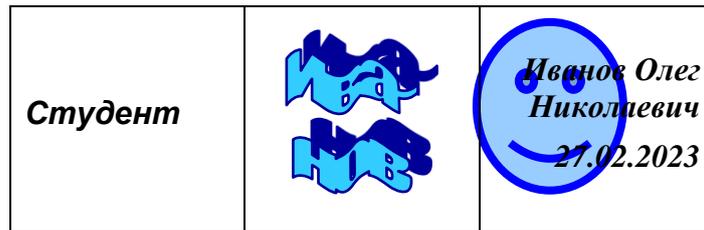


Рис. 214. Таблица с выровненными по ширине столбцами

Вставьте в третью ячейку **Фигуру** (рис. 207, 210) поверх ФИО и Даты, имитируя печать (вкладка **Вставка**->**Фигуры**, рис. 211).

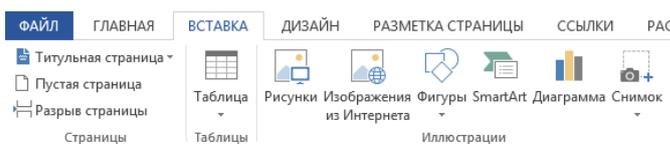


Рис. 215. Пункт **Фигуры** в блоке **Иллюстрации** Главной панели

Для выбранной фигуры (рис. 216) выполните **Обтекание текста** —> **За текстом** (рис. 208, то есть сквозь фигуру будут видны ФИО и Дата).

Для этого выделите фигуру левой кнопкой мыши, затем щелкните правой кнопкой мыши по выделенной фигуре, появится выпадающий список, из которого выбрать пункт **Формат автофигуры** и выполнить действия **Формат автофигуры**->**Макет**->**Дополнительно**->**За текстом**.

Уберите границы таблицы. Для этого снова выделите таблицу, как было рассказано выше, выберите пункт **Свойства таблицы**, откроется окно **Свойства таблицы**, в которой выберите кнопку **Границы и заливка** (рис. 212). Откроется окно **Границы и заливка**, в котором выберите пункт с отсутствием границ (рис. 217).

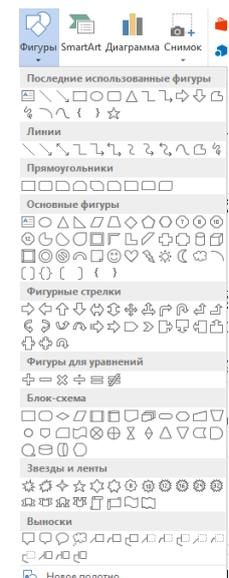


Рис. 216. Виды фигур

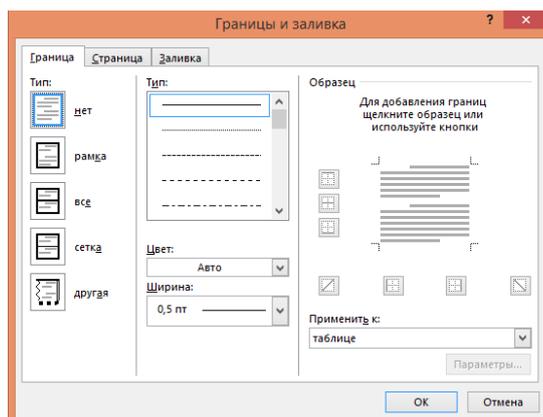


Рис. 217. Окно **Границы и заливка**

Замечание. Важно отметить, что полезно установить дату, которая может быть обновлена до текущей (рис. 219) при нажатии на неё левой клавишей мыши.

Для вставки такой даты можно предложить 2 способа. Но в обоих случаях сначала щелкните по месту в документе, подходящему для установки значений даты и времени.

Первый способ. Вставка даты или времени непосредственно в пункт **Дата и время**.

Откройте вкладку **Вставка** и в блоке **Текст** нажмите на кнопку **Дата и время** (рис. 218). Откроется новое окно, в котором выберите подходящие параметры для отображения формата даты или времени.



Рис. 218. Блок **Текст** и пункт **Дата и Время**



Рис. 219. Обновление до текущего времени

При этом в поле **Форматы** окна **Дата и время** представлены различные типы записей, подходящие для русского языка: несколько форматов даты, даты и времени, только времени. Доступны всевозможные варианты дат с использованием числа, месяца, года, дня недели, и времени в часах, минутах и секундах. Чаще всего используется самый распространенный формат: «ДД.ММ.ГГ» (дата, месяц, год).

Второй способ. Вставка даты или времени с помощью горячих клавиш.

Нажмите на сочетания клавиш:

- **Alt + Shift + T** — вставка времени в заданном по умолчанию формате.
- **Alt + Shift + D** — вставка даты в формате по умолчанию.

Установите дату с автоматическим обновлением (рис. 219).

Сохраните файл (*Table.doc*).

6. Надпись и Буквица. Откройте файл «*Zayavlenie*». Измените заявление (рис. 220).

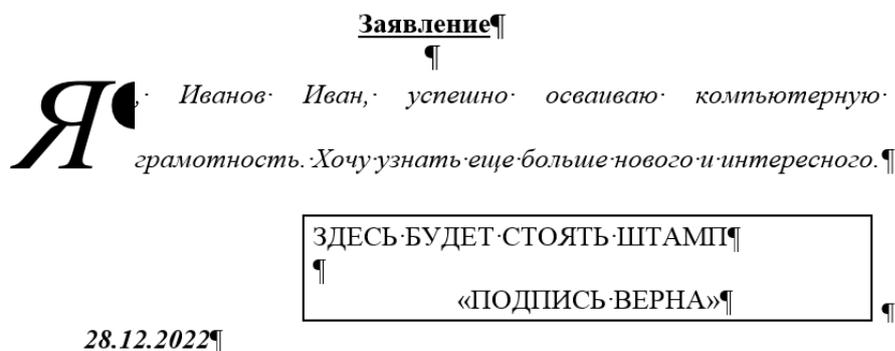


Рис. 220. Вид **Заявления** с буквицей, надписью и обновляемой датой

Добавьте буквицу (**Я**) в заявление для демонстрации оформления текста с буквицей.

Для этого установите курсор в начале нужного абзаца (в нашем случае в начале текста собственно **Заявления**).

На вкладке **Вставка** (рис. 221) в блоке **Текст** выберите значок , обозначающий **Буквицу** (рис. 222).

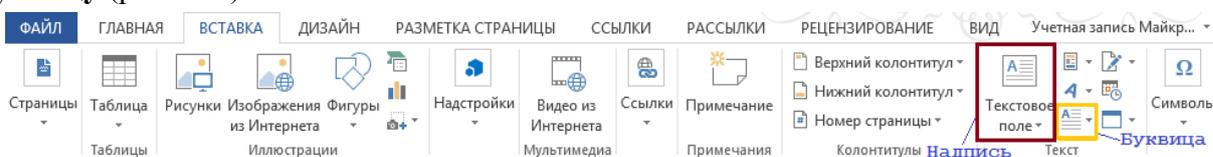


Рис. 221. Вкладка **Вставка** блок **Текст** пункт **Текстовое поле**

Выберите нужный вариант размещения знака **Буквицы** (*В тексте, На поле* и есть возможность индивидуальной настройки через *Параметры буквицы*, рис. 222), нажмите **Enter**.



Рис. 222. Изображение пункта **Буквица**

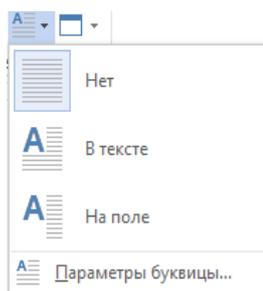


Рис. 223. Размещение **Буквицы** в тексте

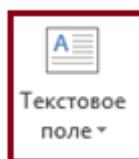


Рис. 224. Изображение пункта для **Надписи**

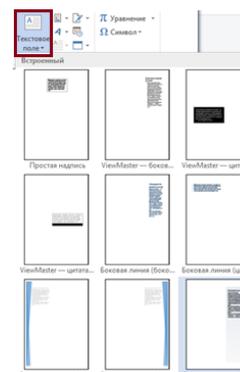


Рис. 225. Размещение пункта **Надпись** в тексте

Ниже текста заявления изобразите штамп, используя инструмент **Текстовое поле** (**Надпись**, рис. 224).

Надпись – это специальное поле, куда можно добавить текст или рисунок.

Установите курсор в требуемое место (рис. 225) текста абзаца и нажмите значок **Надпись** (на вкладке **Вставка**->**Текстовое поле** (**Надпись**)->выбор варианта размещения и вида надписи). Появится рамка с дежурным текстом, поясняющим, как использовать инструмент **Надпись** (рис. 226).

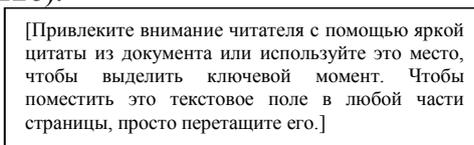


Рис. 226. Рамка **Надписи** и текст пояснения внутри

Вставьте нужный текст или рисунок виде текста (как выше), задавая его размеры протягиванием мыши.

7. Набор математических выражений.

Откройте файл «**formuly**». Добавьте недостающие формулы, используя специальный инструмент **Уравнение**  на панели **Вставка** (рис. 227).

Выполните действие **Вставка**->**Уравнение**.



Рис. 227. Расположение пункта **Уравнение** на вкладке **Вставка**

Обратите инструмент **Уравнение** работает в версиях Word выше 2003 года. Если у Вас файл сохранен, как документ MS WORD 1997-2003, его надо пересохранить как Документ WORD последних версий. Инструмент **Уравнение** станет активным.

Если нажать на пункт **Уравнение**, получим специальную панель (рис. 228), с помощью кнопок которой можно записывать сложные математические выражения.

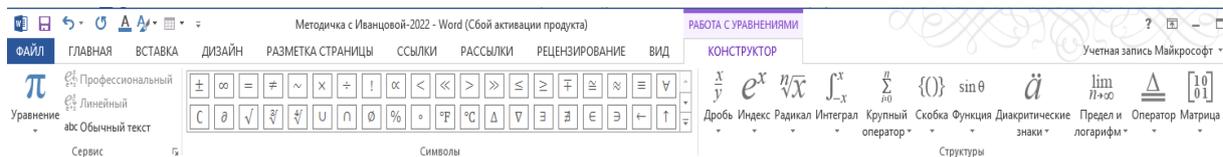


Рис. 228. Спецпанель пункта **Уравнение** для привычного написания математических формул

С помощью инструмента **Уравнение** выполнить запись ниже представленных формул. На вкладке **Вставка** нажмите пункт **Уравнение**, откройте спецпанель, наберите требуемые математические выражения (рис. 229).

Квадратное уравнение: $ax^2 + bx + c = 0$ ¶
 Азотная кислота: HNO_3 ¶
 $ax^2 + bx + c = 0$ ¶
 $3 \leq 4$ ¶
 $k = 1, 2, \dots$ ¶
 $f'(x)$ ¶
 $\alpha + \beta$ ¶
 $\sqrt{2}$ ¶

$\sum_{k=1}^{\infty} x^k$ ¶
 $\int_a^b f(x) dx$ ¶
 $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

Рис. 229. Вид математических выражений, набранных с помощью инструмента **Уравнение**

Сохраните файл **«formuly»** под тем же именем в своей папке.

8. Вставка гиперссылок.

Откройте файл **«поет»**. Установите гиперссылку на файл **«Zayavlenie»**.

Для этого в тексте стихотворения выделите объект (например, ранее вставленную картинку, текст) и на вкладке **Вставка** блока **Ссылки** нажмите пункт **Гиперссылка** (рис.230).



Рис. 230. Пункт **Гиперссылка** на вкладке **Вставка** блока **Ссылки**

Появится диалоговое окно выбора файла для вставки гиперссылки (рис. 231).

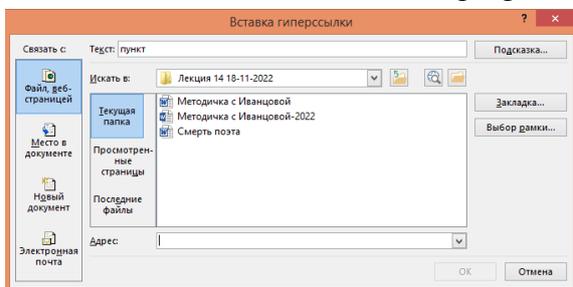


Рис. 231. Диалоговое окно выбора файла для вставки **Гиперссылки**

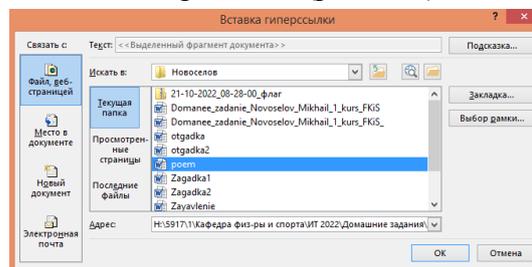


Рис. 232. Выбор файла для ссылки в окне **Вставка гиперссылки**

В появившемся диалоговом окне выберите имя файла — **«Zayavlenie»** (в примере – мои файлы). То есть, в окне **Вставка гиперссылки** (рис. 232) находим требуемый файл, на который будем ссылаться.

Так, в файле заявления выделите свою фамилию и на вкладке **Вставка** нажмите **Гиперссылка**. В появившемся диалоговом окне выберете имя файла — *«poem»*.

Проведите тестирование, то есть проверьте, что по нажатию на фамилию как на ссылку, открывается файл *poem*, то есть Ваше стихотворение.

Сохраните документ под именем *poem-gs*.

9. Оформление колонтитулов.

Откройте файл *«poem»*.

Для установки колонтитулов на вкладке **Вставка** нажмите **Колонтитулы**.



Рис. 233. Расположение **Колонтитулов** на вкладке **Вставка**

При нажатии на верхний/нижний колонтитул предлагаются варианты расположения колонтитулов (например, слева, по центру, справа и т. п.). Выбирается требуемый вариант расположения, размера, вида текста или рисунка (объекта).

Задайте *верхний* колонтитул следующим образом: выбранное Вами имя файла (размер текста 8 пт, выравнивание по левому краю страницы).

Перейдите к *нижнему* колонтитулу, нажав кнопку **нижний колонтитул**.

Задайте нижний колонтитул: дата (размер 8 пт, выравнивание по правому краю страницы).

Для выхода из режима колонтитулов щелкните по кнопке **Закреть** либо дважды щелкните по основному полю страницы.

Сохраните документ под именем *«poem»*.

Практическое занятие № 12. Колонтитулы

Колонтитулы и дополнительные графические возможности

1. Запустите текстовый процессор **MS Word**.

2. Оформление колонтитулов.

Загрузите файл *poem.doc* или *zagadki.doc*.

Для установки колонтитулов используйте **Вставка** —> **Колонтитулы** или дважды щелкните по верхнему или нижнему пространству документа для колонтитулов.

Внимательно изучите меню для работы с колонтитулами.

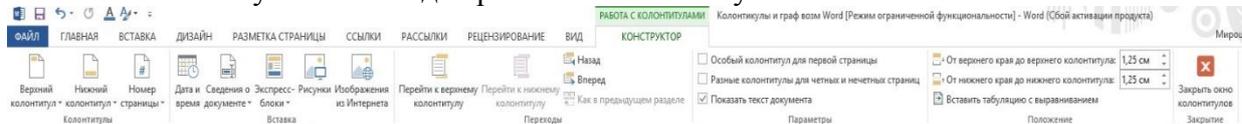


Рис. 234. Панель работы с колонтитулами

Задайте следующее оформление верхнего колонтитула:

- ♦ полное имя файла (размер 8, выравнивание по левому краю страницы) и
- ♦ имя автора (размер 10, полужирный курсив, выравнивание по правому краю страницы).

Для выравнивания по правому краю дважды нажмите клавишу **Tab**. Переход к нижнему колонтитулу осуществляется нажатием кнопки **Верхний/нижний колонтитул**.

Задайте следующее оформление нижнего колонтитула:

- ♦ дата (размер 8, выравнивание по левому краю страницы) и
- ♦ номер страницы в формате стр. № (размер 10, выравнивание по правому краю страницы).

Для выхода из режима колонтитулов щелкните по кнопке **Заккрыть** или дважды щелкните по основному пространству документа.

Сохраните документ под именем *kolontit.doc* в своей папке.

3. Создание водяных знаков. Загрузите файл *poem.doc* или *zagadki.doc*.

Найдите вкладку **Дизайн** на главной панели, перейдите в группу **Фон страницы**, выбрав пункт **Подложка (Дизайн-> Фон страницы-> Подложка)**.

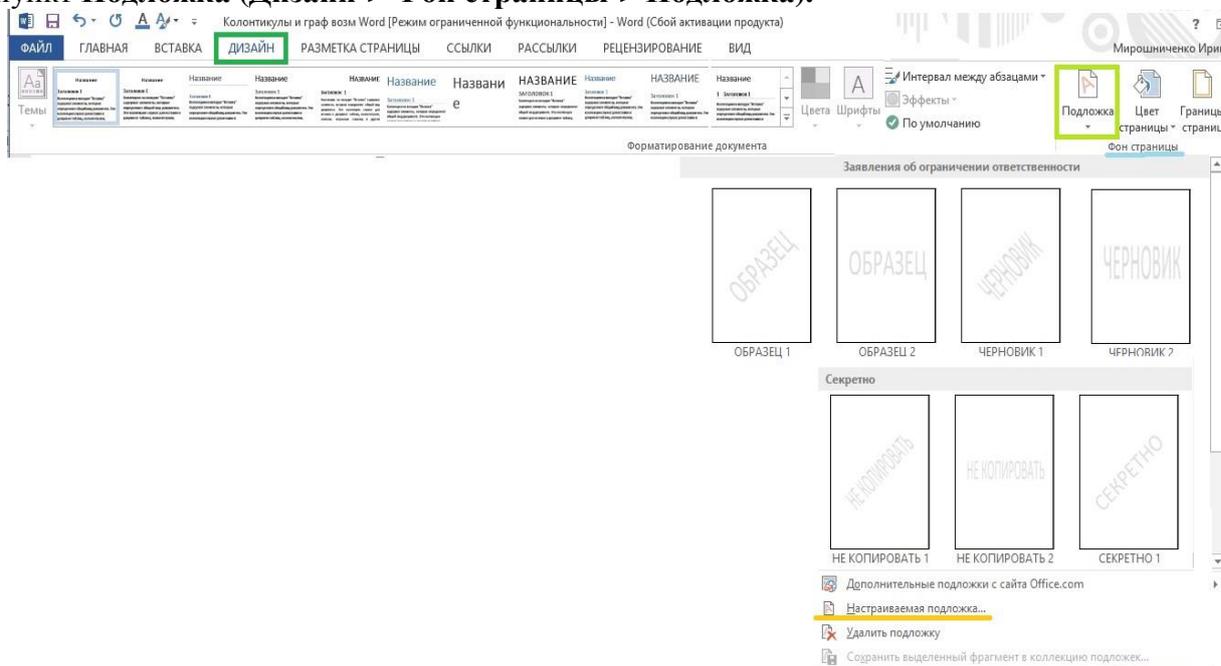


Рис. 235. Пункт **Подложка** в блоке **Фон страницы** вкладки **Дизайн**

В выпадающем списке выбрать подходящую подложку для документа и щелкнуть мышкой по ней мышью для получения нужного шаблона.

При желании, вместо выбора готовых шаблонов можно в меню выбрать **Настраиваемая подложка** и задать собственные необходимые параметры нового изображения или текста, которые планируется использовать в качестве “водяного знака”.

Сохраните документ под именем *vod_znak.doc* в своей папке.

- **Оформление текста готовыми графическими объектами.** Загрузите файл *poem.doc* или *zagadki.doc*.

Можно воспользоваться своей библиотекой изображений (**Вставка** —> **Рисунки**) и далее выбирать рисунок из нужной папки или взять изображение из Интернета (**Вставка** —> **Изображения из интернета**). Вставьте картинки и разместите их в документе подходящим образом (изменить размеры и местоположение рисунка можно с помощью мыши), применяя контекстное меню рисунка и выбирая **Формат рисунка**.

Используйте для рисунков различное обтекание текстом (**Формат рисунка** —> **Макет**—> **Положение** —> **Обтекание** —> **Дополнительно**-><вид обтекания>).

Попробуйте поработать с различными фигурами (**Вставка** —> **Фигуры**), изменяя их форму, объекты SmartArt (**Вставка** —>**SmartArt**) и диаграммы (**Вставка** —> **Диаграммы**).

Сохраните документ под именем *risunok.doc* в своей папке.

- **Оформление визитки графическими объектами или рисунком из файла.**

На новом документе, используя вкладку **Разметка страницы**, настройте параметры страницы: поля — 1,5 см. Ориентация — книжная.

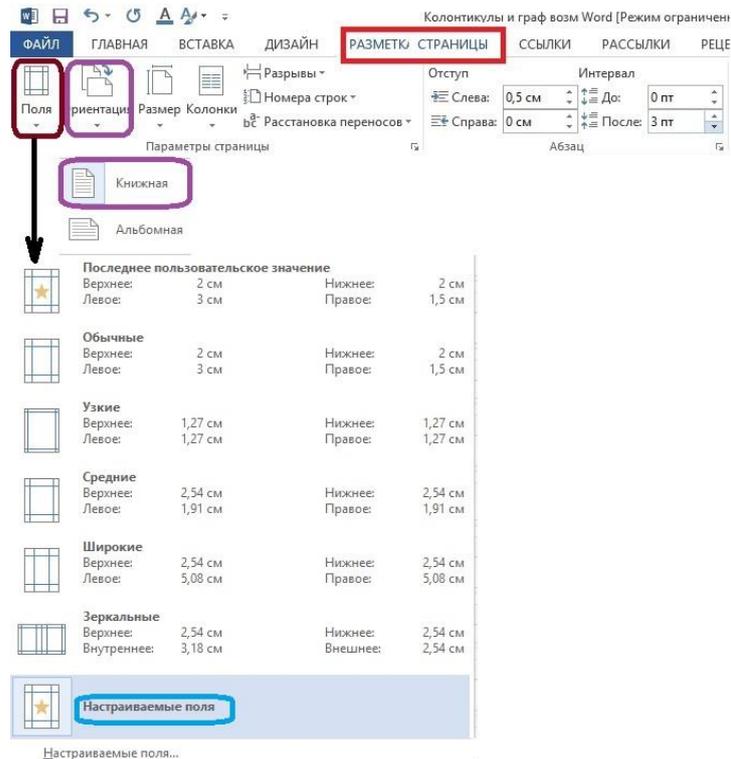


Рис. 236. Вкладка **Разметка страницы**, пункты **Ориентация** и **Поля** и их варианты форматирования. Вставьте подходящий рисунок (**Вставка** → **Рисунок** (из файла)). Используя пункт контекстного меню **Формат рисунка**, установите размеры 4×3,5 см.

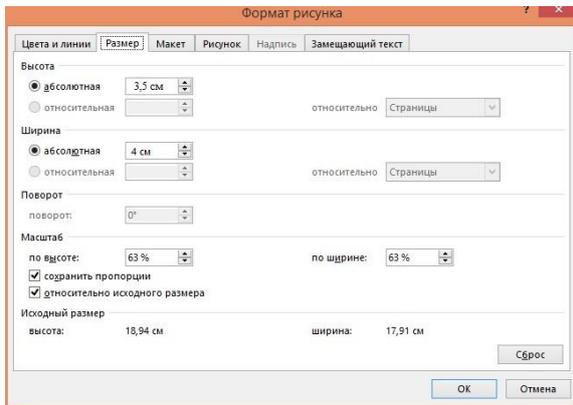


Рис. 237. Формат рисунка. Размер рисунка

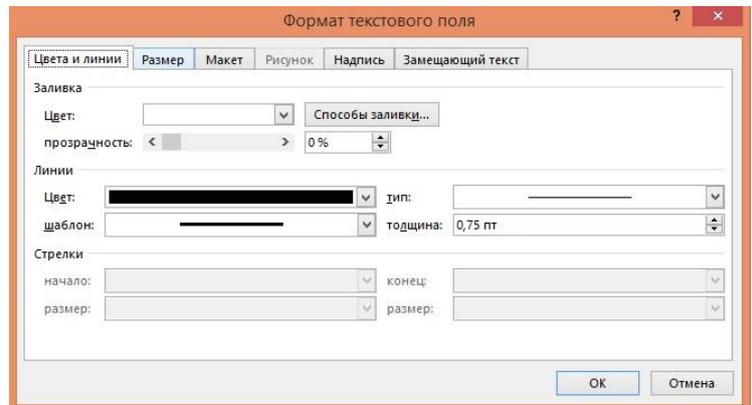


Рис. 238. Формат текстового поля.

Ниже рисунка вставить объект **Надпись** (**Вставить** → **Надпись**) и растянуть объект **Надпись** до размеров 9×5 см (стандартные размеры визитки). Изменение размеров рисунка контролируйте по горизонтальной и вертикальной линейкам.

Переместите рисунок внутрь рамки визитки в левый верхний угол. Поместите рисунок *на передний план* (**Формат рисунка** → **Макет** → **Перед текстом**).



Рис. 239. Исходный рисунок

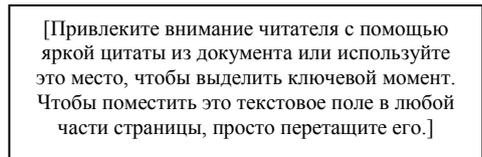


Рис. 240. Исходное Текстовое поле

Внутри первого (объемлющего) **Текстового поля** вставьте правее рисунка еще одно **Текстовое поле** 5×3,5 см. Занесите туда данные: фамилия, имя, отчество и название группы. Разместите по центру, примените шрифтовое оформление.

Ниже рисунка вставьте еще одно **Текстовое поле** размером 9×1,5 см. Занесите туда данные: адрес, телефон. Вставьте символы конверта и телефона, примените шрифтовое оформление.



Рис. 240. Вид визитки, собранной из рисунка, текстовых полей и спецсимволов
Для справки: где найти символы телефона и письма, а также многих других графических символов (**Вставка->Символ**, рис. 241).

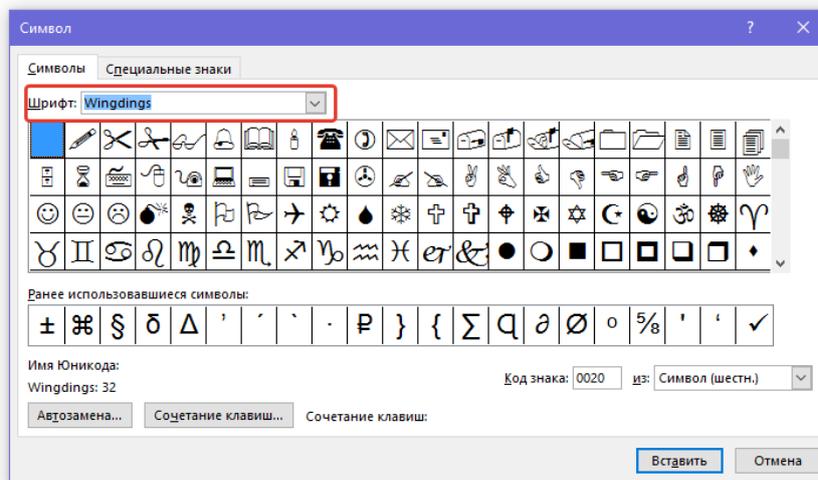


Рис. 241. Шрифт Winding и содержащиеся в нем графические символы
Сгруппируйте рисунок и все **Текстовые поля**. Получим следующий объект, представляющий композицию четырех объектов.

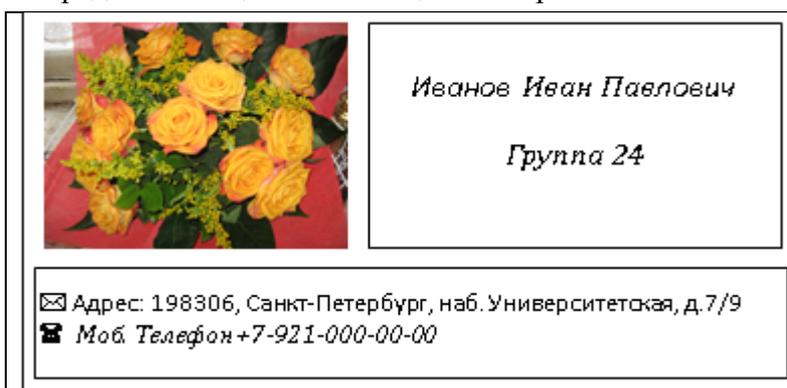


Рис. 242. Визитка как единый объект

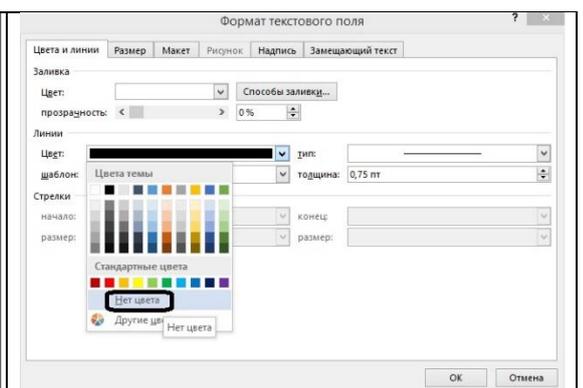


Рис. 243. Выбор пункта Нет цвета

Теперь нужно убрать цвет линий рамки **Текстового поля**: выбираем пункт **Нет цвета** (рисунок справа, рис. 243)

Для удаления рамок **Текстового поля** выделите нужную рамку правой кнопкой вызовите контекстное меню и выберите пункт **Формат текстового поля**. Во вкладках появившегося окна выделите **Цвета и линии** и в пункте **Линии** —> **Цвет** выберите «**Нет цвета**».

Выполните эту операцию для обоих **Текстовых полей**.

Затем выделите последовательно рисунок (держа нажатой клавишу **Ctrl**), текстовое поле с ФИО, текстовое поле с адресом и объемлющее текстовое поле, войдите в контекстное меню и выберите пункт меню **Группировать**. Получите уже связанный объект, части которого не будут смещаться относительно друг друга.



Рис. 244. Вид визитки после удаления границ внутренних полей

Для того, чтобы сделать целую страницу визиток, опустите визитку чуть ниже, вставив перед ней пустую строку (**ENTER**). Затем вставьте таблицу размером 5x2 (5 строк и 2 столбца). Выделите визитку и вставьте её в первую левую ячейку таблицы. Вставьте её же во вторую ячейку (правую в первой строке). Выделите полученную строку и скопируйте её. Затем выделите четыре оставшиеся строки (пустые) и вставьте скопированную строку во все четыре строки за один раз. Осталось только удалить границы таблицы (хотя этого можно и не делать, так как по ним удобно разрезать визитки). Получилось десять визиток.



Рис. 245. Комплект визиток для печати

Сохраните документ под именем *vizitka2.doc* в своей папке.

Замечание. Чтобы было удобнее разрезать, перед тем, как вставлять рисунок в ячейку, вставьте пустую строку размером, например, 3 пт, после рисунка сделайте отступ в 3 пт.

На рисунке сделано именно так.

- **Редактирование готового графического объекта.**

Создайте произвольный объект из нескольких элементов (так, как создавали визитку), вплоть до момента сгруппировать (включая действие сгруппировать). Если необходимо изменить положение, размеры, топологию этих элементов объекта, выделите объект

(который у Вас был сгруппирован) и в контекстном меню для него найдите пункт **Разгруппировать**.

Например, создали объект из четырех фигур: трех кругов и треугольника (треугольники немного изменили с помощью меток, чтобы его повернуть, вытянуть), верхний круг вывели на передний край, чтобы спрятать неровности треугольника. Затем его сгруппировали (рисунок слева, рис. 246). Сгруппированный рисунок скопировали и вставили (справа). Теперь его можно разгруппировать и отдельно изменять части.

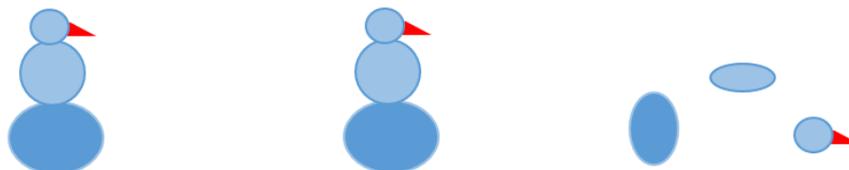


Рис. 246. Демонстрация работы с графическими объектами, принцип группировки и разгруппировки

Теперь над отдельными элементами объекта можно производить различные действия, которые позволяют маркеры, например, повернуть/отразить, растянуть по разным маркерам, добавить надпись, изменить цвет линий и заливки и т. п.). Когда все необходимые на ваш взгляд изменения внесены, *сгруппируйте* все элементы снова с помощью выделения элементов при нажатой клавише **Shift** или **CTRL** и выполните действие **Группировать**, выбрав его из контекстного меню Вашего объекта (**Формат->Группировать**).

Сохраните документ под именем *risun1.doc* в своей папке.

- **Фигурный текст**. Создайте документ формата А5 (148 × 210 мм). Представьте в нем свой ник, девиз и эмблему. Эмблему можно взять из своих изображений или из Интернета. Ник и девиз исполните фигурным текстом.

Для этого поместите курсор в место, где будет расположен фигурный текст **WordArt**. Откройте вкладку **Вставка**, в блоке **Текст** появится элемент «**WordArt**».

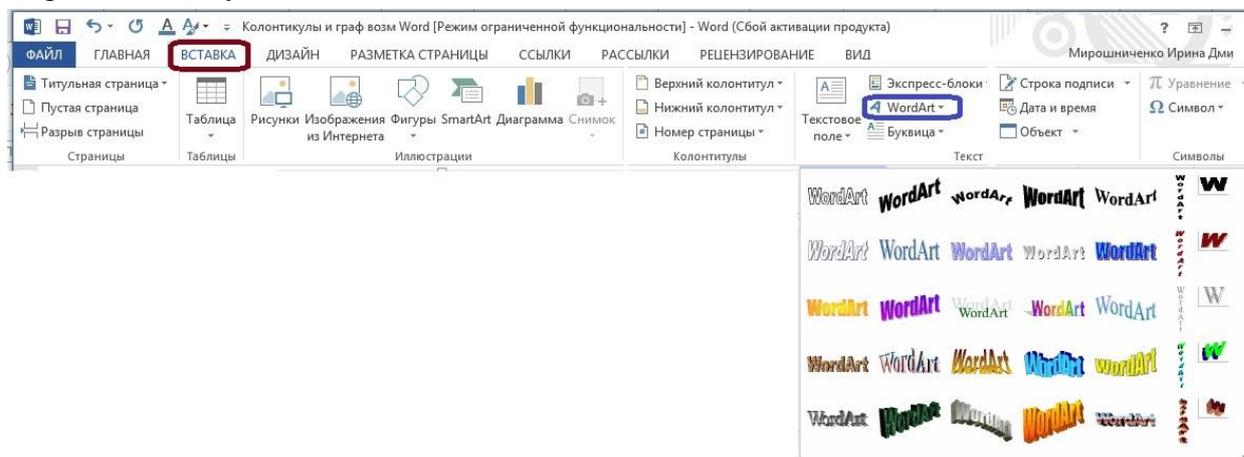


Рис. 247. Выбор элемента **WordArt** из блока **Текст** вкладки **Вставка**. Представление библиотеки **WordArt**

Нажмите кнопку **WordArt** и в открывшемся меню выберите один из шаблонов **Стиля надписи**. В области редактирования содержимого объекта **WordArt** введите нужный текст вместо «Поместите здесь ваш текст», задайте размер объекту **WordArt**, гарнитуру и начертание.



Рис. 248. Элемент WordArt, видоизмененный с помощью маркеров и залитый градиентной заливкой

Изучите дополнительные возможности (**Формат объекта WordArt** или на панели **WordArt**), примените понравившиеся возможности.

Сохраните документ под именем *nickname.doc* в своей папке.

Практическое занятие № 13. MS Excel.

Знакомство с адресацией в MS Excel

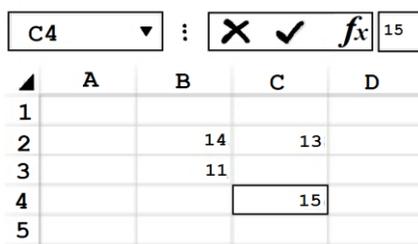
Напомним основные сведения об адресации в MS Excel. Понятие адресации — одно из центральных в Excel: рассматривают абсолютную и относительную адресацию.

Как известно, Лист Excel представляет двумерную матрицу ячеек, где ячейки одного столбца имеют одинаковую ширину для всех своих значений, а ячейки одной строки имеют одинаковую высоту для всех своих значений (в отличие от ячеек таблицы в Word).

В зависимости от версии операционной системы и разрядности (32 или 64 битной карты) количество ячеек по горизонтали и вертикали отличается. Так, например, в MS Office 10 (64 разрядной) последняя ячейка по горизонтали имеет номер **XFD**, а по вертикали — **1 048 576**.

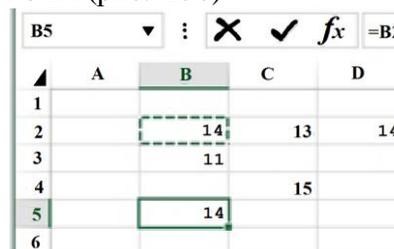
В виду того, что Лист Excel представляет двумерную матрицу ячеек, адреса ячеек нужно конструировать из двух номеров: номера по горизонтали и номера по вертикали (именно в таком порядке), например, **A14** — это ячейка с номером **A** по горизонтали и номером **14** по вертикали. Однако адресация в Excel организована гораздо сложнее. Она связана с топологией движения по двумерной матрице при копировании адреса в некоторую другую ячейку. Адрес **A14** называется *относительным* адресом (адресом относительно верхней левой точки Листа, которую условно можно считать адресом **00**).

Рассмотрим пример, в котором в ячейку **B2** поместили значение 14, в ячейку **C2** — 13, в ячейку **B3** — 11 и в ячейку **C4** — значение 15 (рис. 249). Так, если выделить и скопировать ячейку **B2** в ячейку **D2** или **B5**, то есть скопировать **значения** ячейки **B2**, то в указанных ячейках получим именно значение ячейки **B2**, а именно **14** (рис. 250).



	A	B	C	D
1				
2		14	13	
3		11		
4			15	
5				

Рис. 249. Исходное заполнение ячеек



	A	B	C	D
1				
2		14	13	14
3		11		
4			15	
5		14		
6				

Рис. 250.

А вот если в ячейку **B5** записать адрес ячейки **B2** (**=B2**), то в ячейке **B5** получим значение **14**, которое содержится также в ячейке **B2**. Но надо понимать, что само значение 14 содержится в **B2**, и к нему в ячейке и **B5** обращаемся через *адрес* ячейки (**B2**), в которой он содержится.

Если теперь скопировать содержимое ячейки **B5** (адрес ячейки **B2**) в ячейку **B6** (спустились вниз на 3 ячейки), получим в качестве содержимого адрес **B(2+3)**, то есть **B5**. А по адресу **B5** (**=B5**) находится значение **11**. Видим, что в ячейке **B6** находится значение **11**. Топологически, если сдвигаемся по вертикали вверх или вниз на n ячеек, то для результирующей ячейки адрес, содержащийся в исходной ячейке, соответственно, изменяется на $-n$ или n единиц числового параметра адреса.

Если же скопировать содержимое ячейки **B3** (в ней теперь адрес ячейки **B2**) в ячейку **E3** (сдвинулись вправо на 3 ячейки), получим в результирующей ячейке в качестве содержимого ячейки **E3** адрес (**B+3 буквы**) **2**, то есть **E2**. А по адресу **E2** (**=E2**) находится значение **11**. Видим, что в ячейке **E3** находится адрес **E2** (**=E2**), а по этому адресу в ячейке содержится значение **11**. Топологически, если сдвигаемся по горизонтали вправо или влево на n ячеек, адрес, содержащийся в исходной ячейке, соответственно, изменяется на $-n$ или n букв символьного параметра адреса.

Если же скопировать содержимое ячейки **B3** (адрес ячейки **B2**) в ячейку **E6** (сдвинулись вниз на 3 ячейки и вправо на 3 ячейки), получим в качестве содержимого ячейки **E6** адрес (**B+3 буквы**)(**2+3**), то есть **E5**. А по адресу **E5** (**=E5**) находится значение **8**. Видим, что в ячейке **E6** находится адрес **E5** (**=E5**), а по этому адресу находится значение **8**.

Топологически, если сдвигаемся по какой-то траектории, например, по горизонтали вправо или влево на n ячеек и по вертикали вверх или вниз на m ячеек, то адрес, содержащийся в исходной ячейке изменяется по такой же траектории, а именно, адрес в результирующей ячейке, соответственно, изменяется на $-n$ или n букв символического параметра адреса и на $-m$ или m единиц числового параметра адреса.

	A	B	C	D	E
1					
2		14			10
3		14			10
4			11		
5		11	15		8
6		11			8
7					

Рис. 251. Пример применения абсолютной адресации к ячейкам B2-> B3, B5-> B6, E2-> E3, E5-> E6

	A	B	C	D	E
1					
2			14		10
3		14			14
4			11		
5		11	15		8
6		14			14
7					

Рис. 252. Пример применения абсолютной адресации к ячейкам B2-> B3, B3-> E3, B2-> E3, B3-> E6

Кроме относительного адреса в Excel существует понятие *абсолютного* адреса, то есть такого адреса, который не изменяется ни при какой топологии движения при копировании адреса. Этому добиваются записью знака \$ перед обоими параметрами адреса ячейки, например, \$B\$2.

Так если в ячейку B3 записать адрес \$B\$2 (=B\$2), ячейка B3 будет ссылаться на значение, содержащееся в ячейке B2, то есть 14 (рис. 251).

Если же в ячейку E3 скопировать адрес, содержащийся в ячейке B3 (=B\$2), ячейка E3 будет ссылаться на содержимое, содержащееся в ячейке B3, а это адрес B2 (=B\$2), то есть 14 (рис. 252).

Аналогичным образом, если в ячейку B6 скопировать адрес, содержащийся в ячейке B3 (=B\$2), ячейка B6 будет ссылаться на содержимое, содержащееся в ячейке B3, а это адрес B2 (=B\$2), то есть 14.

Более того, если в ячейку E6 скопировать адрес, содержащийся в ячейке B3 (=B\$2), ячейка E6 также будет ссылаться на содержимое, содержащееся в ячейке B3, а это адрес B2 (=B\$2), то есть 14.

То есть, какова бы ни была траектория копирования ячейки с абсолютным адресом, в результирующей ячейке будет всегда значение, содержащееся в исходной ячейке (на которую ссылаются абсолютным адресом).

Существует и адресация промежуточного вида, когда замораживается движение по одному из направлений, например, по вертикали (B\$2) или по горизонтали (\$B2). Такая адресация также очень часто используется.

При замораживании адресации по вертикали (B\$2), какое бы ни было движение копирования вверх или вниз, значение всегда будет братья со 2 строки, а движение по горизонтали будет изменяться, как описано выше.

	A	B	C	D	E
1					
2		14			10
3		14			10
4			11		
5		11	15		8
6		14			10
7					

Рис. 253. Пример оперирования с относительной адресацией (закрепление по числовому аргументу)

	A	B	C	D	E
1					
2		14			10
3		14			14
4			11		
5		11	15		8
6		11			11
7					

Рис. 254. Пример оперирования с относительной адресацией (закрепление по буквенному аргументу)

По рисунку выше видно, что копирование содержимого ячейки B3 (=B\$2) в ячейку B6 дает содержимое ячейки B3 (=B2), а именно значения 14 (рис. 253).

Копирование содержимого ячейки **B3 (=B\$2)** в ячейку **E3** дает содержимое ячейки **E3 (= (B+3 буква)2)**, то есть **E2**, а именно значения **10** (рис. 253).

Копирование содержимого ячейки **B3 (=B\$2)** в ячейку **E6** дает содержимое ячейки **E2 (= (B+3 буква)2)**, то есть **E2**, а именно значения **10** (потому что по вертикали сдвигаться не можем, а по горизонтали можем, рис. 253).

При замораживании адресации по горизонтали (**\$B2**), какое бы ни было движение копирования влево или вправо, значение всегда будет браться со столбца **B**, а движение по вертикали будет изменяться, как описано выше (рис. 254).

По рисунку выше видно, что копирование содержимого ячейки **B3 (=B2)** в ячейку **B6** дает содержимое ячейки **B(2+3) (=B5)**, а именно значения **11** (рис. 254).

Копирование содержимого ячейки **B3 (=B2)** в ячейку **E3** дает содержимое ячейки **B3 (=B2)**, то есть **B2**, а именно значения **14** (потому что по горизонтали сдвигаться не можем, рис. 254).

Копирование содержимого ячейки **B3 (=B2)** в ячейку **E6** дает содержимое ячейки **E2 (= (B+3 буква)2)**, то есть **E2**, а именно значения **10** (потому что по горизонтали сдвигаться не можем, а по вертикали можем, рис. 254).

Задание 1 Excel.

Построить таблицу чисел и таблицу квадратов чисел от **0** до **99** и оформить, как на рисунках 255 и 256).

Заполнить таблицу нужно, записав адрес, представляющий *формулу вычисления квадратов*, но с помощью адресов, ссылающихся на значения десятков и единиц, в единственную ячейку **C4** (использовать нужную адресацию: относительную, абсолютную или смешанную).

Затем скопировать адрес из ячейки **C4**, протянув за маркер-квадратик в правом нижнем углу до ячейки **L4**. Далее, выделив ячейки от **C4** до **L4** и ухватив за маркер-квадратик (для выделенной строки), протяните эту частичную строку вплоть до строки с индексом 13. При правильной формуле для адреса должна получиться вся таблица квадратов.

Сначала сделать таблицу чисел, затем сделать таблицу квадратов.

	A	B	C	D	E	F	G	H	I	J	K	L
1			Таблица чисел									
2			единицы									
3			0	1	2	3	4	5	6	7	8	9
4		0	0	1	2	3	4	5	6	7	8	9
5		1	10	11	12	13	14	15	16	17	18	19
6		2	20	21	22	23	24	25	26	27	28	29
7		3	30	31	32	33	34	35	36	37	38	39
8		4	40	41	42	43	44	45	46	47	48	49
9		5	50	51	52	53	54	55	56	57	58	59
10		6	60	61	62	63	64	65	66	67	68	69
11		7	70	71	72	73	74	75	76	77	78	79
12		8	80	81	82	83	84	85	86	87	88	89
13		9	90	91	92	93	94	95	96	97	98	99
14												

Рис. 255. Таблица чисел

	A	B	C	D	E	F	G	H	I	J	K	L
1			Таблица квадратов									
2			единицы									
3			0	1	2	3	4	5	6	7	8	9
4		0	0	1	4	9	16	25	36	49	64	81
5		1	100	121	144	169	196	225	256	289	324	361
6		2	400	441	484	529	576	625	676	729	784	841
7		3	900	961	1024	1089	1156	1225	1296	1369	1444	1521
8		4	1600	1681	1764	1849	1936	2025	2116	2209	2304	2401
9		5	2500	2601	2704	2809	2916	3025	3136	3249	3364	3481
10		6	3600	3721	3844	3969	4096	4225	4356	4489	4624	4761
11		7	4900	5041	5184	5329	5476	5625	5776	5929	6084	6241
12		8	6400	6561	6724	6889	7056	7225	7396	7569	7744	7921
13		9	8100	8281	8464	8649	8836	9025	9216	9409	9604	9801

Рис. 256. Таблица квадратов чисел

Указание ширина столбцов **C-L** равна 5 пунктам.

Строки с текстом «Таблица чисел» или «Таблица квадратов», «единицы», «десятки» должны быть в объединенных ячейках и расположены по центру.

Практическое занятие № 14. Поиск решения

Задачи линейного программирования и функция «Поиск решения»

Несмотря на свою эффективность, функция «Поиск решения» не находится в первых рядах панели инструментов или контекстного меню. Многие пользователи, работающие в Excel годами, даже не подозревают о ее существовании. Дело в том, что *по умолчанию* она вообще отключена и для ее добавления на ленту нужно проделать следующие шаги:

1. Откройте меню **Файл**, кликнув по соответствующему пункту **Файл** (рис. 257).

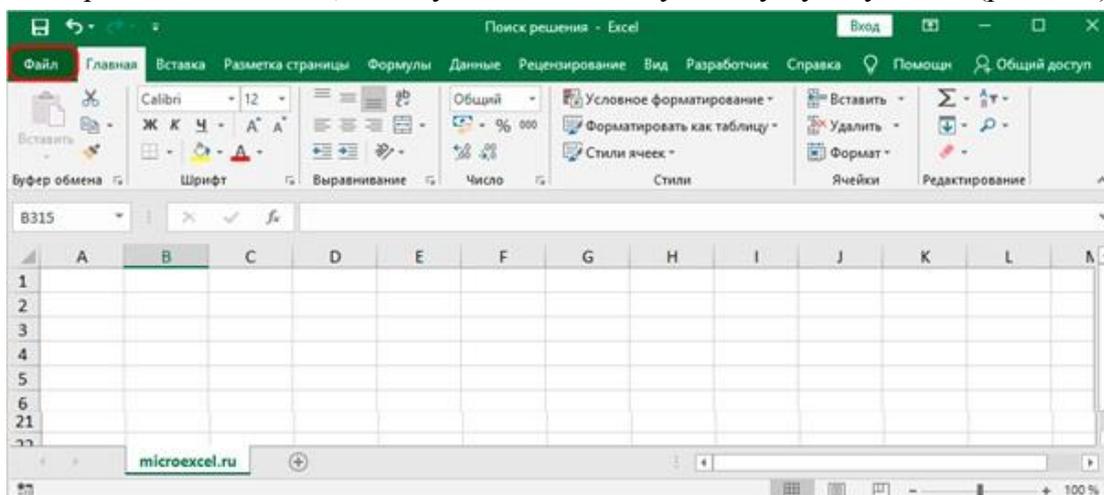


Рис. 257. Для активирования приложения **Поиск решения** выбрать пункт **Файл**

2. Кликните по разделу **Параметры**, который находится внизу вертикального перечня пунктов меню, расположенного с левой стороны экрана (рис. 258).

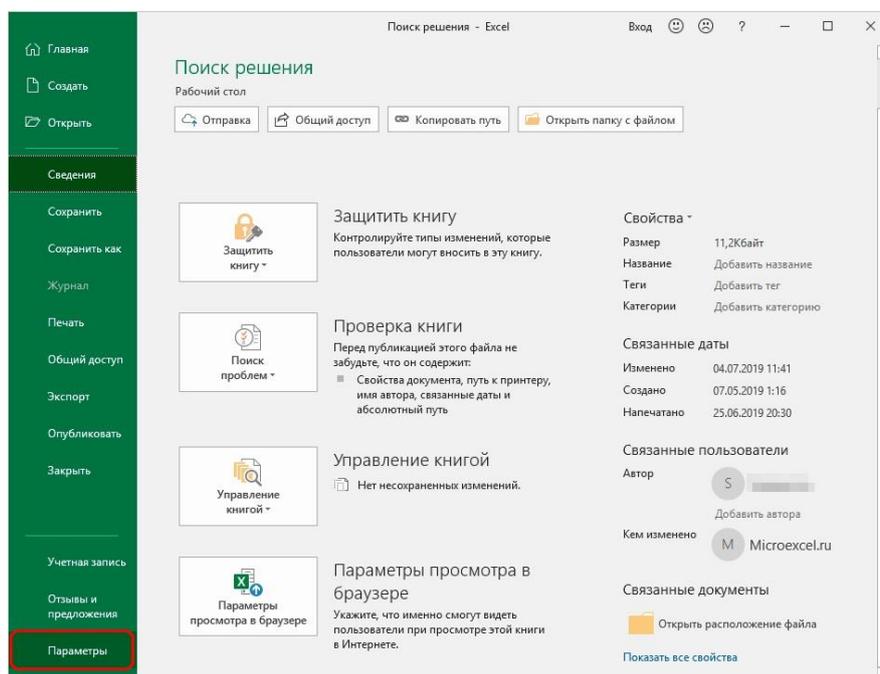


Рис. 258. Расположение пункта **Параметры** в окне **Поиск решения**

3. Перейдете на следующую вкладку, в которой в меню слева расположен пункт **Надстройки** (рис. 259). Щелкните по подразделу **Надстройки**. В этом подразделе отображаются все надстройки программы, а внизу находится пункт **Управление**. Справа от него представлено выпадающее меню, в котором должны быть выбраны **Надстройки Excel**, обычно уже установленные по умолчанию. Нажмите кнопку **Перейти**.

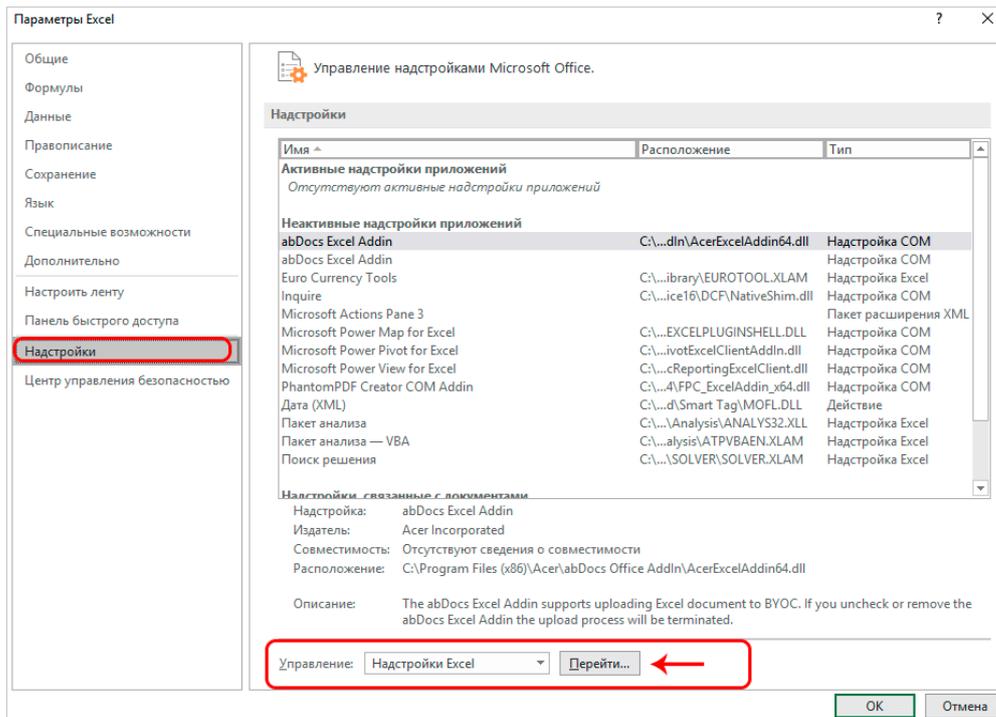


Рис. 259. Расположение вкладки **Надстройка**; в окне параметров **Надстройки** пункта **Управление** и кнопки **Перейти**

4. Появится новое вспомогательное окно **Надстройки** (рис. 260). Установите флажок напротив опции **Поиск решения** и нажмите **ОК**.

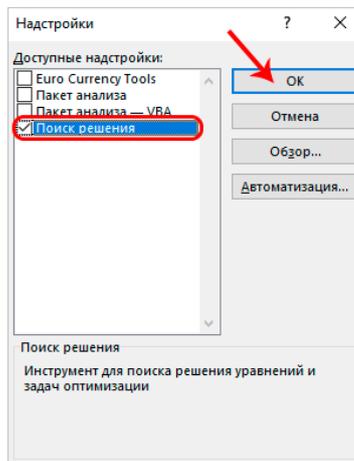


Рис. 260. Окно **Надстройки**, меню **Доступные надстройки**, отмеченный пункт **Поиск решения**

5. Все готово. Требуемая функция появится на главной панели в правой части вкладки **Данные** (рис. 261).

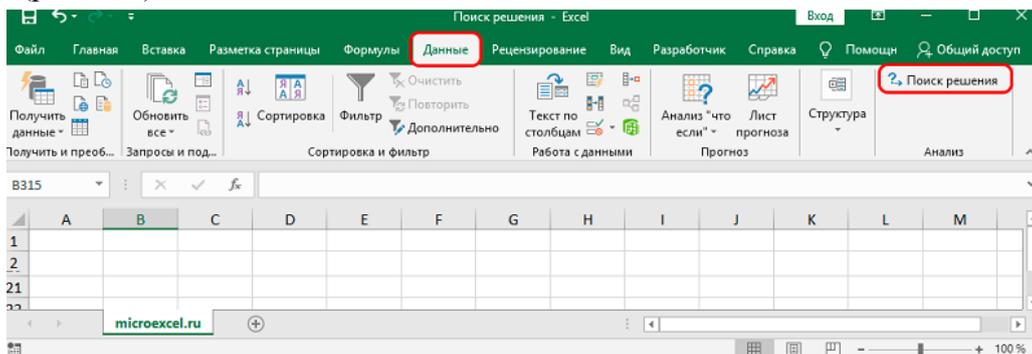


Рис. 261. Расположение вкладки **Данные** на главной панели и блока **Анализ** с функцией **Поиск решения**

Принципы решения задач оптимизации (управленческих и экономических задач)

Задачи оптимизации очень часто встречаются в управленческой, финансовой и научной деятельности. Они позволяют отыскать наилучшее (оптимальное) решение (например, дающее максимальную прибыль или обеспечивающее минимальные затраты). При этом требуется учитывать ряд дополнительных условий на значения используемых параметров. Для решения подобных задач используются, как правило, методы математического программирования с использованием приложения Excel в режиме **Поиск решения**.

Режим Поиск решения позволяет:

- использовать одновременно до 200 изменяемых параметров;
- задавать ограничения для этих параметров;
- используя метод последовательных приближений (то есть итерационные вычисления) отыскивать оптимальное решение.

Задачи, для решения которых используют режим **Поиск решения**, должны обладать рядом свойств:

- иметь единственную ячейку (целевую), содержащую формулу (целевую функцию), значение которой должно быть получено как максимальное/минимальное/равное конкретному значению (максимальная прибыль, минимальный фонд зарплаты, ограничение, равное имеющейся наличной сумме);
- формула в этой ячейке должна содержать адреса ячеек (ссылок), в которых будут находиться неизвестные или переменные решаемой задачи ((изменяемые ячейки) Поиск решения задачи заключается в том, чтобы подобрать такие значения этих переменных, которые бы давали оптимальное значение для формулы в целевой ячейке. Изменяемые ячейки могут содержать, например, себестоимость или цену товара, транспортные товары или налоговые ставки.
- кроме того, может быть задано некоторое количество *ограничений* — условий или соотношений, которым должны удовлетворять некоторые параметры из изменяемых ячеек. Например, можно потребовать, чтобы общие затраты не превосходили 100000 рублей или чтобы затраты на рекламную кампанию составили от 10 до 15% от общих расходов.

Типичные задачи, решаемые с помощью Поиска решений: транспортная задача, задача о назначениях, задача о расписании сотрудников, задача о распределении ресурсов, задача о выборе портфеля ценных бумаг/

Транспортная задача.

Имеется несколько пунктов производства и пунктов потребления некоторого продукта. Для каждого из пунктов производства задан объем производства, а для каждого из пунктов потребления — объем потребления. Известна также стоимость перевозки из каждого пункта производства в пункт потребления. Все пункты потребления должны быть обеспечены необходимой продукцией, но из каждого пункта производства не может вывозиться продукции больше, чем там производится, а стоимость перевозки должна быть минимальной.

Задача о выборе оптимального меню.

Имеются несколько должностей и соответствующее количество претендентов на эти должности. Назначение претендентов на должности связано с некоторыми затратами. Требуется так распределить претендентов по должностям, чтобы суммарные затраты были минимальны.

Задача о выборе портфеля ценных бумаг.

Вкладчик хочет выбрать портфель ценных бумаг, при этом известны средние значения доходов от каждого вида ценных бумаг, и ожидаемый порядок этих доходов. Требуется отыскать оптимальный портфель, обеспечивающий максимальный ожидаемый доход при минимальном рассеянии, и, следовательно, минимальном риске.

Рассмотрим решения этих задач.

I. Транспортная задача.

Фирме необходимо реализовать перевозку продукции с трех складов в пять магазинов. Сведения о наличии продукции на складах, о потребности в этой продукции у магазинов и о стоимости перевозки единицы продукции с каждого склада во все магазины.

Решение задачи включает три этапа:

1. Построение математической модели
2. Построение начального плана решения
3. Оптимизация начального плана решения

Склады		Магазины				
		M1	M2	M3	M4	M5
Номер склада	Запас	Стоимость перевозок				
S1	15	1	0	3	4	2
S2	25	5	1	2	3	3
S3	20	4	8	1	4	3
		Потребности магазинов				
		20	12	5	8	15

Рис. 262. Начальный план решения

1. Построение математической модели

Обозначим

X_{ij} — количество продукции, отправляемой со склада i в магазин j ;

C_{ij} — стоимость перевозки единицы продукции со склада i в магазин j .

Математическая модель будет состоять из ряда ограничений:

- а) исходя из физического смысла задачи (количество и стоимость не могут быть отрицательными величинами): $X_{ij} \geq 0 \quad C_{ij} \geq 0$ (1)

- б) ограничения по предложению (со складов нельзя вывезти продукции больше, чем там имеется):

$$\begin{aligned} X_{11} + X_{12} + X_{13} + X_{14} + X_{15} &\leq 15 \\ X_{21} + X_{22} + X_{23} + X_{24} + X_{25} &\leq 25 \\ X_{31} + X_{32} + X_{33} + X_{34} + X_{35} &\leq 20 \end{aligned} \quad (2)$$

- в) ограничения по спросу (в магазины следует завезти не меньше продукции, чем им требуется):

$$\begin{aligned} X_{11} + X_{21} + X_{31} &\geq 20 \\ X_{12} + X_{22} + X_{32} &\geq 12 \\ X_{13} + X_{23} + X_{33} &\geq 5 \\ X_{14} + X_{24} + X_{34} &\geq 8 \\ X_{15} + X_{25} + X_{35} &\geq 15 \end{aligned} \quad (3)$$

Общая стоимость перевозок (целевая функция) равна:

$$\begin{aligned} Z = \sum_{i=1}^3 \sum_{j=1}^5 C_{ij} \cdot X_{ij} = & 1 \cdot X_{11} + 0 \cdot X_{12} + 3 \cdot X_{13} + 4 \cdot X_{14} + 2 \cdot X_{15} + \\ & + 5 \cdot X_{21} + 1 \cdot X_{22} + 2 \cdot X_{23} + 4 \cdot X_{24} + 3 \cdot X_{25} + \\ & + 4 \cdot X_{31} + 8 \cdot X_{32} + 1 \cdot X_{33} + 4 \cdot X_{34} + 3 \cdot X_{35} \end{aligned} \quad (4)$$

Необходимо определить такие значения X_{ij} , которые удовлетворяют ограничениям (1), (2), (3) и обращают в минимум целевую функцию Z (4).

В такой постановке задачи является транспортной задачей линейного программирования.

Необходимым и достаточным условием разрешимости транспортной задачи является условие баланса:

$$Z = \sum_{i=1}^3 S_i = \sum_{j=1}^5 m_j,$$

где $\sum_{i=1}^3 S_i$ — суммарное количество продукции на складах,

(при этом $S_i = \sum_{j=1}^5 X_{ij}$ — количество продукции на складе, $i=1, 2, 3$);

$\sum_{j=1}^5 m_j$ — суммарное количество продукции, требуемое в магазинах (при этом $m_j = \sum_{i=1}^3 X_{ij}$ — количество продукции, которое требуется j -му магазину, $j=1, 2, 3, 4, 5$)

$$\text{В нашем случае } \sum_{i=1}^3 Si = 15+25+20=60 \quad \sum_{j=1}^5 mj = 20+12+5+8+15=60$$

$$\sum_{i=1}^3 Si = \sum_{j=1}^5 mj = 60, \text{ следовательно, задача с балансом.}$$

2. Разработка электронной таблицы с начальным планом решения

	A	B	C	D	E	F	G
1	ОПТИМИЗАЦИЯ ПЛАНА ПЕРЕВОЗОК						
2	Склады		Магазины				
3			1-й	2-й	3-й	4-й	5-й
4	Номер	Запас	Стоимость перевозок				
5	1	15	1	0	3	4	2
6	2	25	5	1	2	3	3
7	3	20	4	8	1	4	3
8	Потребности магазинов						
9			20	12	5	8	15
10	Всего ввозится		План перевозок				
11	1	5	1	1	1	1	1
12	2	5	1	1	1	1	1
13	3	5	1	1	1	1	1
14	Завоз в магазины						
15			3	3	3	3	3
16	Стоимость перевозок в каждый магазин		10	9	6	11	8
17	Целевая функция		44				

	A	B	C	D	E	F	G
1	ОПТИМИЗАЦИЯ ПЛАНА ПЕРЕВОЗОК (начальный план)						
2	Склады		Магазины				
3			1-й	2-й	3-й	4-й	5-й
4	Номер	Запас	Стоимость перевозки одной единицы продукции				
5	S1	15	1		0	3	4
6	S2	25	5		1	2	3
7	S3	20	4		8	1	4
8	Потребности магазинов						
9			20	12	5	8	15
10	Всего вывозится		План перевозок				
11	S1	=СУММ(C11:G11)	1		1	1	1
12	S2	=СУММ(C12:G12)	1		1	1	1
13	S3	=СУММ(C13:G13)	1		1	1	1
14	Завоз продукции в магазины						
15			=СУММ(C11:C13)	=СУММ(D11:D13)	=СУММ(E11:E13)	=СУММ(F11:F13)	=СУММ(G11:G13)
16	Стоимость перевозок в каждый магазин		=СУММПРОИЗВ(C5:C7;C11:C13)	=СУММПРОИЗВ(D5:D7;D11:D13)	=СУММПРОИЗВ(E5:E7;E11:E13)	=СУММПРОИЗВ(F5:F7;F11:F13)	=СУММПРОИЗВ(G5:G7;G11:G13)
17	Целевая функция		=СУММ(C16:G16)				

Рис. 263. Начальный план решения в режиме вычислений

Рис. 264. Начальный план решения в режиме показа формул

1. Подготовка блока ячеек с исходными данными

В ячейках **B4:B7** поместите сведения о наличии продукции на складах. В ячейках **C9:G9** — сведения о потребностях магазинов. В ячейках **C5:G7** — данные о стоимости перевозки единицы продукции со складов в магазин.

2. Построение начального плана перевозок

Считаем, что с каждого склада в каждый магазин везут одну единицу товара (ячейки **C11:G13** заполним единицами)

3. Вычисление количества перевозимой продукции

а) В ячейку **B11** введите формулу вычисления количества продукции, вывозимой с 1-го склада =СУММ(C11:G11) (для OpenOffice.org Calc =SUM(C11:G11))

Аналогично, в ячейки **B12**, **B13** введите формулу вычисления количества продукции, вывозимой со второго и третьего складов

$$=СУММ(C12:G12) \text{ (Calc=SUM(C12:G12))}$$

$$=СУММ(C13:G13) \text{ (Calc=SUM(C13:G13))}$$

Для начала перевозок все суммы равны 5.

б) В ячейку **C15** вставьте формулу вычисления количества продукции, которую везут в 1-й магазин =СУММ(C11:C13) (Calc=SUM(C11:C13))

Аналогично, в ячейки **D15:G15** введите формулу вычисления количества продукции, которую везут во 2-й, 3-й, 4-й и 5-й магазины.

$$\text{В ячейку } \mathbf{D15} \quad =СУММ(D11:D13) \text{ (Calc=SUM(D11:D13))}$$

$$\text{В ячейку } \mathbf{E15} \quad =СУММ(E11:E13) \text{ (Calc=SUM(E11:E13))}$$

$$\text{В ячейку } \mathbf{F15} \quad =СУММ(F11:F13) \text{ (Calc=SUM(F11:F13))}$$

$$\text{В ячейку } \mathbf{G15} \quad =СУММ(G11:G13) \text{ (Calc=SUM(G11:G13))}$$

4. Определение стоимости перевозок в каждый магазин

Для определения стоимости перевозок в 1-й магазин $Z_1 = X_{11}C_{11} + X_{12}C_{12} + X_{13}C_{13}$ введите в ячейку **C16** формулу =СУММПРОИЗВ(C5:C7;C11:C13)

$$\text{(В OpenOffice.org Calc=SUMPRODUCT(C5:C7;C11:C13))}$$

В ячейке **D16** вычислите стоимость перевозок во 2-й магазин $Z_2 = X_{21}C_{11} + X_{22}C_{12} + X_{23}C_{13}$. Для этого введите в ячейку **D16** формулу =СУММПРОИЗВ(D5:D7;D11:D13)

$$\text{(В OpenOffice.org Calc=SUMPRODUCT(D5:D7;D11:D13))}$$

Аналогично, в ячейки **E16:G16** введите формулы вычисления количества продукции в остальные магазины

В ячейку **E16** =СУММПРОИЗВ(E5:E7;E11:E13)

(Calc=SUMPRODUCT(E5:E7;E11:E13))

В ячейку **F16** =СУММПРОИЗВ(F5:F7;F11:F13)

(Calc=SUMPRODUCT(F5:F7;F11:F13))

В ячейку **G16** =СУММПРОИЗВ(G5:G7;G11:G13)

(Calc=SUMPRODUCT(G5:G7;G11:G13))

5. Определение общей стоимости перевозок (целевая функция ЦФ)

$$Z = Z_1 + Z_2 + Z_3 + Z_4 + Z_5$$

Для этого введите в ячейку **B17** формулу =СУММ(C16:G16) (Calc=SUM(C16:G16))

Для нашего начального плана (рис. 263, 264) полученная стоимость равна 44 денежным единицам (в **B17**).

	A	B	C	D	E	F	G
1	ОПТИМИЗАЦИЯ ПЛАНА ПЕРЕВОЗОК (начальный план)						
2	Склады		Магазины				
3			M1	M2	M3	M4	M5
4	Номер	Запас	Стоимость перевозки одной единицы продукции				
5	S1	15	1	0	3	4	2
6	S2	25	5	1	2	3	3
7	S3	20	4	8	1	4	3
8			Потребности магазинов				
9			20	12	5	8	15
10	Всего вывозится		План перевозок				
11	S1	=SUM(C11:G11)	1	1	1	1	1
12	S2	=SUM(C12:G12)	1	1	1	1	1
13	S3	=SUM(C13:G13)	1	1	1	1	1
14			Завоз продукции в магазины				
15			=SUM(C11:C13)	=SUM(D11:D13)	=SUM(E11:E13)	=SUM(F11:F13)	=SUM(G11:G13)
16	Стоимость перевозок в каждый магазин		=SUMPRODUCT(C5:C7;C11:C13)	=SUMPRODUCT(D5:D7;D11:D13)	=SUMPRODUCT(E5:E7;E11:E13)	=SUMPRODUCT(F5:F7;F11:F13)	=SUMPRODUCT(G5:G7;G11:G13)
17	Целевая функция		=SUM(C16:G16)				

Рис. 265. Начальный план решения в режиме показа формул в OpenOffice.org

6. Улучшение (оптимизация) плана перевозок

I. Использование режима Поиск решения в Excel MS Office

1) Выполните команды **Сервис->Поиск решения**. Откроется диалоговое окно **Поиск решения** (рис. 266). (При работе с MS office 2007 команды **Данные->Анализ->Поиск решения**)

2) Ввод данных

Установите целевую ячейку **B17**

Равной: **минимальному значению**

Изменяя ячейки: **C11:G13**

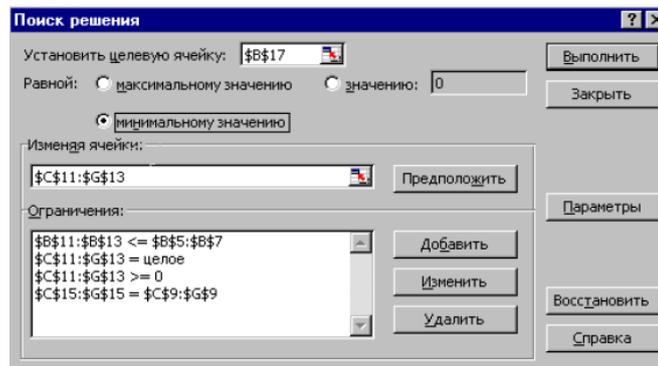


Рис. 266. Диалоговое окно Поиск решения

3) Щелкните по кнопке **Добавить** для ввода ограничений (рис. 266).

4) В открывшемся окне **Изменение ограничения** введите ограничения (рис. 267)

$$C11:G13 \geq 0$$

$$C11:G13 = \text{целое}$$

$$B11:B13 \leq B5:B7$$

$$C15:G15 = C9:G9$$

(левая часть каждого ограничения вводится в поле **Ссылка на ячейку**, правая часть — в поле **Ограничения**, знак выбирается в средней части окна. После ввода каждого ограничения нужно нажать кнопку **Добавить**, после ввода последнего ограничения нажмите кнопку **ОК**).

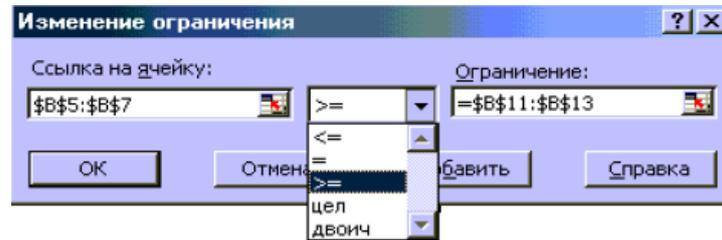


Рис. 267. Диалоговое окно **Изменение ограничений** для ввода ограничений

5) Для запуска режима **Поиск решения** щелкните по кнопке **Выполнить** (на рисунке 266 **Поиск решения**). Если все формулы таблицы начального плана решения и все ограничения в диалоговых окнах введены правильно, появится окно с сообщением о том, что решение найдено. Щелкнув по кнопке **ОК**, получите решение (рис. 268).

	A	B	C	D	E	F	G
1	ОПТИМИЗАЦИЯ ПЛАНА ПЕРЕВОЗОК						
2	Склады		Магазины				
3			1-й	2-й	3-й	4-й	5-й
4	Номер	Запас	Стоимость перевозки одной единицы продукции				
5	S1	15	1	0	3	4	2
6	S2	25	5	1	2	3	3
7	S3	20	4	8	1	4	3
8			Потребности магазинов				
9			20	12	5	8	15
10	Всего вывозится		План перевозок				
11	S1	15	15	0	0	0	0
12	S2	25	0	12	0	8	5
13	S3	20	5	0	5	0	10
14			Завоз продукции в магазины				
15			20	12	5	8	15
16	Стоимость перевозок в		35	12	5	24	45
17	Целевая функция		121				

Рис. 268. Решение, найденное в режиме **Поиск решения**

II. Использование режима **Поиск решения** в *OpenOffice.org Calc*

1) После выполнения команд **Сервис->Поиск решения** откроется диалоговое окно **Поиск решения** (рисунок 269 **Оптимальное решение**). (При работе с MS office 2007 команды **Данные->Анализ->Поиск решения**)

2) Ввод данных

Целевая функция **B17**

Цель: **минимальному значению**

Параметры функции: **C11:G13**

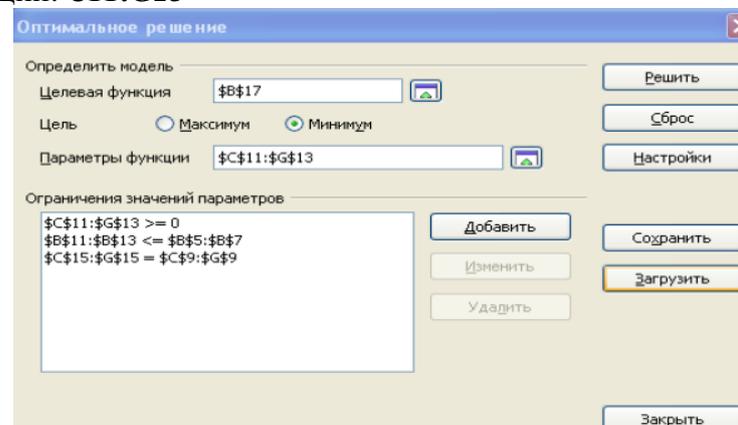


Рис. 269. Диалоговое окно **Поиск решения** в *OpenOffice.org Calc*

3). Щелкните по кнопке **Добавить** для ввода ограничений

4). В открывшемся окне **Ограничение** (рисунок 270) значений параметров ввести

$C11:G13 \geq 0$
 $B11:B13 \leq B5:B7$
 $C15:G15 = C9:G9$

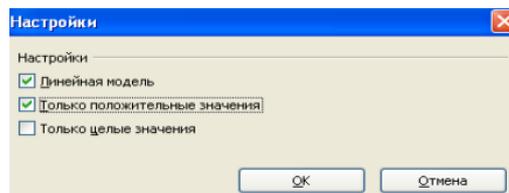
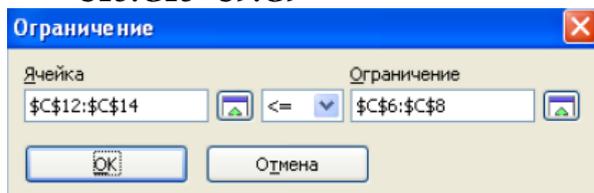


Рис. 270. Диалоговое окно **Ограничения** OpenOffice.org Calc Рис. 271. Диалоговое окно **Настройки** OpenOffice.org Calc

После ввода каждого ограничения нажмите кнопку **ОК**.

5) Щелкните по кнопке **Настройки** и перейдите в диалоговое окно **Настройки** (рис. 269), установите флажок (поставьте галочку) в пункте **Только положительные значения** (рис. 271), затем нажмите **ОК**.

В результате улучшения плана получите оптимальный план стоимости перевозок с целевой функцией, равной 121 денежной единицей

	A	B	C	D	E	F	G
1	ОПТИМИЗАЦИЯ ПЛАНА ПЕРЕВОЗОК						
2	Склады		Магазины				
3			1-й	2-й	3-й	4-й	5-й
4	Номер	Запас	Стоимость перевозки одной единицы продукции				
5	S1	15	1	0	3	4	2
6	S2	25	5	1	2	3	3
7	S3	20	4	8	1	4	3
8			Потребности магазинов				
9			20	12	5	8	15
10	Всего вывозится		План перевозок				
11	S1	15	15	0	0	0	0
12	S2	25	0	12	0	8	5
13	S3	20	5	0	5	0	10
14			Завоз продукции в магазины				
15			20	12	5	8	15
16	Стоимость перевозок в		35	12	5	24	45
17	Целевая функция		121				

Рис. 272. Оптимальный план стоимости перевозок

II. Решение задачи о штате фирмы

Администрации брокерской фирмы требуется определить штаты и составить график работы обслуживающего персонала. При этом необходимо обеспечить следующие условия:

- каждый из сотрудников должен иметь пять рабочих дней в неделю и два выходных дня подряд;
- все сотрудники фирмы имеют одинаковую заработную плату (500 руб. в день);
- всего в фирме на текущий момент работает 30 человек;
- исходя из специфики работы фирмы, имеются требования к минимальному количеству работающих сотрудников для каждого из дней недели.

День недели	Пн.	Вт.	Ср.	Чт.	Пт.	Сб.	Вс.
Требуемое число сотрудников	16	12	14	16	16	25	25

Требуется определить минимальное количество сотрудников в фирме для обеспечения названных условий. Решим эту задачу с помощью инструмента **Поиск решения**.

Решение задачи включает три этапа:

1. Построение математической модели
2. Построение начального плана решения
3. Оптимизация начального плана решения

1. Построение математической модели

Поскольку одним из требований является наличие пяти рабочих дней у каждого сотрудника и двух выходных рядом. Составим таблицу (рис. 273), в которую занесем возможный режим работы для каждого сотрудника. Внизу этой таблицы запишем для оценки необходимого числа работающих данные выше приведенную таблицу требований к минимальному количеству работающих сотрудников.

Выходные дни	Число имеющих этот график	Рабочие дни						
		Пн	Вт	Ср	Чт	Пт	Сб	Вс
Понедельник, вторник	$X_1 = 6$	$a_{11}=0$	$a_{12}=0$	$a_{13}=1$	$a_{14}=1$	$a_{15}=1$	$a_{16}=1$	$a_{17}=1$
Вторник, среда	$X_2 = 6$	$a_{21}=1$	$a_{22}=0$	$a_{23}=0$	$a_{24}=1$	$a_{25}=1$	$a_{26}=1$	$a_{27}=1$
Среда, четверг	$X_3 = 6$	$a_{31}=1$	$a_{32}=1$	$a_{33}=0$	$a_{34}=0$	$a_{35}=1$	$a_{36}=1$	$a_{37}=1$
Четверг, пятница	$X_4 = 6$	$a_{41}=1$	$a_{42}=1$	$a_{43}=1$	$a_{44}=0$	$a_{45}=0$	$a_{46}=1$	$a_{47}=1$
Пятница, суббота	$X_5 = 2$	$a_{51}=1$	$a_{52}=1$	$a_{53}=1$	$a_{54}=1$	$a_{55}=0$	$a_{56}=0$	$a_{57}=1$
Суббота, воскресенье	$X_6 = 2$	$a_{61}=1$	$a_{62}=1$	$a_{63}=1$	$a_{64}=1$	$a_{65}=1$	$a_{66}=0$	$a_{67}=0$
Воскресенье, понедельник	$X_7 = 2$	$a_{71}=0$	$a_{72}=0$	$a_{73}=0$	$a_{74}=0$	$a_{75}=1$	$a_{76}=1$	$a_{77}=0$
ВСЕГО:	30	22	18	18	18	22	26	26
ТРЕБУЕТСЯ	30	16	12	14	16	16	25	25

Рис. 273. Таблица возможного графика работы сотрудников (начальный план)

Определение возможных режимов работы

В первом столбце выше приведенной таблицы поместим возможные режимы работы сотрудников. Так как сотрудник должен иметь два выходных дня рядом, получаем следующие возможные режимы выходных дней:

- Понедельник, вторник
- Вторник, среда
- Среда, четверг
- Четверг, пятница
- Пятница, суббота
- Суббота, воскресенье
- Воскресенье, понедельник

Определение возможного графика работы

Обозначим число сотрудников, имеющих выходные в понедельник и вторник, X_1 . Число сотрудников, имеющих выходные во вторник и среду, — X_2 и так далее в соответствии с таблицей выше (второй столбец).

Из таблицы выше (её нижней части «требуется») видно, что больше всего сотрудников должны работать в субботу и воскресенье. Следовательно, число выходных у работников в эти дни должно быть меньше всего.

В условии задачи сказано, что всего в фирме 30 работников. Попробуйте составить предварительный график работы сотрудников.

В напряженные рабочие дни (то есть в строках x_5 , x_6 , x_7 таблицы) спланируйте по два человека отдыхающих. В остальные дни — по 6 человек. Всего $6 \cdot 4 + 2 \cdot 3 = 30$ человек. Таким образом, $X_1 = X_2 = X_3 = X_4 = 6$; $X_5 = X_6 = X_7 = \frac{30 - 4 \cdot 6}{4} = 2$.

В дальнейшем, изменяя эти значения, будете искать оптимальное решение, обеспечивающем минимальный фонд заработной платы (то есть минимальное количество работающих) при соблюдении требований к рабочему графику.

Определение числа работников, выходящих на работу каждый день согласно данному графику

В столбцах «Рабочие дни» введите параметры a_{ij} — индикаторы выхода на работу сотрудников, имеющих режим выходных в строке i в рабочий день j .

Например, a_{11} — индикатор выхода сотрудника, имеющего выходной в первой строке, в Понедельник. Так как этот день у него выходной, $a_{11} = 0$. В среду у этого сотрудника рабочий день, поэтому $a_{13} = 1$.

В строке «Всего» подсчитывается количество сотрудников, выходящих на работу согласно составленному графику.

Во втором столбце (Число имеющих этот график)

$$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 = 6 + 6 + 6 + 6 + 2 + 2 + 2 = 30.$$

В столбце «Всего» для понедельника общее число работающих подсчитывается по формуле:

$X_1*a_{11}+X_2*a_{21}+X_3*a_{31}+X_4*a_{41}+X_5*a_{51}+X_6*a_{16}+X_7*a_{71}=6*0+6*1+6*1+6*1+2*1+2*1+2*0=22$,
то есть находится сумма произведений элементов столбца «X» на элементы столбца «ВТ».

В последней строке «Требуются» записано число работников, которые должны работать каждый день недели согласно заданию (в таблице, рис.).

Определение целевой функции задачи

Как видим, в строке «Всего» удовлетворяются требования строки «Требуются» с избытком, то есть, план явно, не оптимальный. Следует улучшить этот план, имея в виду нахождение минимума для целевой функции — фонда недельной зарплаты, которая считается следующим образом:

$$F=B*K,$$

где F — фонд заработной платы, B — зарплата работника за день (согласно заданию $B=500$ руб.),

K — число выходов сотрудника в течение недели (в нашем случае согласно таблице ниже, строка «Всего»)

$$K=22+18+18+18+22+26+26=150$$

Таким образом, $F=B*K=500*150=75000$ (рис. 273).

Разработка электронной таблицы

1) Составим электронную таблицу (показ вычислений, показ формул), реализующую планирование списочного состава и графика работы фирмы.

	A	B	C	D	E	F	G	H	I	J	K
1	График работы персонала фирмы										
2			Имеющих этот график								
3	Выходные дни				Пн	Вт	Ср	Чт	Пт	Сб	Вс
4	Понедельник, вторник		6		0	0	1	1	1	1	1
5	Вторник, среда		6		1	0	0	1	1	1	1
6	Среда, четверг		6		1	1	0	0	1	1	1
7	Четверг, пятница		6		1	1	1	0	0	1	1
8	Пятница, суббота		2		1	1	1	1	0	0	1
9	Суббота, воскресенье		2		1	1	1	1	1	0	0
10	Воскресенье, понедельник		2		0	1	1	1	1	1	0
11											
12											
13					По дням недели						
14		Всего	30		22	18	18	18	22	26	26
15		Требуются	30		18	12	14	16	16	25	25
16	Дневная зарплата сотрудников	500									
17	Число выходов за неделю	150									
18	Общая недельная зарплата	75000									

Рис. 274. Начальный план решения в режиме вычислений

	A	B	C	D	E	F	G	H	I	J	K
1	График работы персонала фирмы (Реализация в Excel)										
2			Имеющих этот график								
3	Выходные дни				Пн	Вт	Ср	Чт	Пт	Сб	Вс
4	Понедельник, вторник	0	0		0		1	1	1	1	1
5	Вторник, среда	0	1		0		0	1	1	1	1
6	Среда, четверг	0	1		1		0	0	1	1	1
7	Четверг, пятница	5	1		1		1	0	0	1	1
8	Пятница, суббота	5	1				1	1	0	0	1
9	Суббота, воскресенье	20	1				1	1	1	0	0
10	Воскресенье, понедельник	15	0				1	1	1	1	0
11											
12											
13					По дням недели						
14		Всего	=СУММ(C4:C10)		=СУММПРОИЗВ(\$C\$4:\$C\$10;E4:E10)		=СУММПРОИЗВ(\$C=C4=C5=C6=C7=C8=C9=C10)				
15		Требуются	50		30		45	40	35	10	10
16	Дневная зарплата	55									
17	Число выходов за неделю		=СУММ(E14:K14)								
18	Общая недельная зарплата		=B16*B17								

Рис. 275. Начальный план решения в режиме показа формул

	A	B	C	D	E	F	G	H	I	J	K	
1					График работы персонала фирмы (Реализация в Calc)							
2			Именован этот график									
3	Выходные дни				Пн	Вт	Ср	Чт	Пт	Сб	Вс	
4	Понедельник				0	0	1	1	1	1	1	
5	Вторник		0		1	0	0	1	1	1	1	
6	Среда		0		1	1	0	0	1	1	1	
7	Четверг											
8	Пятница		5		1	1	1	0	0	1	1	
9	Суббота		5		1	1	1	1	0	0	1	
10	Суббота воскресенье		20		1	1	1	1	1	0	0	
11	Воскресенье											
12	понедельник		15		0	1	1	1	1	1	0	
13												
14		Всего	=СУММ(C4:C10)		По дням недели							
15		Требуется	50		30	45	45	40	35	10	1	
16	Дневная зарплата сотрудников	55										
17	Число выходов за неделю	=СУММ(E14:K14)										
18	Общая недельная зарплата	=E16*E17										

Рис. 276. Начальный план решения в режиме показа формул в OpenOffice.org

2) Введите на рабочий лист необходимые исходные данные согласно таблице «Начальный план решения в режиме вычислений»: ячейки **A1:K10**.

3) Введите формулы, описывающие количество работающих сотрудников. Поместите в ячейку **C14** общее количество сотрудников, работающих по данному графику, а в интервал ячеек **E14:K14** — количество работающих сотрудников в соответствующий день недели. Для этого в ячейку **C14** нужно поместить формулу суммирования значений интервала **C4:C10**, **=СУММ(C4;C10)** (Calc **SUM(C4;C10)**), а в ячейки интервала **E14:K14** следует просуммировать произведения ячеек из столбца **C** на значения соответствующего столбца из интервала **E:K**, заполненного нулями и единицами.

Для этого:

- щелкните по кнопке **Мастер функции**, выберите функцию **СУММПРОИЗВ** (SUMPRODUCT). В первом окне наберите **\$C\$4:\$C\$10**, во втором — **E4:E10**.
- щелкните по кнопке **ОК**.

Пояснения. Поскольку собираемся распространить формулу из ячейки **E14** в другие ячейки интервала **F14:K14**, но при этом постоянно должны ссылаться на интервал **C4:C10**, чтобы эта ссылка была абсолютной (используем знак абсолютного адреса \$ для ячеек **C4:C10**, получаем **\$C\$4:\$C\$10**). Таблица (рис. 274).

Теперь установите указатель мыши на маркер выделения в ячейке **E14**, и дотяните его до ячейки **K14**, и нужные формулы будут распространены на весь выделенный интервал. Осталось ввести данные о необходимом количестве работников в каждый из дней недели, которые будут использованы в качестве данных для ограничений решаемой задачи, формулу для общей заработной платы и необходимые текстовые комментарии, а также выполнить, если это нужно, дополнительное форматирование для окончательного оформления рабочего листа.

4) Введите комментарии в ячейки **A16:A18**, требуемое число работников в ячейки **E15:K15**, дневную зарплату в **C16**.

В ячейку **C17** введите число выходов за неделю: **=СУММ(E14:K14)** (Calc: **=SUM(E14:K14)**).

В **C18** поместите формулу для вычисления недельной зарплаты работников **=C16*C17**.

Оптимизация решения

Исходные данные подготовлены. Сформулируем задачу для режима **Поиск решения**. В условиях нашей задачи целевой функцией будет функция в ячейке **C18** (общая недельная зарплата), изменяемыми ячейками — интервал **C4:C10**, содержащий число сотрудников, имеющих данный график работы и, наконец, интервал **E14:K15** будет использоваться при определении ограничений.

Запустите режим **Поиск решения**. Для этого введите команду: Сервис->**Поиск решения** (появится окно **Поиск решения**).

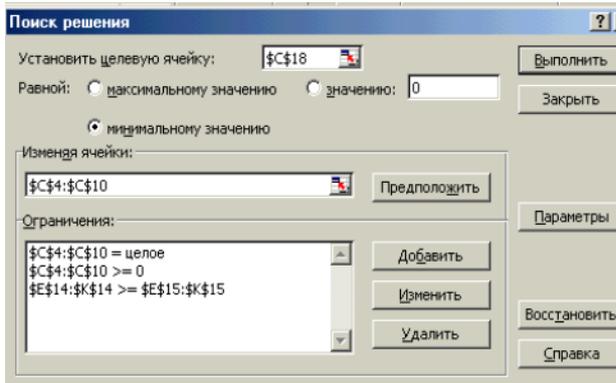


Рис. 277. Окно **Поиск решения** в MS Excel



Рис. 278. Окно **Поиск решения** в OpenOffice Calc

1.1. Определим параметры окна **Поиск решения** для MS Excel (для программы Calc установки проводятся аналогично).

- в поле **Установить целевую ячейку** (Целевая функция) введите **=C\$18**
- **равной** (Цель) установите минимальному значению (Минимум)
- в поле **Изменяя ячейки** (Параметры функции) введите **=C\$4:C\$10** (можно использовать выделение данного интервала мышью).

1.2. Ввод ограничений

а) щелкните по кнопке **Добавить** (**Ограничения значений параметра**), появится окно **Изменение ограничения**

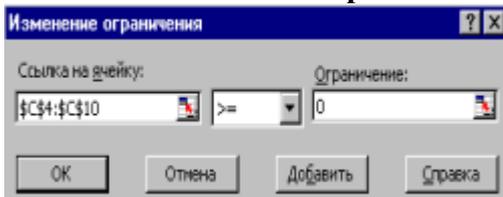


Рис. 279. Окно **Изменение ограничения** в MS Excel

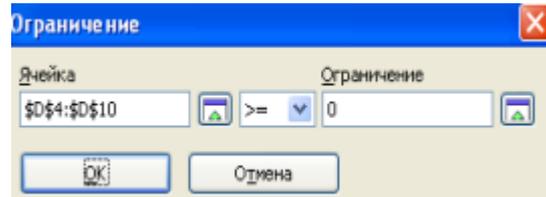


Рис. 280. Окно **Изменение ограничения** в OpenOffice Calc

б) в окне **Изменение ограничения**

- введите ссылку на ячейки: **C4:C10**,
- выберите пункт **Ограничение цели**,
- щелкните по кнопке **Добавить** (кнопка **Настройки**->установить флажок **Только положительные значения**), то есть получаем ограничение **C4:C10** целые (так как в ячейках **C4:C10** хранится информация о количестве работающих, эти величины не могут быть дробными)

с) установите ссылку на ячейки: **=E\$14:\$K\$14**;

- выберите ограничение **>=**;
- в правой части выберите **Ограничение: =E15;E15**;
- щелкните по кнопке **Добавить**, то есть получите ограничение **E14:K14 ≥ E15:K15**. (Число работающих по нашему графику не может быть меньше требуемого условием задачи)

д) установите ссылку на ячейки **C4:C10**

- выберите ограничение **>=**;

е) в правой части выберите **Ограничение: 0** (то есть ввели ограничение **C4:C10 ≥ 0** — число работающих неотрицательно)

- щелкните по кнопке **ОК**, так как ввод ограничений завершен. Происходит возврат к окну **Поиск решения**.

1.3. Запустите программу **Поиск решения** на выполнение:

- в окне **Поиск решения** щелкните по кнопке **Выполнить (Решить)**.

Дополнение **Поиск решения** приступает к итерациям. Затем, закончив вычисления, открывает диалоговое окно **Результаты поиска решения**, в котором выводится сообщение о том, найдено или нет решение поставленной задачи.

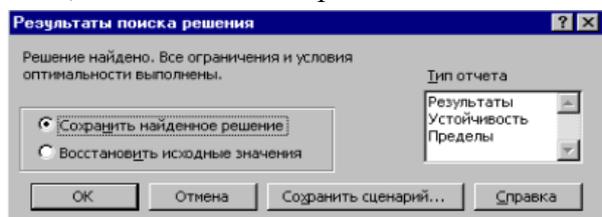


Рис. 281. Окно Результаты поиска решения MS Excel

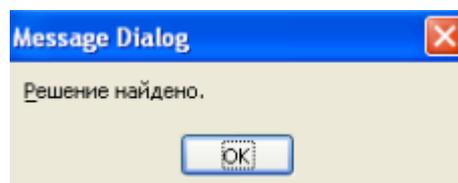


Рис. 282. Окно Результатов OpenOffice Calc

Если найденное решение Вас устраивает, можно сохранить его на рабочем столе:

- выберите опцию **Сохранить найденное решение (ОК)**
- щелкните по кнопке **ОК**.

В результате получаем таблицу (рис.), из которой видно, что теперь число работников 26 (вместо 30), следовательно, фонд зарплаты 65000 руб. (вместо 75000 руб.).

	A	B	C	D	E	F	G	H	I	J	K
1	График работы персонала фирмы										
2			график								
3	Выходные дни			Пн	Вт	Ср	Чт	Пт	Сб	Вс	
4	Понедельник, вторник	5	0	0	1	1	1	1	1	1	
5	Вторник, среда	9	1	0	0	1	1	1	1	1	
6	Среда, четверг	1	1	1	0	0	1	1	1	1	
7	Четверг, пятница	9	1	1	1	0	0	1	1	1	
8	Пятница, суббота	1	1	1	1	1	0	0	1	1	
9	Суббота, воскресенье	0	1	1	1	1	1	1	0	0	
10	Воскресенье, понедельник	1	0	1	1	1	1	1	1	0	
11											
12											
13				По дням недели							
14		Всего	26	20	12	16	16	16	25	25	
15		Требуется		16	12	14	16	16	25	25	
16											
17	Дневная зарплата сотрудников	500		Пн	Вт	Ср	Чт	Пт	Сб	Вс	
18	Число выходов за неделю	124									
19	Общая недельная зарплата	65000									

Рис. 283. Оптимальное решение задачи

III. Задача о производстве красок

Фабрика выпускает два вида красок для внутренних работ (**B**) и наружных работ (**H**).

Для производства красок используются два исходных продукта **P1** и **P2**. Максимально возможные суточные запасы этих продуктов, расходы на 1 т соответствующих красок и оптовые цены одной тонны приведены в таблице ниже.

Исходный продукт	Расход исходных продуктов (в тоннах) на тонну краски		Максимально возможный запас, в тоннах
	Краска B	Краска H	
P1	1	2	6
P2	2	1	8
Оптовая цена за одну тонну	2000 руб.	3000 руб.	

Изучение рынка сбыта показало, что суточный спрос на краску H никогда не превышает спроса на краску B более, чем на 1 т. Кроме того, установлено, что спрос на краску H никогда не превышает 2 т в сутки.

Какое количество краски каждого вида должна производить фабрика, чтобы доход от реализации продукции был максимален?

Решение включает три этапа, описанные в предыдущих двух задач.

Построение математической модели

1. Определение целевой функции

Обозначим X_B — суточный объём производства красок B ,
 X_H — суточный объём производства краски H .

Целевой функцией будет суммарная суточная прибыль от производства красок:

$$Z=2000*X_B+300*X_H$$

2. Определение ограничений на переменные

На переменные X_B и X_H накладывается три вида ограничений.

а) Ограничение по физическому смыслу задачи — объём производства красок не может быть отрицательным. Следовательно,

$$\begin{cases} X_B \geq 0 \\ X_H \geq 0 \end{cases}$$

б) Ограничение по ресурсам. Расход исходного продукта для производства обоих видов красок не может превосходить максимально возможный запас данного исходного продукта.

Таким образом, получим получаем систему из двух неравенств для P_1 :

$$\begin{cases} X_B + 2 * X_H \leq 6 \\ 2 * X_B + 2 * X_H \leq 8 \end{cases}$$

в) Ограничения на величину спроса краски имеют вид

$$\begin{cases} X_H - X_B \leq 1 \\ X_H \leq 2 \end{cases}$$

Итак, целью фабрики является определение среди множества допустимых значений X_H и X_B таких, которые обеспечат максимальное значение целевой функции Z при сделанных ограничениях (три системы выше).

3. Построение начального плана решения

Создайте электронную таблицу реализации решения в режиме показа формул (рис. 284) (в режиме показа вычислений таблица выглядит как на рис. 285).

	А	В
1	Оптимизация производства красок	
2	Переменные	
3	X1	X2
4	0	0
5	Целевая функция	
6	=3000*A4+2000*B4	
7	Ограничения	
8	Левая часть	Правая часть
9	=A4+2*B4	6
10	=2*A4+B4	8
11	=A4-B4	1
12	=A4	2

	А	В
1	Оптимизация производства красок	
2	Переменные	
3	X1	X2
4	0	0
5	Целевая функция	
6	0	
7	Ограничения	
8	Левая часть	Правая часть
9	0	6
10	0	8
11	0	1
12	0	2

Рис. 284. Реализации решения в режиме показа формул Рис. 285. Реализации решения в режиме показа вычислений
 Строка 5 предназначена для текста **Целевая функция**, а строка 6 — для формулы вычисления целевой функции.

В ячейках **A9:A12** вычисляются левые части ограничений для текущего объёма производства красок, а в ячейках **B9:B12** находятся правые части этих ограничений.

4. Оптимизация плана решения

1) Запустите средство **Поиск решения**, выполняя **Сервис->Поиск решения**

2) Установите в окне **Поиск решения** параметры, показанные на рисунках 286-287

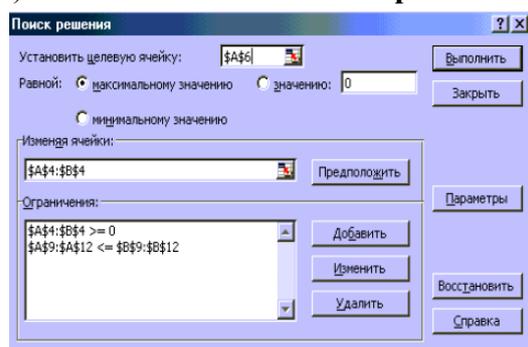


Рис. 286. Установка ограничений в Excel

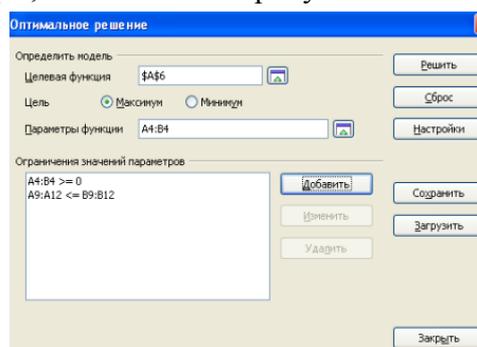


Рис. 287. Установка ограничений Calc

3) Щелкните по кнопке **Выполнить** (рис. 286). Получите результат, представленный на рисунке 288.

	А	В
1	Оптимизация производства красок	
2	<i>Переменные</i>	
3	X1	X2
4	3,33333333	1,33333333
5	<i>Целевая функция</i>	
6	12666,66667	
7	<i>Ограничения</i>	
8	Левая часть	Правая часть
9	6	6
10	8	8
11	-2	1
12	1,33333333	2

Рис. 288. Результат оптимизации плана решения производства красок

IV. Расчет общей суммы скидок различных товаров.

Задача такова: есть набор предметов. Требуется назначить каждому товару скидку таким образом, чтобы сумма по всем скидкам составила 4,5 млн. рублей. Эта сумма должна отобразиться в отдельной ячейке, которая называется целевой. Ориентируясь на нее, рассчитать остальные значения.

1. *Представление данных по продаже товаров* (представлены в табличном виде, рис. 289).

Наименование	Продано, шт.	Цена, руб.	Итого
Кроссовки беговые, размер 35	221	3 190	704 990
Кроссовки беговые, размер 39	400	6 990	2 796 000
Кроссовки беговые, размер 41	664	6 990	4 641 360
Кроссовки беговые, размер 42	334	6 990	2 334 660
Кроссовки беговые, размер 44	222	6 990	1 551 780
Кроссовки беговые, размер 45	221	6 990	1 544 790
Кроссовки для баскетбола, размер 39	98	5 990	587 020
Кроссовки для баскетбола, размер 43	334	5 890	1 967 260
Кроссовки теннисные, размер 38	443	7 990	3 539 570
Кроссовки теннисные, размер 41	553	7 990	4 418 470
Кроссовки теннисные, размер 42	123	7 990	982 770

Рис. 289. Представление данных по продаже товаров

Наименование	Продано, шт.	Цена, руб.	Итого	Сумма скидок	% скидки
Кроссовки беговые, размер 35	221	3 190	704 990		
Кроссовки беговые, размер 39	400	6 990	2 796 000		
Кроссовки беговые, размер 41	664	6 990	4 641 360		
Кроссовки беговые, размер 42	334	6 990	2 334 660		
Кроссовки беговые, размер 44	222	6 990	1 551 780		
Кроссовки беговые, размер 45	221	6 990	1 544 790		
Кроссовки для баскетбола, размер 39	98	5 990	587 020		
Кроссовки для баскетбола, размер 43	334	5 890	1 967 260		
Кроссовки теннисные, размер 38	443	7 990	3 539 570		
Кроссовки теннисные, размер 41	553	7 990	4 418 470		
Кроссовки теннисные, размер 42	123	7 990	982 770		
Итого:	3 613		25 068 670		

Рис. 290. Формирование столбца скидок и целевой ячейки

Задача сводится к вычислению скидки, на которую будут умножены все суммы по продажам всех наименований. Она будет найдена с помощью функции **Поиск решения**, а ячейка с этой скидкой будет называться искомой целевой функцией.

Данные ячейки (искомая и целевая) связываем вместе формулой, которую пишем в целевой ячейке следующим образом: **=D13*\$G\$2**, где ячейка **D13** содержит итоговую сумму по продажам всех товаров, а ячейка **\$G\$2** – абсолютные (неизменные) координаты искомой ячейки (рис. 291).

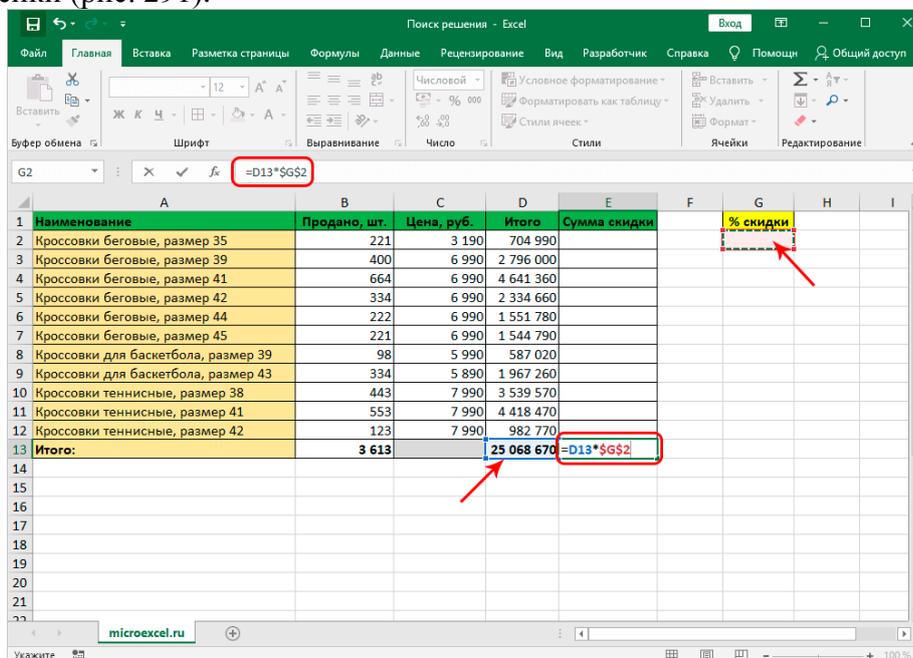


Рис. 291. Формирование формулы для целевой функции

2. Применение функции и ее настройка

Формула готова. Теперь нужно применить саму функцию (рис. 292).

1. Переключитесь во вкладку **Данные** и нажмите кнопку **Поиск решения**.

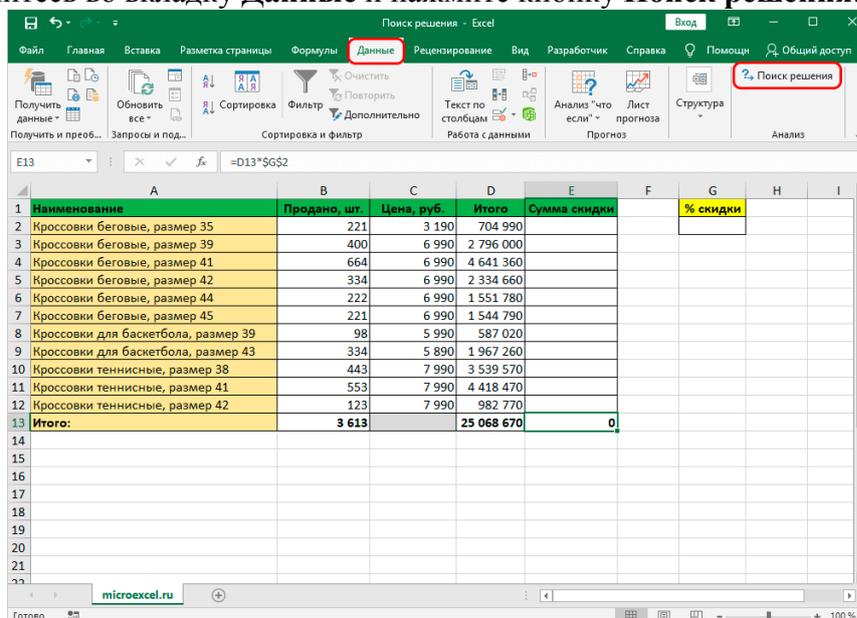


Рис. 292. Расположение вкладки **Данные** и кнопки **Поиск решения** в блоке **Анализ**

2. Откроется окно **Параметры поиска решения**, где задайте нужные настройки. В поле **Оптимизировать целевую функцию** (рис. 293): укажите адрес целевой ячейки, где планируется вывести сумму по всем скидкам. Можно прописать координаты вручную либо выбрать из таблицы, для чего сначала кликнуть по области ввода, затем – по нужной ячейке.

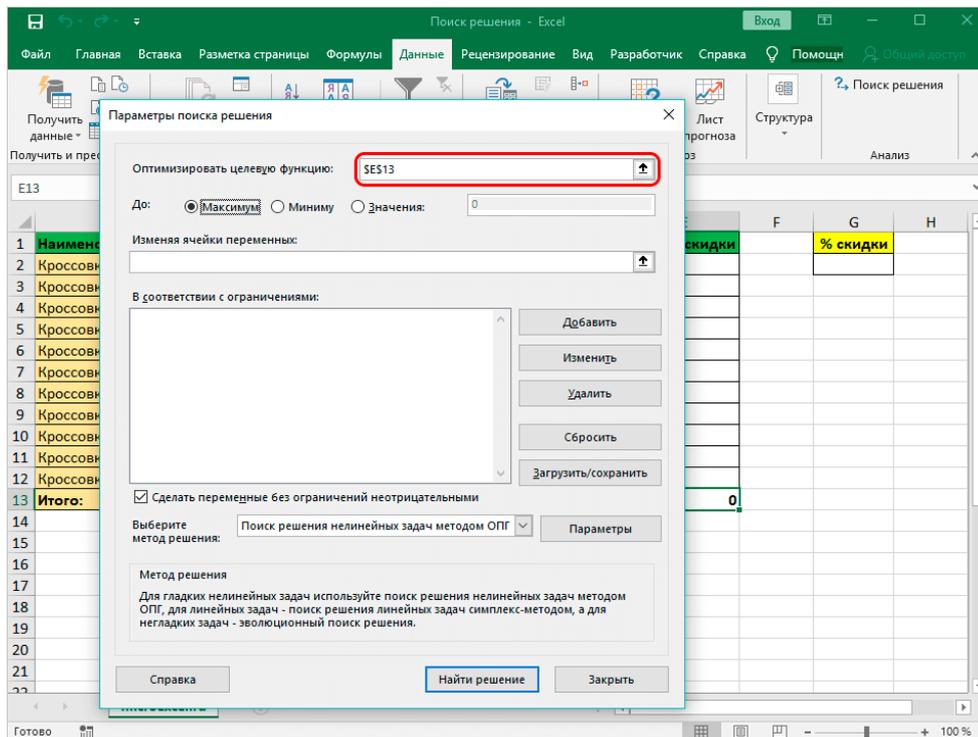


Рис. 293. Заполнение информации о характере целевой функции

3. Настройка других параметров (рис. 294). В пункте **До**: задайте максимальную границу, минимальную границу или же точное число. Исходя из поставленной задачи поставьте отметку рядом с опцией **Значение** и введите число **4500000** – сумму скидков по всем наименованиям.

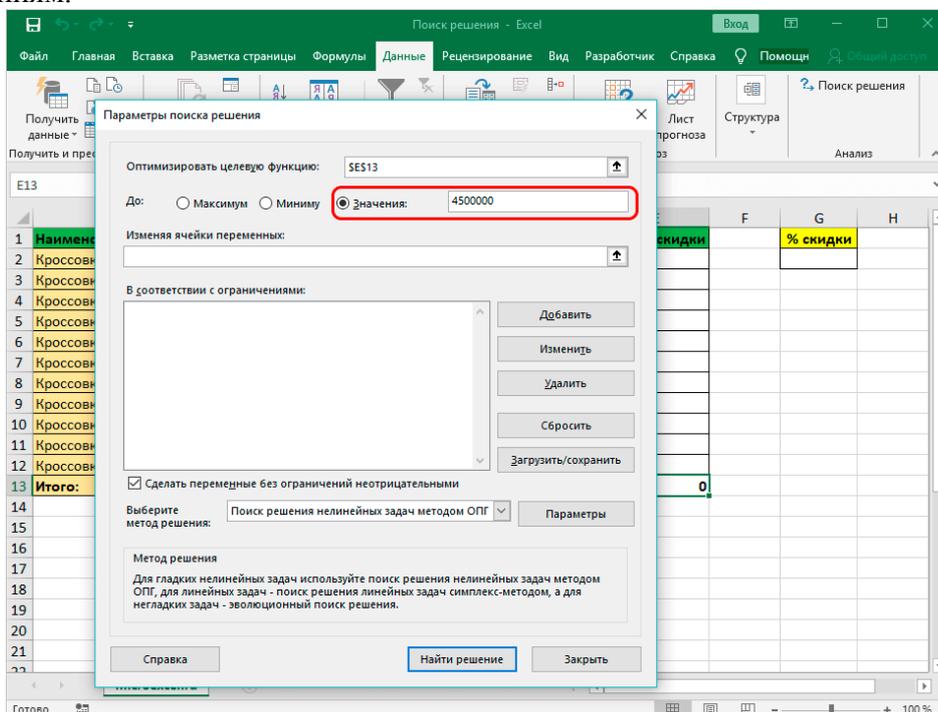


Рис. 294. Заполнение ограничения на значение целевой функции

4. Следующее для заполнения поле – **Изменяя значения переменных**: В него внесите координаты искомой ячейки, содержащей определенное значение. Это значение и есть та самая скидка, которую требуется вычислить. Также, как и с выбором целевой ячейки, координаты можно написать вручную, либо кликнуть по нужной ячейке в самой таблице.

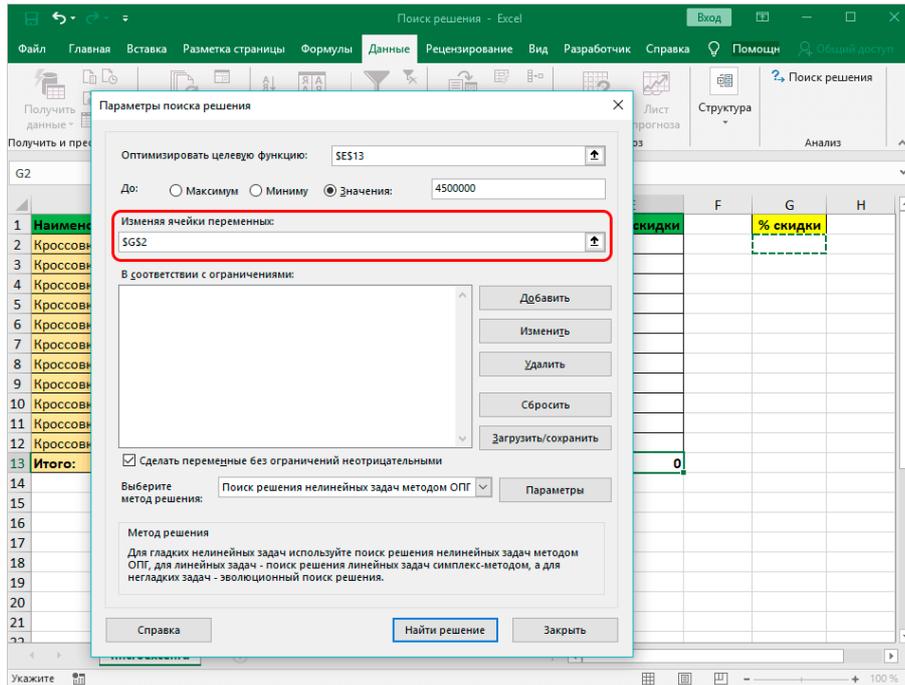


Рис. 295. Заполнение адреса ячейки скидки, которая должна будет изменяться в процессе вычислений

5. Теперь нужно отредактировать раздел **В соответствии с ограничениями:**, в котором задаем ограничения используемых данных. Например, можно исключить десятичные дроби или, скажем, отрицательные числа. Для этого нажмите кнопку **Добавить** (рис. 296).

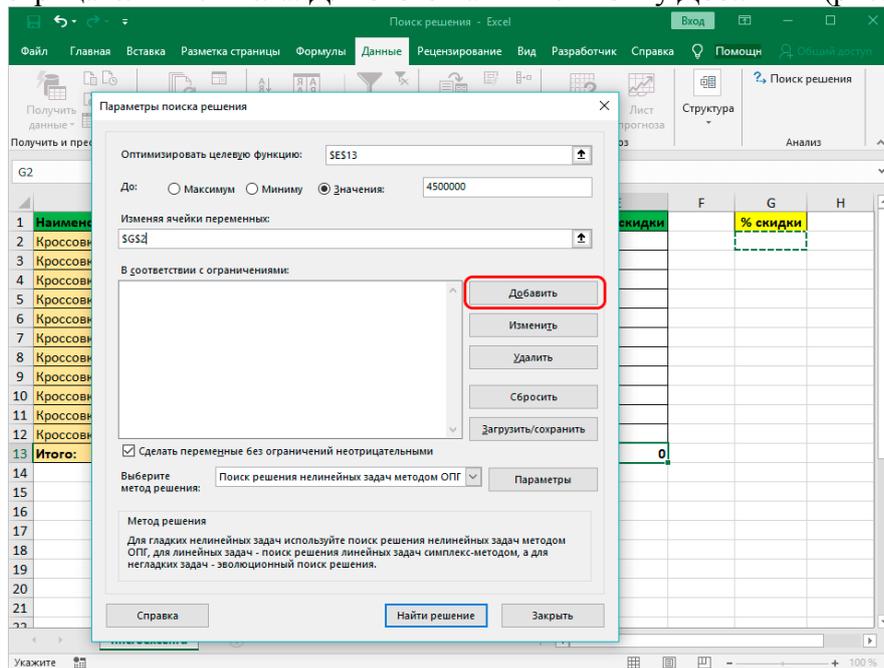


Рис. 296. Расположение кнопки **Добавить** для добавления ограничений

6. Откроется вспомогательное окно, позволяющее добавить ограничения во время вычислений (рис. 297). В первом поле указываем координаты определенной ячейки или области ячеек, для которых это условие должно действовать. Согласно нашей задаче, указываем координаты искомой ячейки, в которой будет выводиться значение скидки. Следующий шаг – определить знак сравнения. Устанавливаем “больше или равно”, чтобы итоговое число не могло быть отрицательным. “Ограничение”, которое устанавливается в третьем поле, в этом случае будет равно цифре 0, поскольку именно относительно этого значения задается условие.

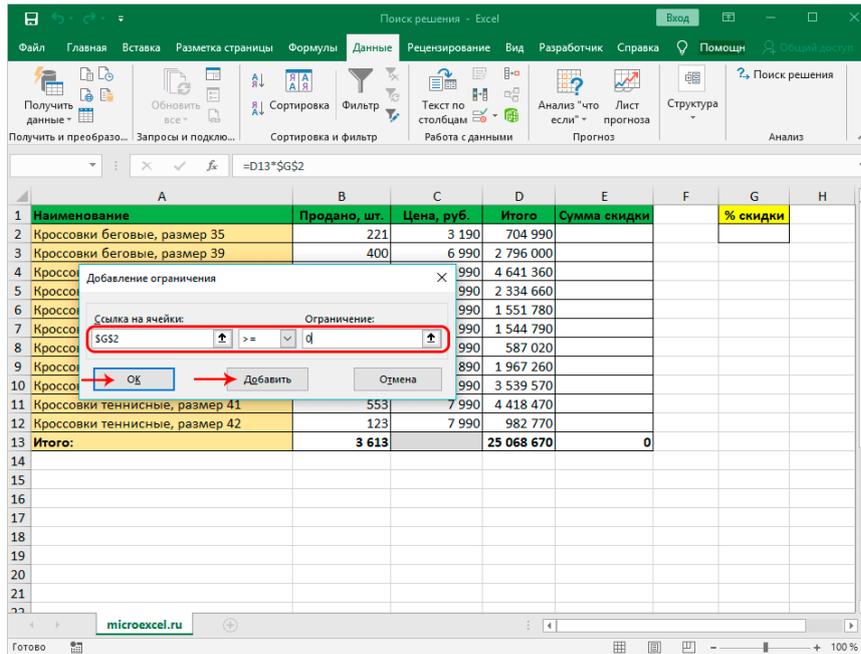


Рис. 297. Вставка первого ограничения

Можно установить еще одно ограничение с помощью кнопки **Добавить**. Дальнейшие действия по его настройке будут аналогичными. По готовности щелкните **ОК**.

7. После выполнения описанных выше действий в самом большом поле окна появится установленное только что ограничение (рис. 298). Список может быть довольно большим и зависит от сложности предполагаемых расчетов, но в данном случае будет достаточно и одного условия.

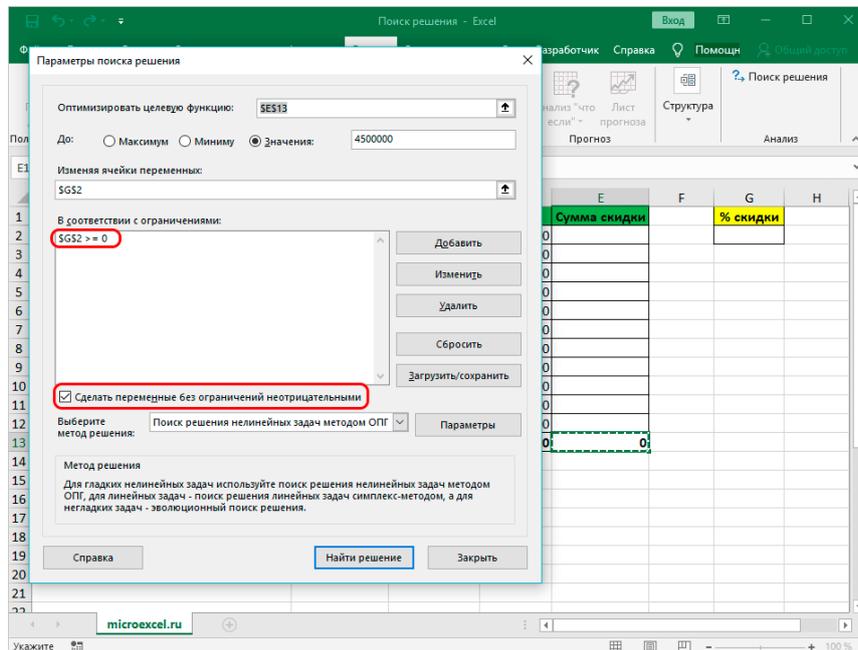


Рис. 298. Результат добавления ограничения

8. Под этим полем также есть опция, позволяющая сделать все остальные переменные, не затрагиваемые ограничениями, неотрицательными. Однако, будьте внимательны и проследите за тем, чтобы между этим параметром и поставленными ограничениями не было противоречия, иначе при расчете в программе может возникнуть конфликт.

9. Также можно задать дополнительные настройки. Ниже справа находится кнопка **Параметры** (рис. 299), позволяющая это сделать. Нажмите на эту кнопку и откройте новое окно.

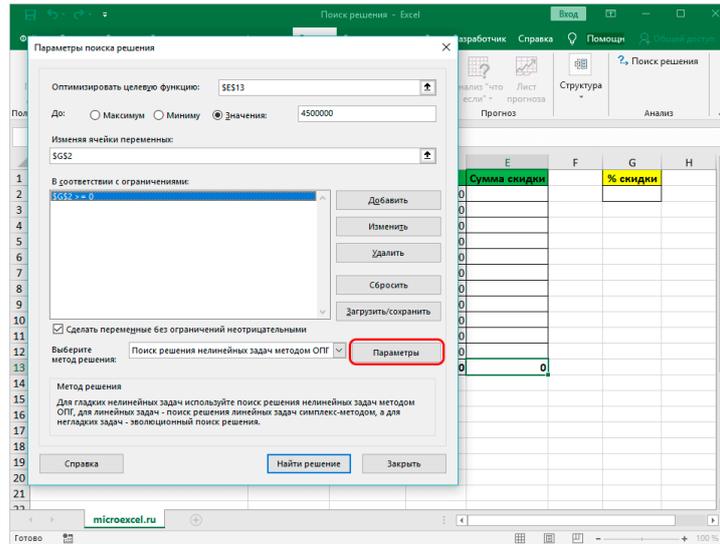


Рис. 299. Расположение кнопки **Параметры**

10. В этих настройках у нас есть возможность установить **Точность ограничения** и **Пределы решения** (рис. 300). В нашем случае задавать данные параметры нет необходимости, поэтому после ознакомления с представленным окном, его можно закрыть, нажав **ОК**.

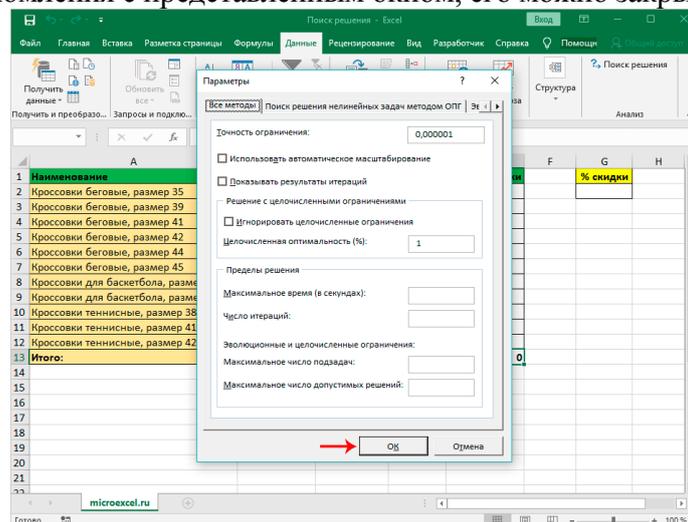


Рис. 300. Установка дополнительных параметров

11. Итак, все настройки выполнены и параметры установлены. Запустите функцию: для поиска оптимального решения. Нажмите кнопку **Найти решение** (рис. 301).

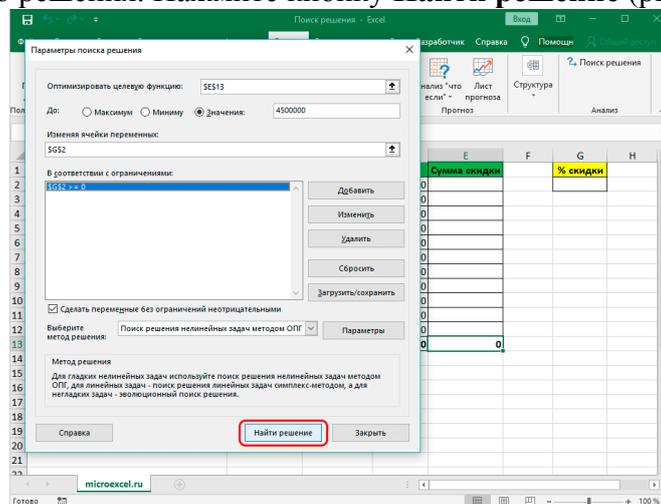


Рис. 301. Расположение кнопки **Найти решение**

12. После этого программа сделает все необходимые расчеты и выдаст результаты в нужных ячейках. При этом сразу же откроется окно **Результаты поиска решения** (рис. 302), где можно сохранить/отменить результаты или настроить параметры поиска заново. Если результаты нас устроят, оставить отметку напротив опции **Сохранить найденное решение** и нажать **ОК**. При этом, если предварительно установить галочку слева от надписи **Вернуться в диалоговое окно параметров поиска решения**, после того, как щелкнуть **ОК**, происходит переключение обратно к настройке функции **Поиск решения**.

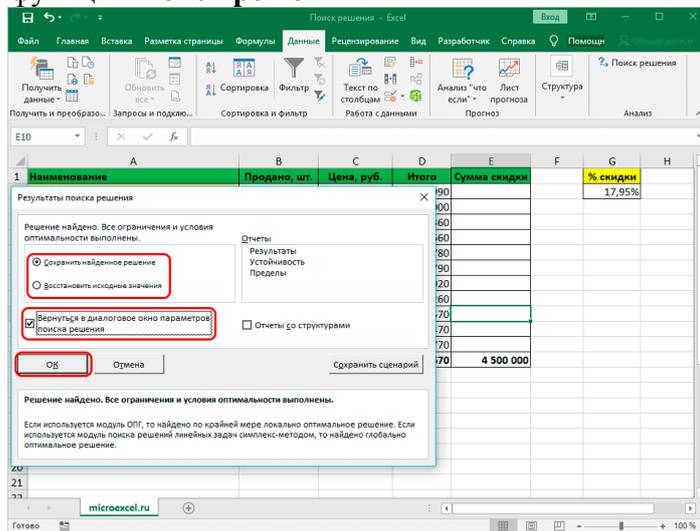


Рис. 302. Окно **Результаты поиска решения**. Параметры для уточнения и перезапуска решения

13. Вполне вероятно, что расчеты могут показаться неправильными, либо возникнет желание немного изменить исходные данные и получить другой результат. В этом случае нужно снова открыть окно с параметрами поиска решения и внимательно посмотреть поля с введенными данными.

14. Если с данными все нормально, можно попробовать задействовать другой метод решения. Для этого щелкаем по текущему варианту и из раскрывшегося перечня выбираем способ, который нам кажется наиболее подходящим:

- Первый ищет решение методом обобщенного приведенного градиента (ОПГ) для нелинейных задач. Стандартно выбран именно этот вариант, но можно попробовать и другие.
- Второй пытается отыскать решение для линейных задач, используя *симплекс-метод*.
- Третий для выполнения поставленной задачи использует *эволюционный поиск*.
- В том случае, если ни один из методов не принес удовлетворительных результатов, стоит проверить данные в таблице и параметрах еще раз, поскольку именно это является самой частой ошибкой в подобного рода задачах.

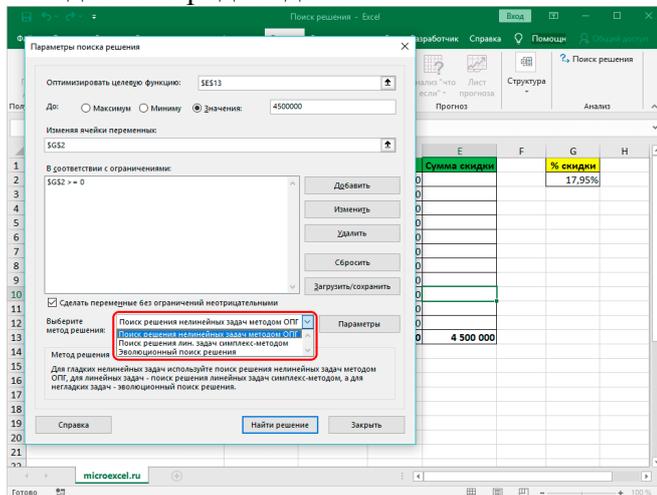


Рис. 303. Возможность выбора другого метода поиска оптимального решения

15. Теперь, когда получена требуемая скидка, примените её, чтобы рассчитать суммы скидок по всем наименованиям. Для этого отметьте первую ячейку столбца **Сумма скидки**, запишите в неё формулу **=D2*\$G\$2** (рис. 304) и нажмите **Enter**. Знаки доллара (абсолютной адресации) ставятся для того, чтобы при растягивании/копировании формулы на другие строки, ячейка **G2** со скидкой оставалась неизменной в расчетах.

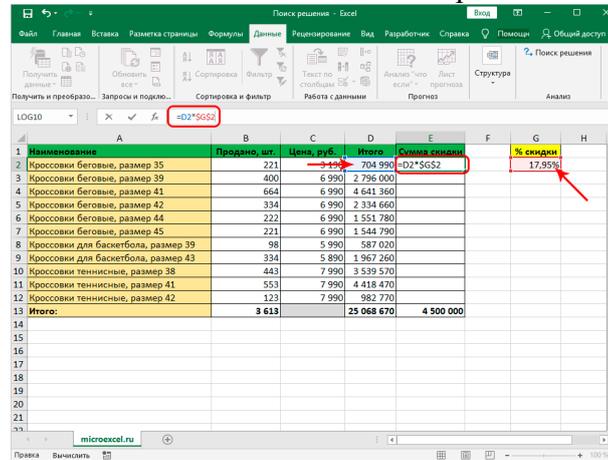


Рис. 304. Установка формулы в ячейку первой ссылки

16. Получена сумма скидки для первого наименования. Теперь наводим курсор на нижний правый угол ячейки с результатом, как только он поменяет форму на крестик, зажав левую кнопку мыши растягиваем формулу на все строки, по которым хотим посчитать аналогичную сумму.

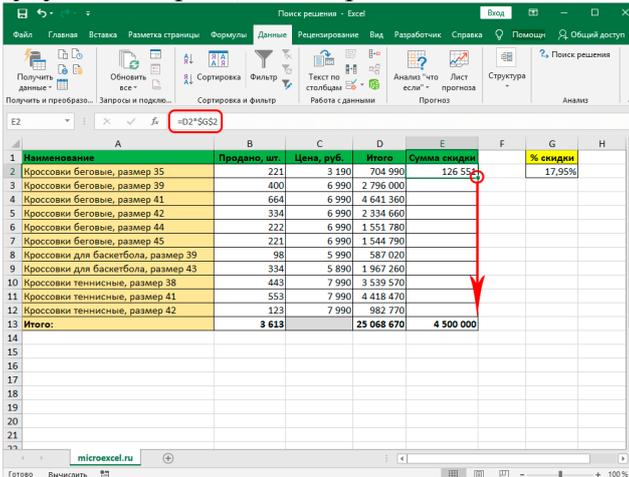


Рис. 305. Подготовка к копированию формулы во все ячейки столбца скидок

17. Теперь наша таблица полностью готова в соответствии с поставленной задачей.

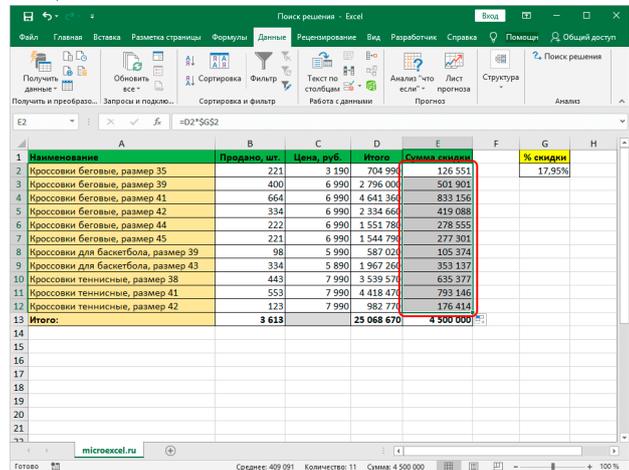


Рис. 306. Выполнены вычисления, получен результат

Закключение

Таким образом, функция **Поиск решения** в Excel может помочь в решении определенных задач, которые достаточно сложно или невозможно решить простыми методами. Однако, проблема в использовании данного способа заключается в том, что по умолчанию данная функция скрыта в программе, из-за чего многие пользователи не догадываются о ее существовании. Также функция довольно трудна в освоении и использовании, но при ее должном изучении, она может принести значительную пользу и облегчить работу.

Задание 3 по Excel

Выполнить одно из заданий по **Оптимизации решения**

Практическая работа № 15. Решение уравнений в Excel

Приближенное решение уравнений (график и 4 метода)

Подробное объяснение читайте по ссылке:

http://www.physchem.chimfak.sfedu.ru/Source/NumMethods/Roots_eq.html

1. Нахождение приближенных корней уравнения $x^2 - \sin x - 1 = 0$.

Сначала постройте график для демонстрационной неявной функции $x^2 - \sin x - 1 = 0$.

Для этого запишите левую часть как функцию: $f(x) = x^2 - \sin x - 1$, оцените приблизительно, где может лежать корень (например, от -3 до 3) и постройте таблицу значений.

В верхнюю ячейку запишите x , ниже в ячейку — y . Затем в ячейки строки для x введите с автозаполнением значения -3 -2 -1 0 1 2 3.

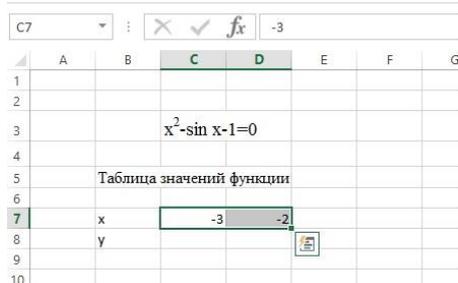


Рис. 307. Подготовка исходных данных по X

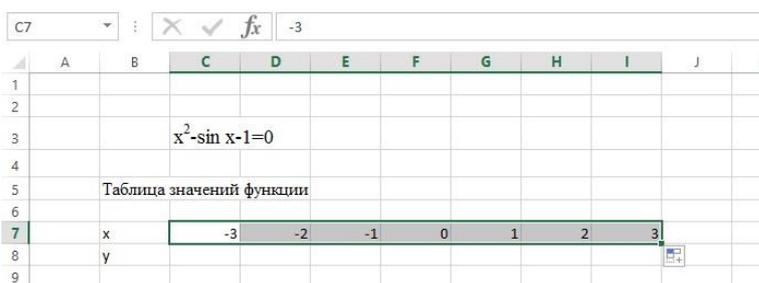


Рис. 308. Строка ячеек полного набора значений по X для графика

Сначала в ячейку для x записали -3, в следующую — -2, выделите обе ячейки (рис. 307) и, взяв на маркер-квадратик в правом нижнем углу (который превратится в плюсик), протяните до значения 3 (рис. 308).

Введите в ячейку $C8$ значение функции $f(x) = x^2 - \sin x - 1$ одним из ранее описанных способов ввода значений функции, а именно: введите в ячейку $C8$ знак = (равно), затем щелкните по ячейке $C7$ (первое значение ячейки x , рис. 309), введите знак умножения (*), затем снова щелкните ячейку $C7$, введите знак минус (-), введите текст $\sin()$, затем снова $C7$ и символы -1 (рис. 310).

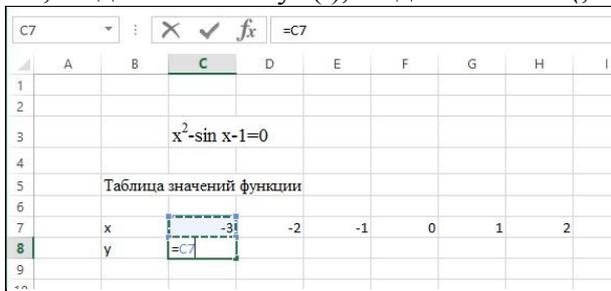


Рис. 309. Вставка начала формулы в первую ячейку Y

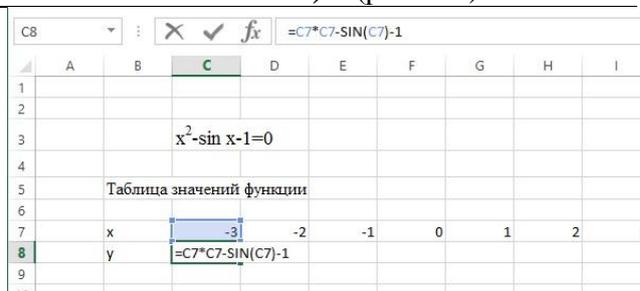


Рис. 310. Вставка полной формулы в первую ячейку Y .

Затем нажмите **Enter**. В ячейке $C8$ получите значение функции **8,14112** при аргументе, равном $x = -3$, полученное в виде ссылки (рис. 311), т.к. в формуле ссылались на адрес аргумента x – ячейку $C7$ (рис. 309). Взяв за маркер-квадратик, протяните вправо до значения $x = 3$ (рис. 312).

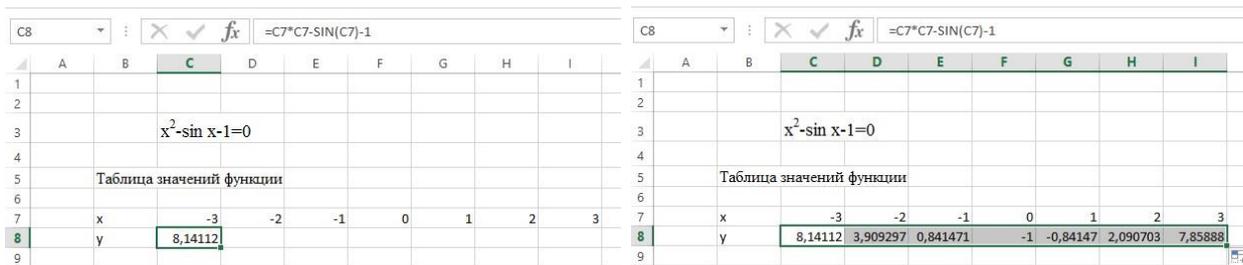


Рис. 311. Получили значение $f(x)$ для первого значения x Рис. 312. Протяжением маркера получили все значения y
 Таблица значений функции построена (рис. 312), можно строить график функции.
 Выделите значения x и y , как на рисунке 313

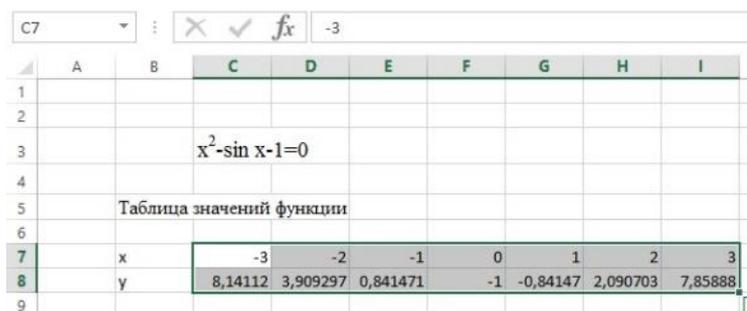


Рис. 313. Выделенный диапазон: значения x (C7:I7) и функции в строке y (C8:I8) и на главной панели выберите **Вставить -> Диаграмму** (вида **График**, рис. 314).

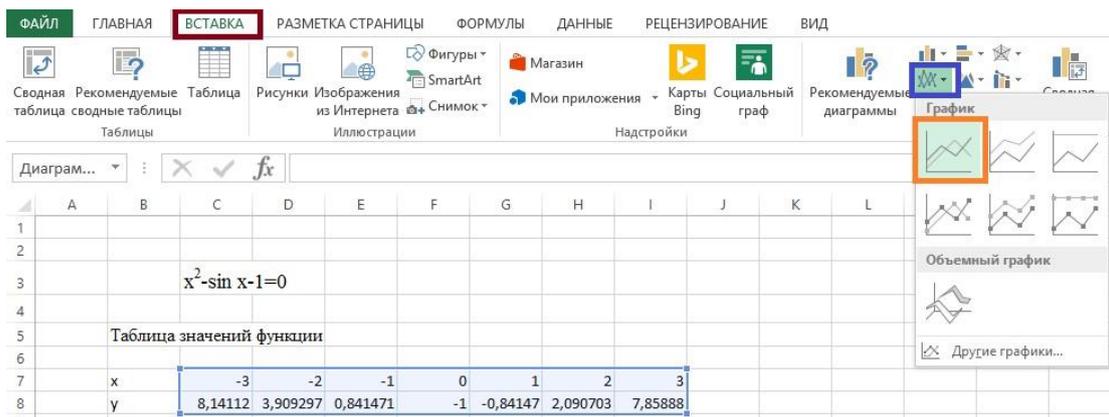


Рис. 314. Выбор графика из блока **Диаграммы** вкладки **Вставка**

Получите график функции $f(x) = x^2 - \sin x - 1$, который еще надо привести к виду, удобному для анализа (рис. 315).

Измените название графика: вместо **Название диаграммы** напишем **График функции $f(x) = x^2 - \sin x - 1$** (рис. 316). Удалите график функции $y=x$: выделите синюю линию (рис. 316) и затем, нажав **Delete**, удалите ее (рис. 317).

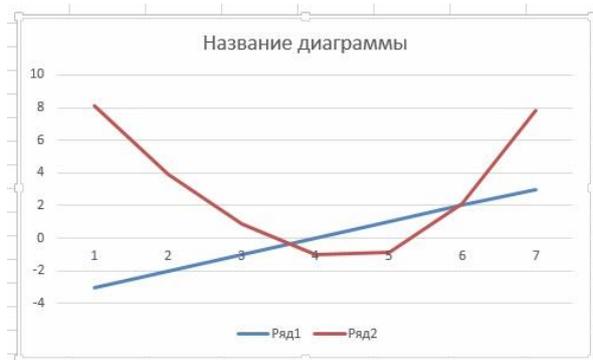


Рис. 315. Первоначальный вид графика

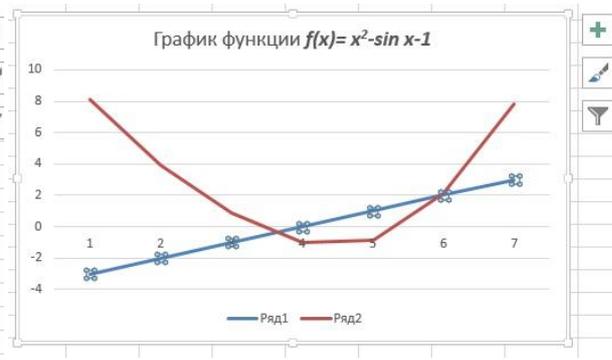


Рис. 316. Выделение линии графика $y=x$

Добавьте заголовок всей работы на листе: **График функции $f(x)=x^2-\sin x-1$** (он понадобится для изменения **Легенды** на графике) в самом верху листа в любой ячейке.

Рассмотрите полученный график (рис. 317). Вертикальная ось слева — это значения функции y . А вот горизонтальная ось мало понятна: на ней отмечены категории (рис. 318), а нужны значения на оси. Поменяйте вид оси. Для этого щелкните по горизонтальной оси левой кнопкой мыши — выделите её (рис. 318).

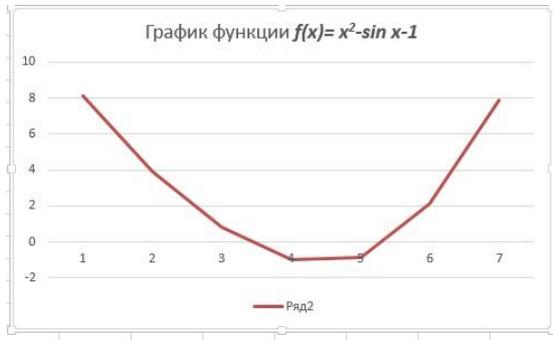


Рис. 317. Вид графика без оси $y=x$

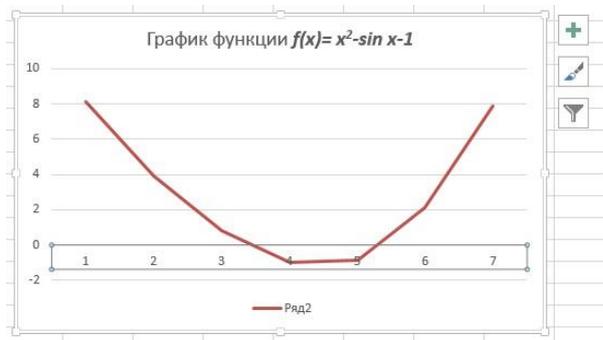


Рис. 318. График с выделенной осью категорий

Затем по выделенной оси щелкните правой кнопкой мыши — появится контекстное окно, в котором выберите пункт **Выбрать данные** (рис. 319).

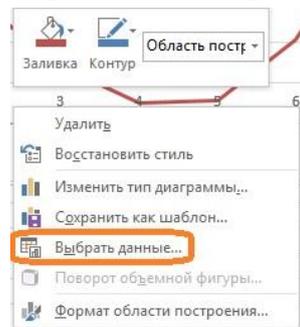


Рис. 319. Пункт **Выбрать данные** в контекстном окне горизонтальной оси

Появится окно **Выбор источника данных** (рис. 320), в котором с помощью элементов правой части (**Изменить**) можно изменить данные (визуализировать значения аргумента), а с помощью элементов левой части (**Элементы легенды**) изменить легенду, выбрав действие **Изменить** и объект **Ряд2** (условное обозначение в графике — **Ряд2**).

Измените сначала легенду. Выделите **Ряд2** и нажмите кнопку **Изменить**. Появится окошко **Изменение ряда** (рис. 321), в котором либо можно вписать имя ряда, либо щелкнуть по выделенной кнопке (рис. 321), перейти во вспомогательное окно **Изменение ряда** (рис. 322) и выбрать нужные данные уже из таблицы данных для построения графика.

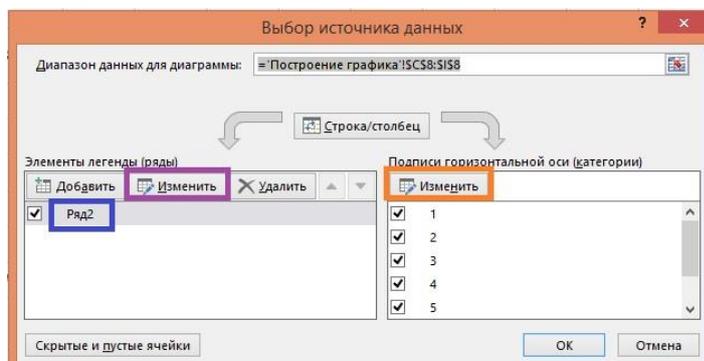


Рис. 320. Окно **Выбор источника данных**

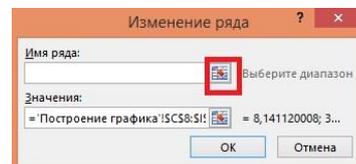


Рис. 321. Окно **Изменение ряда**

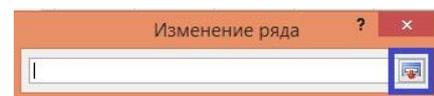


Рис. 322. Поле для выделения данных из таблицы

Щелкните по кнопке (выделена красным, рис. 321), появится еще одно окно **Изменение ряда** (рис. 322), в котором также щелкните по выделенной кнопке (выделена синим,

рис. 322). Теперь найдите нужный нам текст — он находится в выделенной пунктиром ячейке таблицы формирования графика (рис. 323). При этом в окне **Изменение ряда** появится запись:

= *Построение графика* \$D\$4, где \$D\$4 — абсолютный адрес ячейки, в которой находится нужный текст «График функции $f(x)=x^2-\sin x-1$ » (рис. 323).

Пояснение. *Построение графика* — это название листа, в котором строился график, тогда его адрес имеет составной вид, связывающий название листа и адрес ячейки на этом листе с помощью восклицательного знака: = *Построение графика*!\$D\$4 (рис. 324).

Для возврата к графику снова щелкните по выделенной кнопке окна **Изменение ряда** (выделена синим цветом, рис. 324).

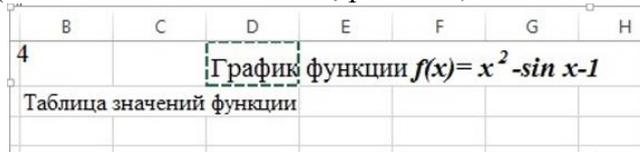


Рис. 323. Текст для названия графика в ячейке \$D\$4

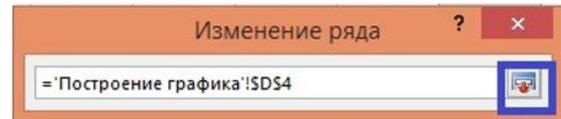


Рис. 324. Адрес ячейки, где находится текст для названия графика

Вернётесь к предыдущему окну **Изменение ряда** (оно уже с нужной ссылкой) и щелкните по кнопке **ОК** (рис. 325). Теперь вернётесь к окну **Выбор источника данных** (рис. 326), в котором **Легенда** уже имеет название **График функции $f(x)=x^2-\sin x-1$** (рис. 326).

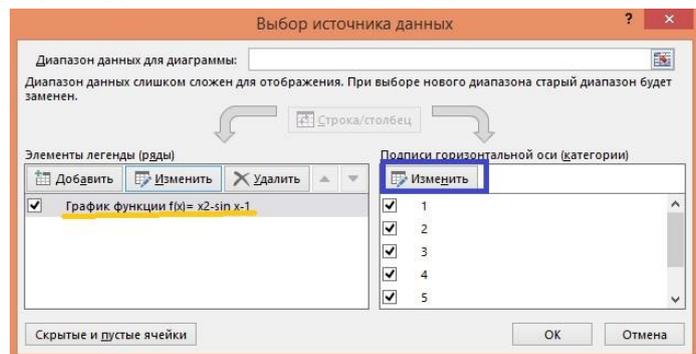
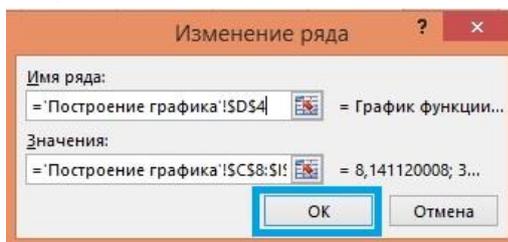


Рис. 325. Вид окна **Изменение ряда** с данными Рис. 326. Окно **Выбор источника данных** с введенным названием

Чтобы посмотреть, что изменилось на графике, щелкните по кнопке **ОК** (рис. 326). Как видно на рисунке 327 **Легенда** изменилась (специально подчеркнута жёлтым).

Теперь вернитесь к правой части окна **Выбор источника данных** (рис. 320). Если Вы из этого окна вышли, снова выделите горизонтальную ось на графике.

Измените *Подписи горизонтальной оси*: это действие выполняют с помощью нажатия кнопки **Изменить** правого блока окна **Выбор источника данных** (рис. 326).

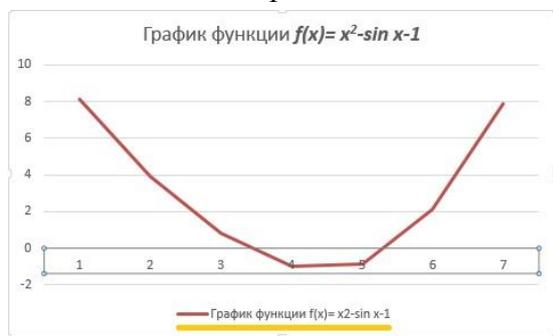


Рис. 327. Вид графика с измененной **Легендой**

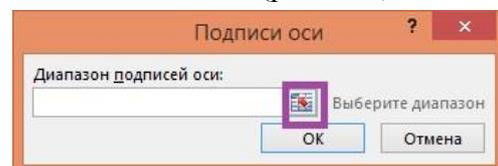


Рис. 328. Окно **Подписи оси** для выбора диапазона данных

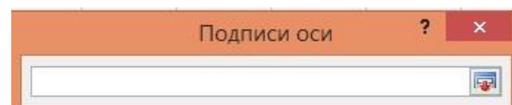


Рис. 329.

Появится окно **Подписи оси** с полем *Диапазон подписей оси* (рис 328). Щелкните по выделенной кнопке (выделена фиолетовым цветом), появится еще одно окно **Подписи оси** (рис. 329).

Перейдите к выбору данных для подписей на оси. Для этого обратитесь к таблице значений функции и выделите строку значений аргумента x (выделено зеленым пунктиром, рис. 330).

	B	C	D	E	F	G	H	I
4			График функции $f(x) = x^2 - \sin x - 1$					
5	Таблица значений функции							
6								
7	x	-3	-2	-1	0	1	2	3
8	y	8,14112	3,909297	0,841471	-1	-0,84147	2,090703	7,85888

Рис. 330. Выделение диапазона данных для оси

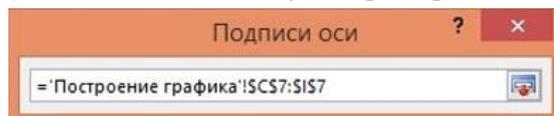


Рис. 331. Абсолютный адрес диапазона данных

В поле окна **Подписи оси** появился текст: *=Построение графика \$C\$7:\$I\$7* — это целая строка данных записанная в абсолютных адресах (рис. 331).

Нажимая выделенную кнопку (рис. 331) или **Enter**, возвратитесь в предыдущее окно **Подписи оси** (рис. 332), в котором видно, какие значения будут подписаны в горизонтальной оси (подчеркнуто синим цветом, рис. 332). Нажав **OK** в текущем окне **Подписи оси** (рис. 332), возвратитесь в окно **Выбор источника данных**, где показаны все сделанные изменения (рис. 333).

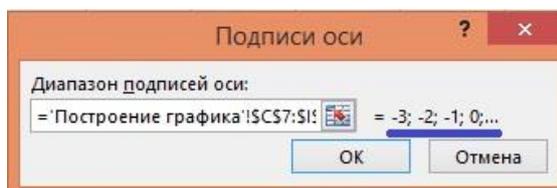


Рис. 332. Результат ввода значений данных оси

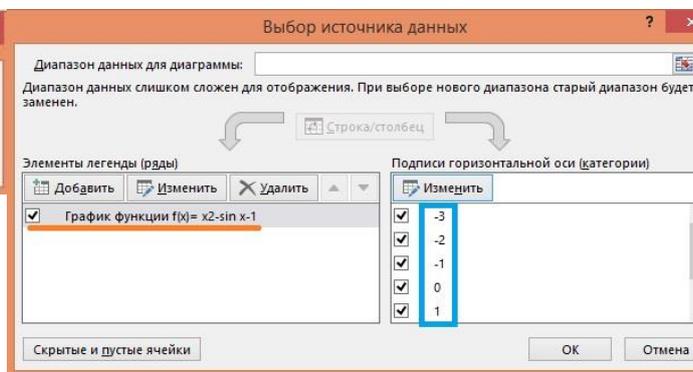


Рис. 333. Изменения, сделанные для изменения Легенды и оси

Нажмите еще раз **OK** и получите требуемый график (рис. 334).

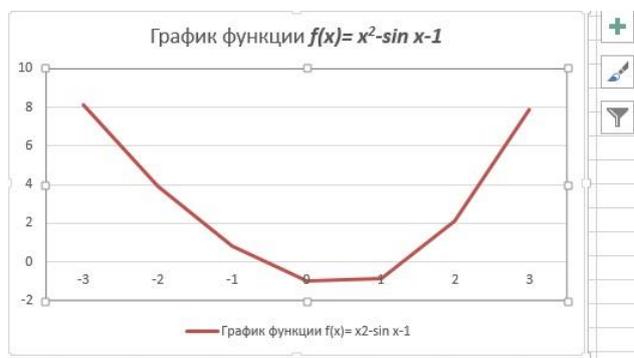


Рис. 334. Результат применения сделанных изменений

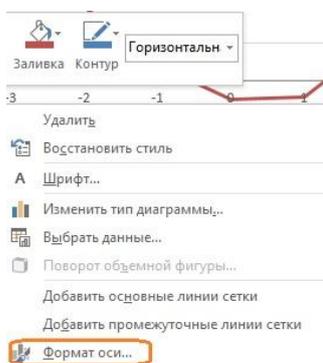


Рис. 335.

Чтобы выделить цветом и толщиной линии саму линию оси (чтобы было понятно, где находится корень уравнения), еще раз щелкните по горизонтальной оси и выберите **Формат оси** (рис. 335).

Справа от рабочего листа откроется вкладка **Формат оси** (рис. 336), в которой сгруппированы свойства оси в две группы: **Параметры оси** и **Параметры текста**. Каждая из этих двух групп подразделяется на подгруппы своих параметров.

Так вкладка **Параметры оси** подразделяется на *заливка и границы*, *эффекты*, *размер и свойства* и *параметры оси*. Активная подгруппа выделяется зеленым цветом, неактивные имеют серый цвет (рис. 336). Выбираете нужную подгруппу. Например,

Заливка и границы , а на ней **Цвет** (рис. 337). Выбираете требуемый цвет, например, черный. Затем переходите со вкладки на лист (щелкните по какой-либо ячейке листа).

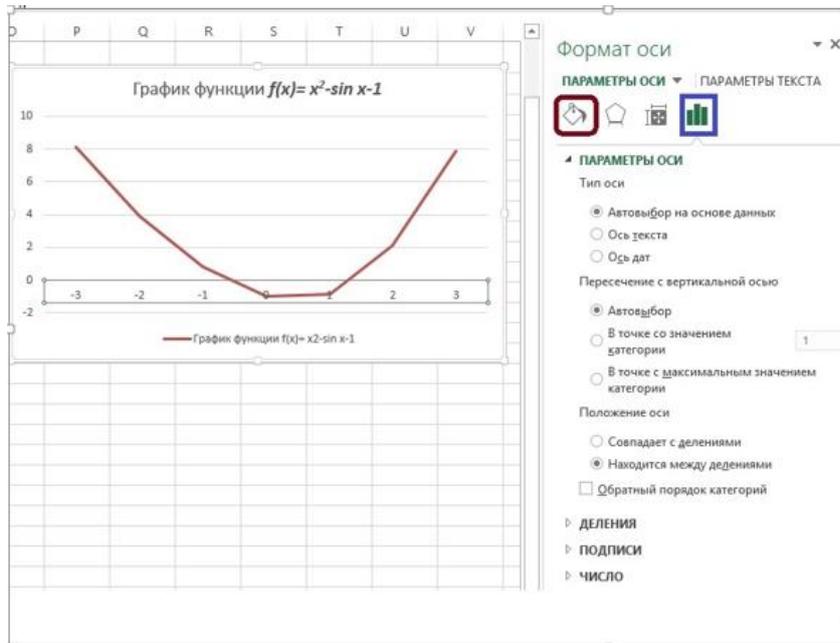


Рис. 336. Выделение оси для оформления ее цветом и толщиной линии.

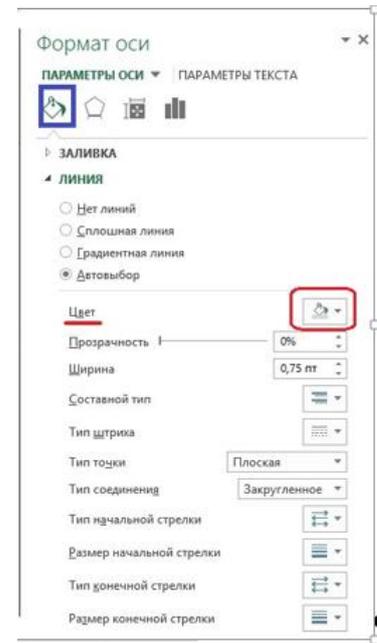


Рис. 337. Вкладка **Формат оси**.

Справа вкладка **Формат оси** с группами **Параметры оси** и **Параметры текста** **Параметры линии** оси
График получает четкую черную ось (рисунок 338).

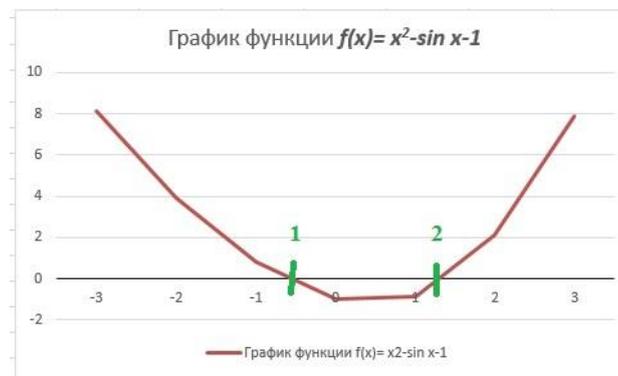


Рис. 338. Вид графика с выделенной черным цветом горизонтальной осью и отмеченным положением корней

Благодаря сделанным преобразованиям, можно легко проанализировать, где лежит корень уравнения и уточнить диапазон значений, между которыми он лежит. Так по рисунку видно, что выбранная функция имеет два корня (рис. 338): один отрицательный (лежит в диапазоне от -1 до 0), другой — положительный (лежит в диапазоне от 1 до 2).

Осуществите поиск одного из корней, например, положительного. Уточните график. Замените значение нижней границы вместо -3 значением 1, выберите меньший шаг, например, шаг не 1, а 0,2.

Измените *Таблицу значений аргументов и функции*. Вместо диапазона [-3, 3] введите диапазон [1, 2]. То есть в первую ячейку данных x введите 1 вместо -3, по вторую ячейку данных x введите 1,2 вместо -2 (рис. 339). Обратите внимание, что значения функций меняются одновременно с изменением аргументов (рис. 340).

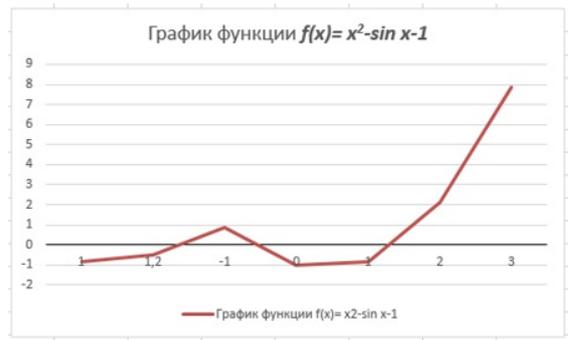
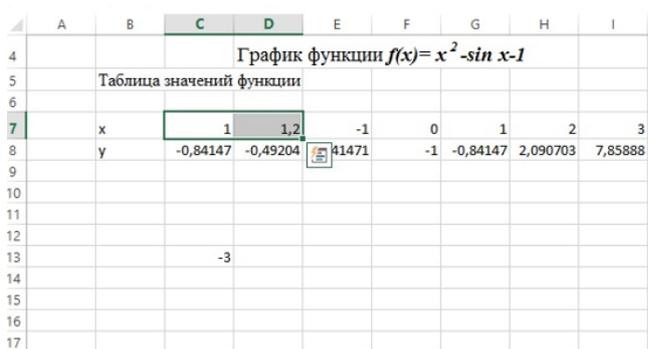


Рис. 339. Изменение начальных значений таблицы данных Рис. 340. Изменение графика с изменением данных

Затем, ухватив за маркер-квадратик, протяните до конца таблицы, изменится диапазон данных и значений (рис. 341). Получите более точный график для нахождения положительного приближенного корня требуемого уравнения (рис. 342).

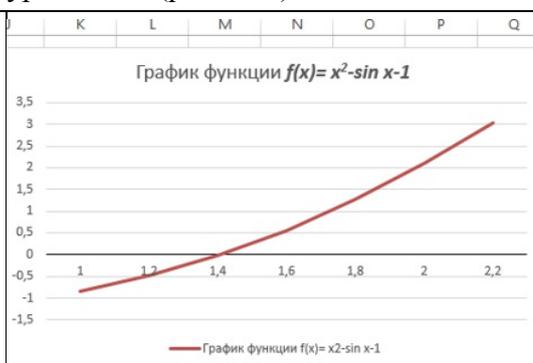
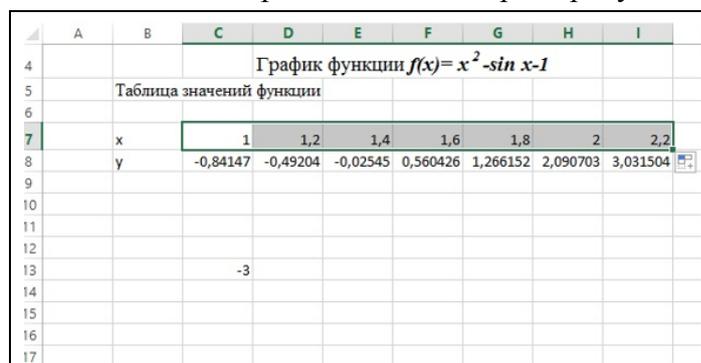


Рис. 341. Изменили весь диапазон таблицы значений x и y

Рис. 342. Вид графика для полученного диапазона

Итак, достигли двух целей: во-первых, научились работать с графиками, во-вторых, уточнили положение корней уравнения.

2. Научимся находить приближенные значения корней уравнений с большой точностью, используя классические методы: итерации, метода Ньютона (касательных), дихотомии и хорд. В качестве дополнительного материала рекомендуется использовать материал сайта, ссылка которого приведена в начале описания практической работы или данную методичку. Каждый метод делайте на отдельном Листе, называя его соответственно названием метода.

Выполнить методы:

- итерации,
- Ньютона (касательных),
- дихотомии и
- хорд.

Прежде всего, отметим, что для вычисления приближенных значений, надо всегда знать, — с какой точностью значение нужно получить и — каков критерий сходимости алгоритма.

Поэтому искать промежуточные значения нужно, во-первых, с запасом значащих цифр, во-вторых, определить критерий сходимости.

Как определяются критерии сходимости при решении уравнений.

2.1. Абсолютное изменение приближения (изменение аргумента) на соседних шагах итерации определяется формулой $|x_k - x_{k-1}| \leq \varepsilon$

2.2. Относительное изменение приближения на соседних шагах итерации определяется следующим соотношением:

$$\frac{|x_k - x_{k-1}|}{x_k} \leq \varepsilon$$

2.3. Близость к нулю вычисленного значения левой части уравнения, или значений функции (иногда это значение называют *невязкой уравнения*, так как для корня невязка равна нулю), $|f(x_k)| \leq \varepsilon$

Нас будет интересовать третий критерий близости.

2.1. Метод простой Итерации.

Открываем новый **Лист**, называем его **Итерация**.

Это простейший из предложенных методов нахождения корней. В качестве итерационной формулы используется выражение независимой переменной из исходного уравнения.

Исходное уравнение: $f(x) = x^2 - \sin x - 1$ путем арифметических преобразований приводится к виду: $x = g(x)$. Этот вид исходного уравнения можно использовать в качестве итерационной формулы.

Преобразование функции для вычисления способом итерации, как правило, не однозначно и, кроме того, совершенно отдельной задачей является оценка

- применимости и
- эффективности

того или иного способа преобразования.

Например, уравнение $x^2 - \sin x = 1$ можно преобразовать тремя способами, например,

так: $x = \sqrt{1 + \sin x}$ или так $x = \arcsin(x^2 - 1)$ или так $x = \frac{1 + \sin x}{x}$

Далее на примерах увидим, что

- не каждое из этих преобразований можно применять и
- применяемые преобразования по-разному эффективны (с разной скоростью сходятся).

Взяв начальное приближение $x_0 = 1$ (для примера ищите правый, положительный корень, а из графика видно, что в качестве левой границы можно взять начальное значение равным 1), следующее значение функции будете получать в качестве нового значения как x_k по одной из выше приведенных формул, где справа стоит формула с аргументом x_{k-1} : например, $x_k = \sqrt{1 + \sin x_{k-1}}$

Постройте три таблицы значений для трех представлений.

1. Таблица значений для представления $x = \sqrt{1 + \sin x}$ $x_k = \sqrt{1 + \sin x_{k-1}}$

В заголовках таблицы запишите последовательные этапы вычислений x , $\sin x$, $1 + \sin x$, $\text{sqrt}(1 + \sin x)$.

Под ячейкой x в ячейку **B5** введите начальное значение **1** (рис. 343). В следующую ячейку по строке **C5** введите

- $=\sin()$, а в скобках либо **B5**,
- либо щелкните по ячейке **B5** (в которой стоит **1**).

Процесс ввода изображен на рисунке 344, а результат на рисунке 345.

	A	B	C	D	E
1					
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$		
3					
4		x	$\sin x$	$1 + \sin x$	$\text{sqrt}(1 + \sin x)$
5		1			
6					
7					
8					

Рис. 343. Ввод начального значения

	A	B	C	D	E
1					
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$		
3					
4		x	$\sin x$	$1 + \sin x$	$\text{sqrt}(1 + \sin x)$
5		$=\sin(B5)$			
6					
7					
8					

Рис. 344. Ввод значения sin (B5)

	A	B	C	D	E
1					
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$		
3					
4		x	$\sin x$	$1 + \sin x$	$\text{sqrt}(1 + \sin x)$
5		1	0,841471		
6					
7					
8					

Рис. 345. Результат ввода sin

В следующую ячейку введите следующий текст $=1+$ и щелкните ячейку **D5** (рис. 346). На рисунке 347 видно значение, которое соответствует значению выражения $1 + \sin 1$.

В ячейку ниже названия $\text{sqrt}(1+\sin x)$ (рис.347) вводим **=корень()**, а внутри скобок ставим **D5** или щелкаем по ячейке **D5** (рис. 347 и 348).

1				
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$	
3				
4	x	sin x	1+sin x	sqrt (1+sin x)
5	1	0,841471	=1+CS	
6				
7				

1				
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$	
3				
4	x	sin x	1+sin x	sqrt (1+sin x)
5	1	0,841471	=1,841471	=корень(D5)
6				
7				

1				
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$	
3				
4	x	sin x	1+sin x	sqrt (1+sin x)
5	1	0,841471	1,841471	1,3570081
6				
7				

Рис. 346. Заполнение ячейки **D5** формулой Рис. 347. Заполнение ячейки **E5** формулой Рис. 348. Результат выполнения итерации

Теперь в ячейку **B6** столбца аргумента **x** скопируйте значение функции в ячейке **E5**,

- записав **=E5** или
- = и щелкнуть ячейку **E5** (рис. 349).

Теперь выделите ячейки **C5:E5** (рис. 350) и опустите за маркер-квадратик вниз на ячейку. Результат на рисунке 351.

1				
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$	
3				
4	x	sin x	1+sin x	sqrt (1+sin x)
5	1	0,841471	1,841471	1,3570081
6	=E5			
7				

1				
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$	
3				
4	x	sin x	1+sin x	sqrt (1+sin x)
5	1	0,841471	1,841471	1,3570081
6	1,357008			
7				

1				
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$	
3				
4	x	sin x	1+sin x	sqrt (1+sin x)
5	1	0,841471	1,841471	1,3570081
6	1,357008	0,977234	1,977234	1,406141602
7				

Рис. 349. Копирование ячейки первой итерации в ячейку для **x** второй итерации Рис. 350. Выделение 3-х ячеек для копирования формул вниз на 1 ячейку Рис. 351. Результат копирования формул 3-х ячеек

Далее выделите ячейки **C6:E6** (рис. 352) и, взяв за маркер-квадратик опустите до тех пор, пока значения в столбце **x** в двух соседних ячейках не совпадут (рис. 353).

Корень с точностью до 6 знаков после запятой найден. Однако, интересно, сколько было выполнено итераций. Для этого добавьте слева столбец **№ итерации** и вставьте значения 0 и 1 в последовательные ячейки, выделите ячейки, содержащие 0 и 1 (рис. 354).

1				
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$	
3				
4	x	sin x	1+sin x	sqrt (1+sin x)
5	1	0,841471	1,841471	1,3570081
6	1,357008	0,977234	1,977234	1,406141602
7				

1				
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$	
3				
4	x	sin x	1+sin x	sqrt (1+sin x)
5	1	0,841471	1,841471	1,3570081
6	1,357008	0,977234	1,977234	1,406141602
7	1,406142	0,986475	1,986475	1,409423644
8	1,409424	0,987008	1,987008	1,409612592
9	1,409613	0,987038	1,987038	1,409623354
10	1,409623	0,98704	1,98704	1,409623967
11	1,409624	0,98704	1,98704	1,409624002
12	1,409624	0,98704	1,98704	1,409624004

1					
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$		
3					
4	№итерации	x	sin x	1+sin x	sqrt (1+sin x)
5	0	1	0,841471	1,841471	1,3570081
6	1	1,357008	0,977234	1,977234	1,406141602
7		1,406142	0,986475	1,986475	1,409423644
8		1,409424	0,987008	1,987008	1,409612592
9		1,409613	0,987038	1,987038	1,409623354
10		1,409623	0,98704	1,98704	1,409623967
11		1,409624	0,98704	1,98704	1,409624002
12		1,409624	0,98704	1,98704	1,409624004

Рис. 352. Выделение диапазона **C6:E6** для копирования формул Рис. 353. Протягивание 4-х ячеек вниз до совпадения значений **x** Рис. 354. Вставка столбцов для **№ итерации** и заполнение первых двух

Взяв за маркер-квадратик, протянем до строки последних значений, получим количество итераций, необходимых для остановки алгоритма **№=7** (рисунок слева ниже).

1					
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$		
3					
4	№итерации	x	sin x	1+sin x	sqrt (1+sin x)
5	0	1	0,841471	1,841471	1,3570081
6	1	1,357008	0,977234	1,977234	1,406141602
7	2	1,406142	0,986475	1,986475	1,409423644
8	3	1,409424	0,987008	1,987008	1,409612592
9	4	1,409613	0,987038	1,987038	1,409623354
10	5	1,409623	0,98704	1,98704	1,409623967
11	6	1,409624	0,98704	1,98704	1,409624002
12	7	1,409624	0,98704	1,98704	1,409624004

1					
2			$x_k = \text{sqrt}(1 + \sin x_{k-1})$		
3					
4	№итерации	x	sin x	1+sin x	sqrt (1+sin x)
5	0	1	0,841471	1,841471	1,3570081
6	1	1,357008	0,977234	1,977234	1,406141602
7	2	1,406142	0,986475	1,986475	1,409423644
8	3	1,409424	0,987008	1,987008	1,409612592
9	4	1,409613	0,987038	1,987038	1,409623354
10	5	1,409623	0,98704	1,98704	1,409623967
11	6	1,409624	0,98704	1,98704	1,409624002
12	7	1,409624	0,98704	1,98704	1,409624004

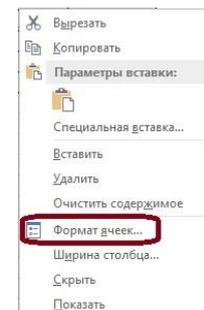


Рис. 355. Заполнение столбца **№ итерации** протягиванием Рис. 356. Выделение столбцов **B:E** для изменения формата Рис. 357. В контекстном меню выбрать **Формат ячеек**

Если же нам нужна большая точность, например, до 10 знаков после запятой, немного видоизмените таблицу. Выделите названия столбцов **В:Е** (рис. 356) и, щелкните правой кнопкой мыши. Получите контекстное меню, в котором выберите пункт **Формат ячеек** (рис. 357) и вкладку **Число**. В следующем окне, выделив в столбце типов форматов **Числовой**, укажите **Число десятичных знаков 10**.

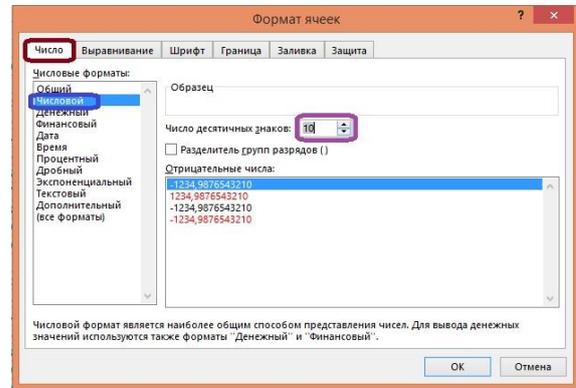
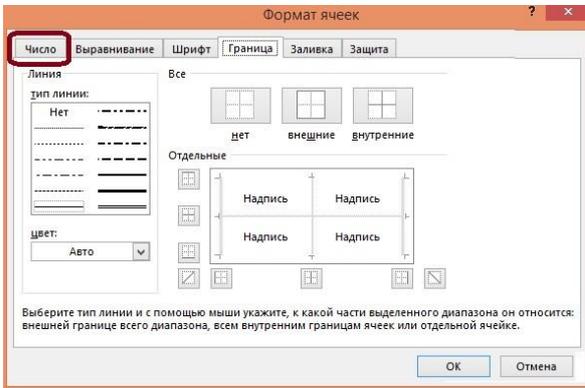


Рис. 358. Выбор вкладки **Число** в пункте **Формат ячеек**

Рис. 359. Выбор точности числа

Получится следующая таблица (рис. 360). Вычисления нужно продолжить. Выделите последнюю строку (рис. 361).

	A	B	C	D	E
1					
2		$x_k = \sqrt{1 + \sin x_{k-1}}$			
3					
4	<i>Итерации</i>	<i>x</i>	<i>sin x</i>	<i>1+sin x</i>	<i>sqrt (1+sin x)</i>
5	0	1,0000000000	0,8414709848	1,8414709848	1,3570081005
6	1	1,3570081005	0,9772342057	1,9772342057	1,4061416023
7	2	1,4061416023	0,9864750088	1,9864750088	1,4094236442
8	3	1,4094236442	0,9870076600	1,9870076600	1,4096125922
9	4	1,4096125922	0,9870380013	1,9870380013	1,4096233544
10	5	1,4096233544	0,9870397284	1,9870397284	1,4096239670
11	6	1,4096239670	0,9870398267	1,9870398267	1,4096240019
12	7	1,4096240019	0,9870398323	1,9870398323	1,4096240039
13					
14					
15					

	A	B	C	D	E
1					
2		$x_k = \sqrt{1 + \sin x_{k-1}}$			
3					
4	<i>Итерации</i>	<i>x</i>	<i>sin x</i>	<i>1+sin x</i>	<i>sqrt (1+sin x)</i>
5	0	1,0000000000	0,8414709848	1,8414709848	1,3570081005
6	1	1,3570081005	0,9772342057	1,9772342057	1,4061416023
7	2	1,4061416023	0,9864750088	1,9864750088	1,4094236442
8	3	1,4094236442	0,9870076600	1,9870076600	1,4096125922
9	4	1,4096125922	0,9870380013	1,9870380013	1,4096233544
10	5	1,4096233544	0,9870397284	1,9870397284	1,4096239670
11	6	1,4096239670	0,9870398267	1,9870398267	1,4096240019
12	7	1,4096240019	0,9870398323	1,9870398323	1,4096240039
13					
14					
15					

Рис. 360. Таблица результатов с точностью до 10 знаков
Рис. 361. Выделение строку формул для продолжения вычислений

Протянув за маркер-квадратик до совпадения всех знаков после запятой в двух последовательных ячейках столбца **x**, получим новую таблицу (рис. 362). Теперь потребовалось 10 итераций для получения приближенного корня уравнения.

	A	B	C	D	E
1					
2		$x_k = \sqrt{1 + \sin x_{k-1}}$			
3					
4	<i>Итерации</i>	<i>x</i>	<i>sin x</i>	<i>1+sin x</i>	<i>sqrt (1+sin x)</i>
5	0	1,0000000000	0,8414709848	1,8414709848	1,3570081005
6	1	1,3570081005	0,9772342057	1,9772342057	1,4061416023
7	2	1,4061416023	0,9864750088	1,9864750088	1,4094236442
8	3	1,4094236442	0,9870076600	1,9870076600	1,4096125922
9	4	1,4096125922	0,9870380013	1,9870380013	1,4096233544
10	5	1,4096233544	0,9870397284	1,9870397284	1,4096239670
11	6	1,4096239670	0,9870398267	1,9870398267	1,4096240019
12	7	1,4096240019	0,9870398323	1,9870398323	1,4096240039
13	8	1,4096240039	0,9870398326	1,9870398326	1,4096240040
14	9	1,4096240040	0,9870398327	1,9870398327	1,4096240040
15	10	1,4096240040	0,9870398327	1,9870398327	1,4096240040

Рис. 362. Таблица значений для получения приближенного корня с точностью до 10 знаков после запятой

Аналогичным образом начните строить таблицу значений функции $x = \arcsin(x^2 - 1)$

Постройте строку заголовков, в строку итераций введите **0**, в строку **x** — начальное значение аргумента, равное 1 (рис. 363).

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1			
6					
7					

Рис. 363. Заполнение начальным значением 1 ячейки **x**

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	=H5*H5		
6					
7					

Рис. 364. Заполнение формулой ячейки **I5**

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	1	=I5-1	
6					
7					

Рис. 365. Заполнение формулой ячейки **J5**

Введите в ячейку **I5** адрес **=H5** или щелкните левой кнопкой мыши по **H5**, затем ***** и снова **H5**, получая в ячейке **I5** значение x^2 (рис. 364). В следующей ячейке этой строки введите значение предыдущей ячейки **=I5**, а затем **-1**. Получите значение x^2-1 (рис. 365). Наконец в последнюю ячейку **K5** введите функцию $\arcsin(x^2-1)$, которая в Excel имеет название **asin**: **=asin(J5)** (рис. 366).

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	1	0	=asin(J5)
6					

Рис. 366. В ячейку **K5** введена окончательная формула

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	1	0	0
6					

Рис. 367. Результат выполнения первой итерации

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	1	0	0
6					=K5

Рис. 368. Копирование значения результата первой итерации в ячейку **x** второй итерации

Нажав **Enter**, получите строку значений (рис. 367). Скопируйте, как и в предыдущем примере, значение ячейки **K5** в ячейку **H6**: **=K5** (368). Получите следующее приближенное значение **x** по этому алгоритму. Скопируйте значения ячеек **I5:K5** в соответствующие ячейки **I6:K6** с помощью выделения ячеек **I5:K5** (рис. 369) и протягивания маркера-квадратика на строку ниже. Кроме этого, добавим в столбец **№итерации** число 1 (рис. 370).

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	1	0	0
6					

Рис. 369. Выделение 3-х ячеек для дальнейшего копирования

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	1	0	0
6	1	0	0	-1	-1,570796327
7					

Рис. 370. Копирование формул выделенных ячеек и добавление 1 в столбец итераций

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	1	0	0
6	1	0	0	-1	-1,570796327

Рис. 371. Выделение всей строки для копирования формул

Далее выделите всю строку таблицы **G6:K6** (рис. 371) и протяните на одну-две ячейки вниз (рис. 372). Видно, что уже в строке 6 в ячейке **K7** появляется неопределенность, а в строке 7 неопределенность уже во всех вычисляемых ячейках. Это означает, что *выбрано плохое представление исходной функции для этого метода*.

	G	H	I	J	K
1					
2			$x = \arcsin(x^2 - 1)$		
3					
4	№итерации	x	x*x	x*x-1	arcsin(x*x-1)
5	0	1	1	0	0
6	1	0	0	-1	-1,570796327
7	2	-1,5708	2,467401	1,467401	#ЧИСЛО!
8	3	#ЧИСЛО!	#ЧИСЛО!	#ЧИСЛО!	#ЧИСЛО!

Рис. 372. Копирование **G6:K6** на 2 строки вниз

	M	N	O	P	Q
1					
2			$x = \frac{1 + \sin x}{x}$		
3					
4	№итерации	x	sin x	1+sin x	(1+sin x)/x
5	0	1			
6					

Рис. 373. Оформление таблицы названий и ввод начальных значений

	M	N	O	P	Q
1					
2			$x = \frac{1 + \sin x}{x}$		
3					
4	№итерации	x	sin x	1+sin x	(1+sin x)/x
5	0	1	#####	#####	
6					

Рис. 374. Копирование формул первой таблицы в третью

Наконец, рассмотрим последнее представление $x = \frac{1 + \sin x}{x}$ исходной функции $f(x) = x^2 - \sin x - 1$.

Снова постройте таблицу для последнего представления. Названия столбцов этой таблицы очень похожи на названия столбцов первой таблицы за исключением последней ячейки с наименованием $(1 + \sin x)/x$. Скопируйте заголовки из первой таблицы и добавьте название для последней ячейки.

Далее введите **0** в ячейку для первого значения **№ итерации** и **1** в качестве начального значения для x (рис. 373).

Можно аналогично предыдущим примерам построить таблицу дальше. А можно попытаться оптимизировать процесс. В первой таблице есть адресные значения для \sin и $1 + \sin$, поэтому можно скопировать содержимое ячеек **C5:D5** и вставить их в ячейки **O5:P5**, тем самым получить вид таблицы как на рисунке 374.

Обратите внимание, что после копирования получили в ячейках **O5:P5** текст #####. Это не ошибка. Это несоответствие ширины ячейки и величины мантииссы числа (вспомним, что увеличивали дробную часть в первой таблице до 10 знаков, а в последней таблице по умолчанию 6 знаков в дробной части). Увеличьте, как и в первой таблице, ширину столбцов диапазона **N:Q**, аналогично действиям для первой таблицы (рис. 375).

Видно, что значения первой и третьей таблицы совпадают (375), что естественно.

Теперь введите формулу для последней ячейки: **=P5/N5** либо текстом, либо используя щелчками выделение ячеек **P5** и **N5** и знака деления (/) между ними (рис. 376).

	M	N	O	P	Q
1					
2		$x = \frac{1 + \sin x}{x}$			
3		x			
4	№итерации	x	$\sin x$	$1 + \sin x$	$(1 + \sin x)/x$
5	0	1,0000000000	0,8414709848	1,8414709848	
6					

Рис. 375. Значения первой строки с точностью до 10 знаков после запятой

	M	N	O	P	Q
1					
2		$x = \frac{1 + \sin x}{x}$			
3		x			
4	№итерации	x	$\sin x$	$1 + \sin x$	$(1 + \sin x)/x$
5	0	1,0000000000	0,8414709848	1,8414709848	=P5/N5
6					

Рис. 376. Включение формулы в последний столбец

Нажав **Enter**, получите первую строку с числовыми значениями. Аналогично, работе с предыдущими таблицами, скопируйте содержимое ячейки **Q5** в ячейку **N6** (рис. 377) и добавьте в ячейку **M6** столбца **№итерации** значение **1**.

	M	N	O	P	Q
1					
2		$x = \frac{1 + \sin x}{x}$			
3		x			
4	№итерации	x	$\sin x$	$1 + \sin x$	$(1 + \sin x)/x$
5	0	1,0000000000	0,8414709848	1,8414709848	1,8414709848
6	1	=Q5			

Рис. 377. Копирование формулы ячейки **Q5** в **N2**

	M	N	O	P	Q
1					
2		$x = \frac{1 + \sin x}{x}$			
3		x			
4	№итерации	x	$\sin x$	$1 + \sin x$	$(1 + \sin x)/x$
5	0	1,0000000000	0,8414709848	1,8414709848	1,8414709848
6	1	1,8414709848			

Рис. 378. Результат копирования формулы ячейки **Q5** в **N2**

Взяв за маркер-квадратик, протяните вниз на одну ячейку вниз, получите следующую строку значений, получите вид таблицы, представленный на рисунке 379.

	M	N	O	P	Q
1					
2		$x = \frac{1 + \sin x}{x}$			
3		x			
4	№итерации	x	$\sin x$	$1 + \sin x$	$(1 + \sin x)/x$
5	0	1,0000000000	0,8414709848	1,8414709848	1,8414709848
6	1	1,8414709848	0,9635907245	1,9635907245	1,0663164072

Рис. 379. Вид таблицы после копирования первой строки формул

	M	N	O	P	Q
1					
2		$x = \frac{1 + \sin x}{x}$			
3		x			
4	№итерации	x	$\sin x$	$1 + \sin x$	$(1 + \sin x)/x$
5	0	1,0000000000	0,8414709848	1,8414709848	1,8414709848
6	1	1,8414709848	0,9635907245	1,9635907245	1,0663164072
7					

Рис. 380. Выделение строки формул для дальнейшего копирования протягиванием

Выделяя далее всю строку 6 таблицы М6:Q6 (рис. 380), протяните её вниз. Видно, что в этом случае, количество итераций оказывается гораздо больше №=198 (рис. 381).

	M	N	O	P	Q
1					
2		$x = \frac{1 + \sin x}{x}$			
3					
4	Итерация	x	sin x	1+sin x	(1+sin x)/x
5	0	1,0000000000	0,8414709848	1,8414709848	1,8414709848
6	1	1,8414709848	0,9635907245	1,9635907245	1,0663164072
7	2	1,0663164072	0,8754259748	1,8754259748	1,7587893820
8	3	1,7587893820	0,9823812866	1,9823812866	1,1271283003
9	4	1,1271283003	0,9031832230	1,9031832230	1,6885240328
10	5	1,6885240328	0,9930780939	1,9930780939	1,1803670277
11	6	1,1803670277	0,9247457601	1,9247457601	1,6306332817
12	7	1,6306332817	0,9982103035	1,9982103035	1,2254197960
13	8	1,2254197960	0,9409480444	1,9409480444	1,5839045940
	* * * * *				
195	190	1,4096240039	0,9870398327	1,9870398327	1,4096240041
196	191	1,4096240041	0,9870398327	1,9870398327	1,4096240040
197	192	1,4096240040	0,9870398327	1,9870398327	1,4096240040
198	193	1,4096240040	0,9870398327	1,9870398327	1,4096240040

Рис. 381. Таблица итераций получения последовательных значений для третьего представления

2.2. Метод Ньютона (касательных)

Метод Ньютона основан на линеаризации функции $f(x)$ вблизи приближенного значения и нахождения точки пересечения полученной линии с осью абсцисс.

Тангенс угла наклона касательной равен значению производной в точке касания:
 $tg \alpha = f'(x_i)$

Координата точки пересечения будет x_{i+1} , которую находят по формуле

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{f(x_i)}{f'(x_i)}$$

— это и есть итерационная формула метода Ньютона.

Пример. Для уравнения $x^2 - \sin x - 1 = 0$ имеем:

$$f(x) = x^2 - \sin x - 1$$

$$f'(x) = 2x - \cos x$$

Создайте новый Лист, который называете **Метод Ньютона**.

Постройте таблицу также, как и для **Метода Итерации** (рис)

	A	B	C	D	E	F	G	H	I	J	K	L
1			$f(x) = x^2 - \sin x - 1 = x^2 - (\sin x + 1)$									
2			$f'(x) = 2x - \cos x$									
3			Таблица значений									
4												
5												
6		Итерация	x	A=x*x	sin x	B=(sin x) + 1	f=A-B	D=2*x	C=cos x	f1=D-C	L=f/f1	x-L
7												
8												

Рис. 382. Построение таблицы для метода Ньютона

Заполняем ячейку **Итерации** значением **0**, вводим исходное значение $x_0 = 1$.

Остальные графы заполняем аналогично Методу итераций.

	A	B	C	D	E	F	G	H	I	J	K	L	
1			$f(x) = x^2 - \sin x - 1 = x^2 - (\sin x + 1)$										
2			$f(x) = 2x - \cos x$										
3													
4			Таблица значений										
5													
6			№итераций	x	A=x*x	sin x	B=(sin x) +1	f=A-B	D=2*x	C=cos x	f1=D-C	L=f/f1	x-L
7			0	1	1	0,841471	1,841470985	-0,84147	2	0,540302	1,459698	-0,57647	1,576469
8			1	1,576469	2,485256	0,999984	1,999983908	0,485272	3,152939	-0,00567	3,158612	0,153634	1,422835
9			2	1,422835	2,024459	0,989074	1,989073658	0,035385	2,84567	0,147422	2,698248	0,013114	1,409721
10			3	1,409721	1,987312	0,987055	1,987055338	0,000257	2,819441	0,16038	2,659061	9,66E-05	1,409624
11			4	1,409624	1,98704	0,98704	1,987039834	1,39E-08	2,819248	0,160475	2,658773	5,25E-09	1,409624
12			5	1,409624	1,98704	0,98704	1,987039833	0	2,819248	0,160475	2,658773	0	1,409624

Рис. 383. Таблица итераций получения последовательных значений для метода Ньютона (6 знаков после запятой)

Видно, что в результате работы метода Ньютона потребовалось 5 итераций для нахождения приближенного корня с точностью до 6 знаков после запятой (рис. 383). Если увеличить количество знаков после запятой до 10, потребуется только 6 итераций (рис. 384) в отличие от метода итераций, где потребовалось 10 итераций в лучшем из представлений.

	A	B	C	D	E	F	G	H	I	J	K		
1			$f(x) = x^2 - \sin x - 1 = x^2 - (\sin x + 1)$										
2			$f(x) = 2x - \cos x$										
3													
4			Таблица значений										
5													
6			№итераций	x	A=x*x	sin x	B=(sin x) +1	f=A-B	D=2*x	C=cos x	f1=D-C	L=f/f1	x-L
7			0	1,0000000000	1,0000000000	0,8414709848	1,8414709848	-0,8414709848	2,0000000000	0,5403023059	1,4596976941	-0,5764693527	1,57
8			1	1,5764693527	2,4852556199	0,9999839084	1,9999839084	0,4852717114	3,1529387053	-0,0056729954	3,1586117007	0,1536344943	1,42
9			2	1,4228348583	2,0244590341	0,9890736576	1,9890736576	0,0353853765	2,8456697166	0,1474221823	2,6982475343	0,0131142069	1,40
10			3	1,4097206514	1,9873123150	0,9870553376	1,9870553376	0,0002569774	2,8194413028	0,1603800503	2,6590612525	0,0000966422	1,40
11			4	1,4096240092	1,9870398475	0,9870398335	1,9870398335	0,0000000139	2,8192480185	0,1604754407	2,6587725778	0,0000000052	1,40
12			5	1,4096240040	1,9870398327	0,9870398327	1,9870398327	0,0000000000	2,8192480080	0,1604754459	2,6587725621	0,0000000000	1,40
13			6	1,4096240040	1,9870398327	0,9870398327	1,9870398327	0,0000000000	2,8192480080	0,1604754459	2,6587725621	0,0000000000	1,40

Рис. 384. Таблица итераций получения последовательных значений для метода Ньютона (10 знаков после запятой)

2.3. Метод дихотомии (половинного деления)

Метод дихотомии (метод деления отрезка пополам) основан на известной **теореме Больцано-Коши**:

Если непрерывная на отрезке $[a, b]$ функция $f(x)$ на концах его имеет противоположные знаки, т.е. $f(a) \cdot f(b) < 0$, то на интервале (a, b) она хотя бы раз обращается в нуль.

Данная теорема не дает вопрос о количестве корней (он может быть как один, так и произвольное нечетное число) в случае выполнения данного условия и не позволяет утверждать, что корней точно нет, если условие не выполняется (их может быть произвольное четное число).

А вот если функция на отрезке является строго монотонной, то тогда можно утверждать следующее.

Если непрерывная и строго монотонная на отрезке $[a, b]$ функция $f(x)$ на концах его имеет противоположные знаки, т.е. $f(a) \cdot f(b) < 0$, то на интервале (a, b) имеется один и только один корень.

Метод дихотомии основан на последовательном делении отрезка локализации корня пополам.

Для этого выбирается начальное приближение к отрезку $[a, b]$, такое, что $f(a) \cdot f(b) < 0$, затем определяется знак функции в точке $c = (a+b)/2$ - середине отрезка $[a, b]$. Если он противоположен знаку функции в точке a , то корень локализован на отрезке $[a, c]$, если же нет – то на отрезке $[c, b]$.

Алгоритм можно записать так:

1. представить решаемое уравнение в виде $f(x)=0$;
2. выбрать такие a, b , что $c=(a+b)/2$;
3. вычислить $c=(a+b)/2$;
4. если $f(a)*f(c)<0$, то $b = c$ иначе $a = c$;
5. если критерий сходимости не выполнен, то перейти к п. 3;
6. напечатать корень из переменной c .

Для нашего примера $f(x)=x^2-\sin x-1$ постройте таблицу сначала для корня с точностью до 6 знаков.

	A	B	C	D	E	F	G	H	
1				Метод дихотомии					
3		Итераци: a		$f(a)=a*a-\sin a-1$	$c=(a+b)/2$	$f(c)=c*c-\sin c-1$	b	$f(b)=b*b-\sin b-1$	
4		0	1	-0,841471	1,5	0,252505	2	2,090703	
5		1	1	-0,841471	1,25	-0,386485	1,5	0,252505	
6		2	1,25	-0,386485	1,375	-0,090268	1,5	0,252505	
7		3	1,375	-0,090268	1,4375	0,075277	1,5	0,252505	
8		4	1,375	-0,090268	1,406250	-0,008954	1,4375	0,075277	
9		5	1,406250	-0,008954	1,421875	0,032797	1,4375	0,075277	
10		6	1,406250	-0,008954	1,414063	0,011830	1,421875	0,032797	
11		7	1,406250	-0,008954	1,410156	0,001416	1,414063	0,011830	
12		8	1,406250	-0,008954	1,408203	-0,003775	1,410156	0,001416	
13		9	1,408203	-0,003775	1,409180	-0,001181	1,410156	0,001416	
14		10	1,409180	-0,001181	1,409668	0,000117	1,410156	0,001416	
15		11	1,409180	-0,001181	1,409424	-0,000532	1,409668	0,000117	
16		12	1,409424	-0,000532	1,409546	-0,000208	1,409668	0,000117	
17		13	1,409546	-0,000532	1,409607	-0,000045	1,409668	0,000117	
18		14	1,409607	-0,000532	1,409637	0,000036	1,409668	0,000117	
19		15	1,409607	-0,000532	1,409622	-0,000005	1,409637	0,000117	

Рис. 385. Таблица итераций получения последовательных значений для метода дихотомии (6 знаков после запятой)

В этом случае достаточно 15 итераций $N=15$ (рис. 385).

А затем с точностью до 10 знаков после запятой.

	K	L	M	N	O	P	Q	
			Метод дихотомии					
		Итерация a	$f(a)=a*a-\sin a-1$	$c=(a+b)/2$	$f(c)=c*c-\sin c-1$	b	$f(b)=b*b-\sin b-1$	
	0	1,0000000000	-0,8414709848	1,5000000000	0,2525050134	2,0000000000	2,0907025732	
	1	1,0000000000	-0,8414709848	1,2500000000	-0,3864846194	1,5000000000	0,2525050134	
	2	1,2500000000	-0,3864846194	1,3750000000	-0,0902680570	1,5000000000	0,2525050134	
	3	1,3750000000	-0,0902680570	1,4375000000	0,0752770590	1,5000000000	0,2525050134	
	4	1,3750000000	-0,0902680570	1,4062500000	-0,0089537082	1,4375000000	0,0752770590	
	5	1,4062500000	-0,0089537082	1,4218750000	0,0327968180	1,4375000000	0,0752770590	
	6	1,4062500000	-0,0089537082	1,4140625000	0,0118303764	1,4218750000	0,0327968180	
	7	1,4062500000	-0,0089537082	1,4101562500	0,0014155442	1,4140625000	0,0118303764	
	8	1,4062500000	-0,0089537082	1,4082031250	-0,0037747789	1,4101562500	0,0014155442	
	9	1,4082031250	-0,0037747789	1,4091796875	-0,0011810417	1,4101562500	0,0014155442	
	10	1,4091796875	-0,0011810417	1,4096679688	0,0001168952	1,4101562500	0,0014155442	
	11	1,4091796875	-0,0011810417	1,4094238281	-0,0005321623	1,4096679688	0,0001168952	
	12	1,4094238281	-0,0005321623	1,4095458984	-0,0002076558	1,4096679688	0,0001168952	
	13	1,4095458984	-0,0005321623	1,4096069336	-0,0000453859	1,4096679688	0,0001168952	
	14	1,4096069336	-0,0005321623	1,4096374512	0,0000357532	1,4096679688	0,0001168952	
	15	1,4096069336	-0,0005321623	1,4096221924	-0,0000048167	1,4096374512	0,0001168952	
	16	1,4096221924	-0,0005321623	1,4096298218	0,0000154682	1,4096374512	0,0001168952	
	17	1,4096221924	-0,0005321623	1,4096260071	0,0000053257	1,4096298218	0,0001168952	

Рис. 386. Таблица итераций получения последовательных значений для метода дихотомии (10 знаков после запятой)

В этом случае число итераций равно 17 $N=17$ (рис. 386).

2.4. Наконец, рассмотрим метод хорд (метод секущих).

Алгебраическое описание метода секущих

Пусть a — абсциссы концов хорды, $y=f(x)$ — уравнение функции, решаемое методом секущих. Найдём коэффициенты и из системы уравнений

Вычтем из первого уравнения второе: затем найдём коэффициенты и

тогда

Уравнение принимает вид

Таким образом, теперь можем найти первое приближение к корню, полученное методом секущих (хорд):

Теперь возьмём координаты $(a, f(a))$ и $(b, f(b))$ и повторим все проделанные операции, найдя новое приближение к корню. Таким образом, итерационная формула метода секущих (хорд) имеет вид:

Повторять операцию следует до тех пор, пока не станет меньше или равно заданному значению погрешности.

2.4. Метод хорд с итерационной формулой

Первые три итерации метода хорд. Синим нарисована функция $f(x)$, красными проводятся хорды (рис. 387, 388).

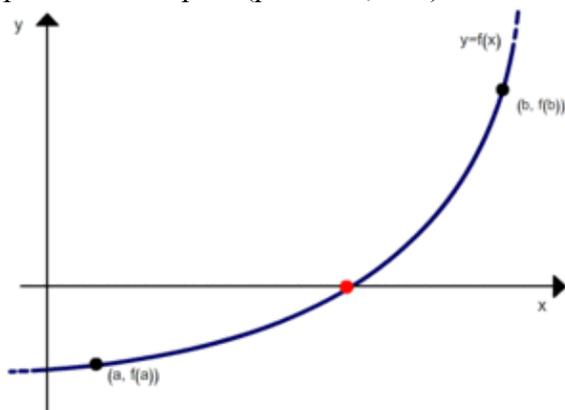


Рис. 387. Выбор начальных точек для метода хорд

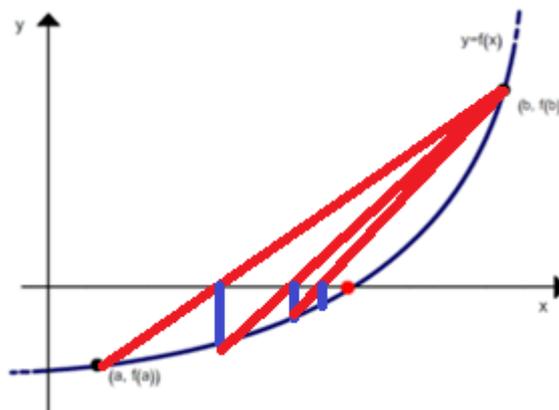


Рис. 388. Процесс последовательных приближений для метода хорд

Иногда методом секущих называют метод с итерационной формулой

Этот метод можно считать разновидностью метода простой итерации, и он имеет меньшую скорость сходимости. Далее для определённости этот метод будем называть методом хорд, а метод, описанный в теоретическом материале, методом секущих.

Для построения таблицы определим: $b=x_0=2$, $a=x_1=1$. Видно, что корень приближенного решения уравнения с точностью до 6 знаков после запятой вычисляется с помощью 11 итераций $N=11$ (рис. 389).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Метод хорд											
2						f(x0)			f(xi)	f(xi)*(xi-x0)	f(xi)-f(x0)		
3		Итерация	x(i)	x0	xi-x0	A=x0*x0-sin x0-1	sin xi	xi*xi	B=xi*xi-sin xi -1	C=B*(xi-x0)	D=B-A	E=-C/D	x(i+1)=xi+E
4		0	1	2	-1	2,090703	0,841471	1	-0,841471	0,841471	-2,932174	0,286979	1,286979
5		1	1,286979	2	-0,713021	2,090703	0,959993	1,656314	-0,303680	0,216530	-2,394382	0,090433	1,377411
6		2	1,377411	2	-0,622589	2,090703	0,981359	1,897261	-0,084098	0,052358	-2,174801	0,024075	1,401486
7		3	1,401486	2	-0,598514	2,090703	0,985701	1,964163	-0,021538	0,012891	-2,112240	0,006103	1,407589
8		4	1,407589	2	-0,592411	2,090703	0,986711	1,981307	-0,005404	0,003202	-2,096107	0,001527	1,409116
9		5	1,409116	2	-0,590884	2,090703	0,986958	1,985609	-0,001349	0,000797	-2,092052	0,000381	1,409497
10		6	1,409497	2	-0,590503	2,090703	0,987020	1,986683	-0,000336	0,000199	-2,091039	0,000095	1,409592
11		7	1,409592	2	-0,590408	2,090703	0,987035	1,986951	-0,000084	0,000049	-2,090786	0,000024	1,409616
12		8	1,409616	2	-0,590384	2,090703	0,987039	1,987018	-0,000021	0,000012	-2,090723	0,000006	1,409622
13		9	1,409622	2	-0,590378	2,090703	0,987040	1,987034	-0,000005	0,000003	-2,090708	0,000001	1,409624
14		10	1,409624	2	-0,590376	2,090703	0,987040	1,987038	-0,000001	0,000001	-2,090704	0,000000	1,409624
15		11	1,409624	2	-0,590376	2,090703	0,987040	1,987039	0,000000	0,000000	-2,090703	0,000000	1,409624
16													

Рис. 389. Таблица итераций получения последовательных значений для метода хорд (6 знаков после запятой)

Аналогично находим корень приближенного решения уравнения с точностью до 10 знаков после запятой, для него количество итераций равно **N=18** (рис. 390).

	A	B	C	D	E	F	G	H	I	J	K	L	N
1		Метод хорд											
2						f(x0)			f(xi)	f(xi)*(xi-x0)	f(xi)-f(x0)		
3		Итерация	x(i)	x0	xi-x0	A=x0*x0-sin x0-1	sin xi	xi*xi	B=xi*xi-sin xi -1	C=B*(xi-x0)	D=B-A	E=-C/D	x(i+1)=xi+E
4		0	1	2	-1	2,090702573	0,841470985	1	-0,841470985	0,841470985	-2,932173558	0,286978573	1,286979
5		1	1,2869785735	2	-0,7130214265	2,0907025732	0,9599933794	1,6563138485	-0,3036795308	0,2165300123	-2,3943821040	0,0904325220	1,377411
6		2	1,3774110955	2	-0,6225889045	2,0907025732	0,9813592785	1,8972613259	-0,0840979526	0,0523584522	-2,1748005258	0,0240750596	1,401486
7		3	1,4014861550	2	-0,5985138450	2,0907025732	0,9857012392	1,9641634427	-0,0215377964	0,0128906694	-2,1122403696	0,0061028421	1,407589
8		4	1,4075889971	2	-0,5924110029	2,0907025732	0,9867112205	1,9813067847	-0,0054044358	0,0032016472	-2,0961070089	0,0015272425	1,409116
9		5	1,4091164225	2	-0,5908835775	2,0907025732	0,9869582512	1,9856090923	-0,0013491589	0,0007971958	-2,0920517321	0,0003810593	1,409497
10		6	1,4094974819	2	-0,5905025181	2,0907025732	0,9870195211	1,9866831514	-0,0003363697	0,0001986271	-2,0910389429	0,0000949897	1,409592
11		7	1,4095924716	2	-0,5904075284	2,0907025732	0,9870347720	1,9869509359	-0,0000838361	0,0000494975	-2,0907864093	0,0000236741	1,409616
12		8	1,4096161457	2	-0,5903838543	2,0907025732	0,9870385716	1,9870176781	-0,0000208935	0,0000123352	-2,0907234667	0,0000059000	1,409622
13		9	1,4096220456	2	-0,5903779544	2,0907025732	0,9870395184	1,9870343115	-0,0000052069	0,0000030741	-2,0907077801	0,0000014703	1,409624
14		10	1,4096235159	2	-0,5903764841	2,0907025732	0,9870397543	1,9870384567	-0,0000012976	0,0000007661	-2,0907038708	0,0000003664	1,409624
15		11	1,4096238824	2	-0,5903761176	2,0907025732	0,9870398131	1,9870394898	-0,0000003234	0,0000001909	-2,0907028966	0,0000000913	1,409624
16		12	1,4096239737	2	-0,5903760263	2,0907025732	0,9870398278	1,9870397472	-0,0000000806	0,0000000476	-2,0907026538	0,0000000228	1,409624
17		13	1,4096239964	2	-0,5903760036	2,0907025732	0,9870398314	1,9870398114	-0,0000000201	0,0000000119	-2,0907025933	0,0000000057	1,409624
18		14	1,4096240021	2	-0,5903759979	2,0907025732	0,9870398324	1,9870398274	-0,0000000050	0,0000000030	-2,0907025782	0,0000000014	1,409624
19		15	1,4096240035	2	-0,5903759965	2,0907025732	0,9870398326	1,9870398313	-0,0000000012	0,0000000007	-2,0907025744	0,0000000004	1,409624
20		16	1,4096240039	2	-0,5903759961	2,0907025732	0,9870398326	1,9870398323	-0,0000000003	0,0000000002	-2,0907025735	0,0000000001	1,409624
21		17	1,4096240040	2	-0,5903759960	2,0907025732	0,9870398327	1,9870398326	-0,0000000001	0,0000000000	-2,0907025733	0,0000000000	1,409624
22		18	1,4096240040	2	-0,5903759960	2,0907025732	0,9870398327	1,9870398326	0,0000000000	0,0000000000	-2,0907025732	0,0000000000	1,409624

Рис. 390. Таблица итераций получения последовательных значений для метода хорд (10 знаков после запятой)

Практическая работа № 16. Работа с векторами и матрицами Некоторые матричные операции в MS Excel

В MS Excel с матрицами можно работать как с диапазоном ячеек. *Диапазон* – это совокупность смежных ячеек, образующих прямоугольную область таблицы, заданную адресами левой верхней и нижней правой ячеек области. При указании диапазона принята форма записи, в которой эти адреса указываются через двоеточие.

Например,

B2:D4 – это диапазон из девяти ячеек **B2, B3, B4, C2, C3, C4, D2, D3, D4** (матрица размера 3x3);

B2:B5 – это диапазон из четырех ячеек **B2, B3, B4, B5** (вектор- столбец);

B2:E2 – это диапазон из четырех ячеек **B2, C2, D2, E2** (вектор-строка).

В MS Excel возможно использование *формул*, результатом которых является не одно число, а диапазон чисел – это так называемые *формулы массива*.

Для ввода таких формул необходимо выполнить следующие действия:

- Выделить ячейки, в которые должен быть помещен *результат выполнения формулы*.
- Ввести формулу массива (начиная со знака "=")
- Нажать комбинацию клавиш **Ctrl+Shift+Enter** для завершения ввода (для этого нажимаются и удерживаются клавиши **Ctrl+Shift** и затем кратковременно нажимается клавиша **Enter**).

Введенные формулы массива отображаются в строке формул в *фигурных скобках*.

Для изменения (или удаления) введенной ранее формулы массива необходимо *предварительно выделить весь диапазон ячеек* этой формулы, а затем выполнить желаемое действие. Часть ячеек формулы массива *изменить нельзя!*

Операции с векторами и матрицами.

1. Результат сложения/вычитания) матриц/векторов одинакового размера $n \times m$ A и B (число столбцов и строк матриц должны совпадать) есть матрица C размера $n \times m$, каждый элемент которой равен сумме (или разности) соответствующих элементов матриц A и B : $c_{ij} = a_{ij} \pm b_{ij}$. Ниже приведен рабочий лист сложения двух векторов-столбцов, заданных в ячейках **A2:A5** и **C2:C5**, суммарный вектор получается в ячейках **E2:E5** (рис. 391).

	A	B	C	D	E	F
1						
2	12		3		15	
3	11,5		2		13,5	
4	5		6		11	
5	2		3		5	
6						
7						

Рис. 391. Сложение векторов

	A	B	C	D	E
1					
2	12	3		95	
3	11,5	2		95	
4	5	6		95	
5	2	3		95	
6					

Рис. 392. Скалярное произведение векторов

Скалярное произведение двух векторов.

Скалярным произведением двух векторов одинаковой длины n называется сумма парных произведений соответствующих компонентов вектора.

$$\text{Для этой } (a_1 \ a_2 \ \dots \ a_n) \cdot (b_1 \ b_2 \ \dots \ b_n) = \sum_{i=1}^n a_i \cdot b_i$$

операции можно использовать встроенную функцию **СУММПРОИЗВ**. У этой функции два параметра, отделяемые точкой запятой. Так как результат вычислений – это одно число, то формула вводится в одну ячейку (рис. 392).

Матричное произведение

Произведением матриц A размером $n \times m$ и B размером $m \times l$ называется матрица C размером $n \times l$ такая, что элемент, стоящий на пересечении i -ой строки и j -го столбца c_{ij} равен скалярному произведению i -ой строки матрицы A и j -ого столбца матрицы B :

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$$

На рисунке 395 приведено матричное умножение матрицы A размера 3×3 (рис. 393) на вектор-столбец B (или матрица размера 3×1 , рис. 394). Результат – вектор-столбец C (или матрица размером 3×1 , рис 395).

$$\begin{pmatrix} 1 & 2 & 5 \\ 3 & 1 & 5 \\ 1 & 5 & -3 \end{pmatrix}$$

$$\begin{pmatrix} 5 \\ 4 \\ 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 5 \\ 3 & 1 & 5 \\ 1 & 5 & -3 \end{pmatrix} \begin{pmatrix} 5 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 28 \\ 24 \\ 16 \end{pmatrix}$$

Рис. 393. Матрица **A** Рис. 394. Вектор **B** Рис. 395. Произведение матрицы **A** на вектор **B**

В MS Excel для матричного умножения используется встроенная функция **МУМНОЖ**. У нее два параметра, соответствующие двум диапазонам, содержащим перемножаемые матрицы. Результат функции - это матрица, поэтому вводиться она должна в диапазон ячеек как функция массива (рис. 396).

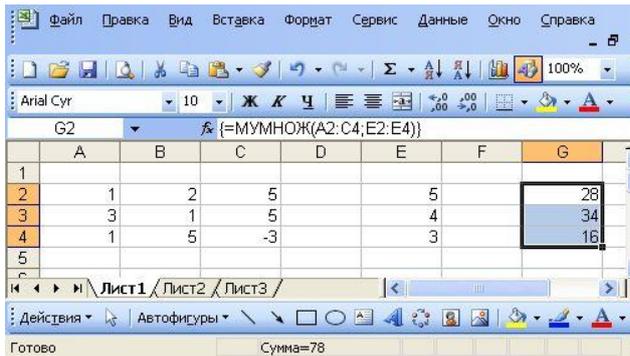


Рис. 396. Умножение матриц (матрицы на вектор)

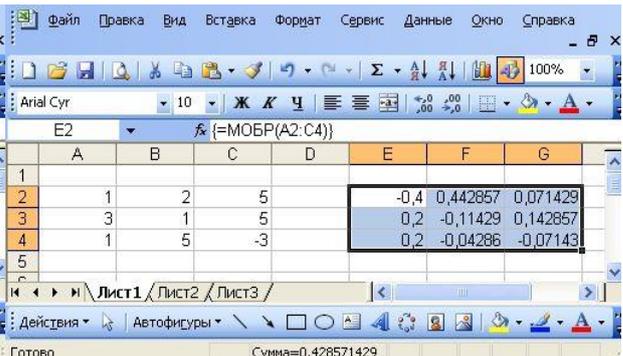


Рис. 397. Вычисление обратной матрицы

Обращение матрицы

Матрицей, обратной матрице **A** размера $(n \times n)$ называется такая матрица A^{-1} размера $(n \times n)$, что при перемножении этих матриц в любом порядке получается **E** — единичная диагональная матрица: $A * A^{-1} = A^{-1} * A = E$;

здесь **E** — это единичная диагональная матрица размера $(n \times n)$ — все элементы которой равны **0**, за исключением диагональных, которые равны 1.

Нахождение обратной матрицы выполняет встроенная функция **МОБР**. У нее единственный аргумент, который является квадратным диапазоном, содержащим обращаемую матрицу. Функция возвращает матрицу, равную по размеру обращаемой матрице, поэтому должна вводиться как функция массива (рис. 397).

Определитель матрицы

Нахождение определителя матрицы выполняет встроенная функция **МОПРЕД**. У нее единственный параметр — это диапазон, содержащий матрицу, определитель которой надо найти. Например, формула **=МОПРЕД(A2:D5)** вычисляет определитель матрицы размера 4×4 , записанной в диапазоне **A2:D5**, результат помещается в ячейку **G2**, определенную до применения формулы (рис. 398)

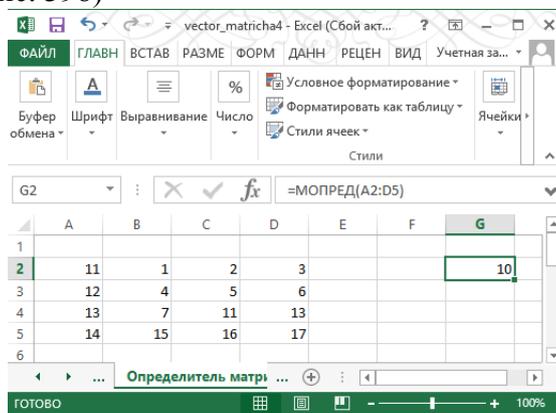


Рис. 398. Нахождение определителя матрицы

Дополнительные задания

Практическое занятие № 17. Составление кроссворда

Работа в Excel и Word

1. Прежде чем работать с приложениями MS, продумайте структуру, и нарисуйте кроссворд на бумаге (в клеточку), придумайте определения для слов, выбранных для кроссворда.

2. Запустите приложение электронные таблицы MS Excel. Оформите шаблон кроссворда. Выделите мышью необходимое для кроссворда количество столбцов и строк. Для выделенного диапазона выберите пункт **Формат** на вкладке **Главная** и задайте

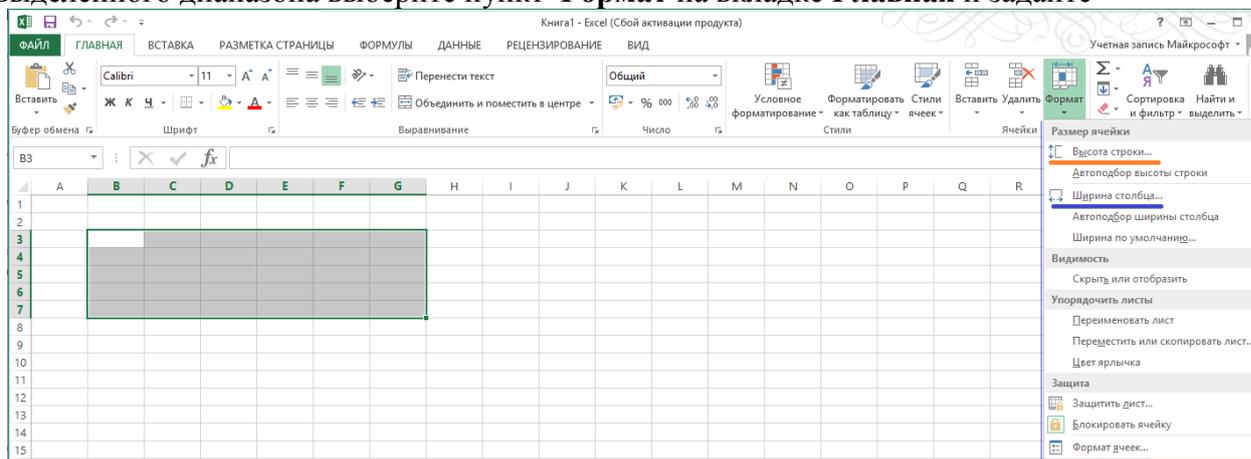


Рис. 399. Выделение диапазона ячеек, расположение пункта **Формат** и меню **Формат** (Высота строки и Ширина столбца)

- ширину столбца – 3 (**Формат** —> **Ширина столбца**, рис. 400),
- высоту строки – 18 (**Формат**->**Высота строки**, рис. 401).

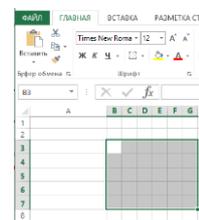
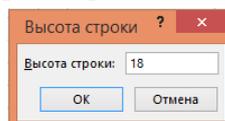
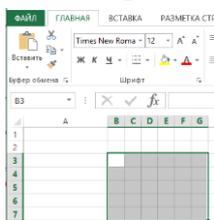
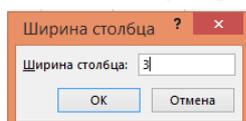


Рис. 400. Установка ширины столбца и вид диапазона

Рис. 401. Установка высоты строки и вид диапазона

Оформите вид кроссворда. Для этого выделите необходимые ячейки строки для слова кроссворда (рис. 402), удерживая левую кнопку мыши, и задайте границы ячеек, выбрав на вкладке **Главная** пункт **Границы** и пункт меню **Все границы** (рис. 403).

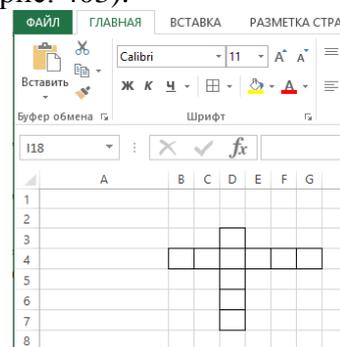
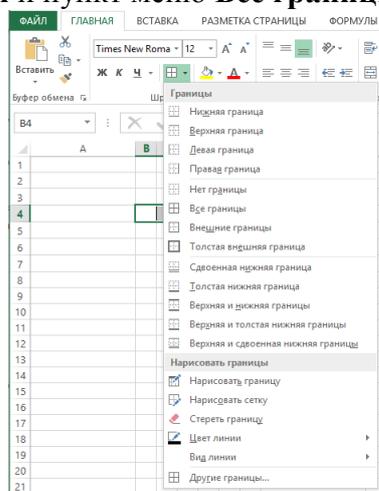
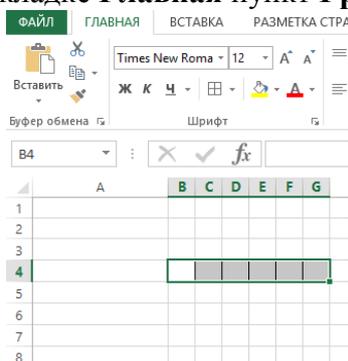


Рис. 402. Выделение ячеек строки в кроссворде

Рис. 403. Выбор пункта меню **Все границы** блока **Формат** вкладки **Главная**

Рис. 404. Выделение ячеек столбца в кроссворде

Далее выделите необходимые ячейки столбца для слова кроссворда (рис. 404), удерживая левую кнопку мыши, и задайте границы ячеек, выбрав на вкладке **Главная** пункт **Границы** и пункт меню **Все границы** (рис. 403).

Выполните эти операции для всех строк и всех столбцов кроссворда (рис. 405).

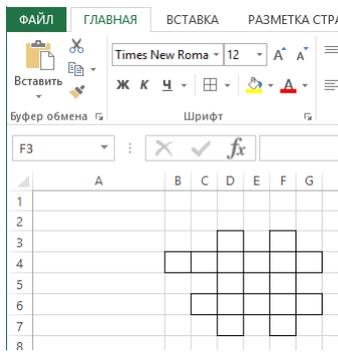


Рис. 405. Вид заготовки кроссворда

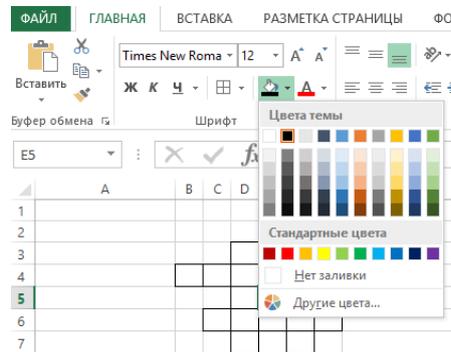


Рис. 406. Выбор цвета заливки

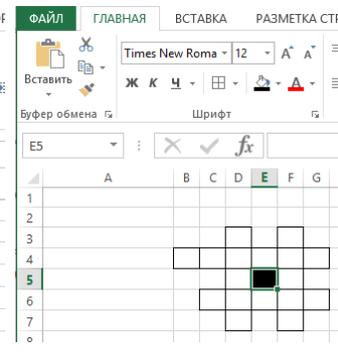


Рис. 407. Заливка ячейки черным цветом

Выполните заливку ячеек, если это необходимо. Для этого выделите требуемые ячейки и на вкладке **Главная** выберите пункт **Цвет заливки** и в раскрывшемся меню - черный цвет (рис. 406). Результат этих действий представлен на рисунке 407.

Расставьте цифры в клетках кроссворда согласно разработанной структуре (рис. 408).

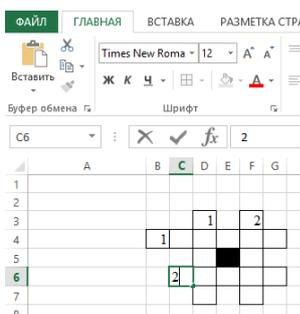


Рис. 408. Заливка ячейки черным цветом



Рис. 409. Выделения ячеек для форматирования текста



Рис. 410. Результат оформления шаблона кроссворда



Рис. 411. Маркер блока **Шрифт**

Оформите расположение и шрифт буквы в ячейке. Выделите ячейки с цифрами, удерживая кнопку **CTRL**. Выделив требуемые ячейки, отпустите **CTRL**, щелкните левой кнопкой мыши на главной панели в правом углу блока **Шрифт** по маркеру (рис. 411).

Откроется меню **Формат ячеек**. На вкладке **Выравнивание** задайте выравнивание по горизонтали **по левому краю** и по вертикали **по верхнему краю**. На вкладке **Шрифт** установите размер шрифта – 8.

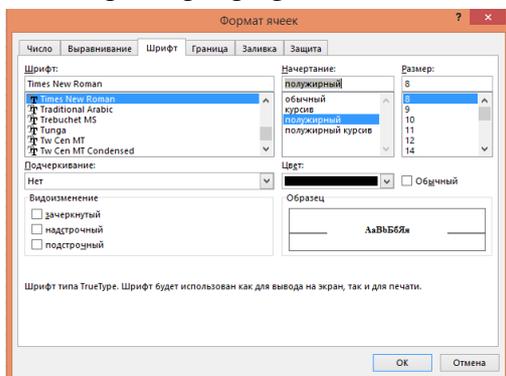


Рис. 412. Установка параметров шрифта

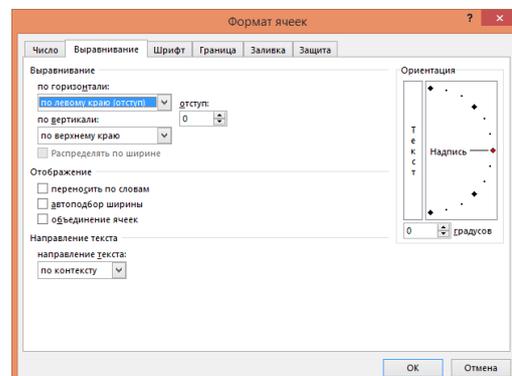


Рис. 413 Установка параметров выравнивания

Сверните (нажмите на значок , а не) электронную таблицу MS Excel.

3. Откройте MS Word.

3.1. Наберите определения к словам, заготовленного на бумаге кроссворда, и оформите для этих определений многоуровневый список.

Для первого уровня используйте стиль **Заголовок 1**, для второго уровня создайте

Стиль: абзаца

Имя: Стиль **Заголовок 2**,

Основанный на стиле: Обычный,

Шрифт Times New Roman, размер 12.

3.2. Выделите весь список целиком.

Выполните **Многоуровневый** —> **Определить новый многоуровневый список**.

Нажмите кнопку **Больше**.

Выберите

Уровень 1.

Связать уровень со стилем: **Заголовок 1**.

Нумерация для этого уровня: нет.

3.3. Выбрать

Уровень 2.

Связать уровень со стилем: **Стиль Заголовок 2**.

Нумерация для этого уровня: 1, 2, 3...

Шрифт — **Размер 12**, **Начертание** полужирное.

3.4. Наберите информацию, подготовленную для кроссворда, используя созданные стили.

По горизонтали:

Государство в Азии, расположенное на острове Калимантан.

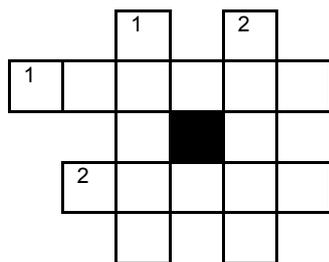
Государство на побережье Персидского залива.

По вертикали:

Государство, имеющее границы только с Китаем и Индией.

Государство Азии со столицей Катманду.

4. Развернуть электронные таблицы MS Excel. Выделить кроссворд. Скопировать его, перейти в MS Word и вставить кроссворд.



5. Добавить подсказки. Выделить цифру в сетке кроссворда и выполнить **Создать примечание** на вкладке **Рецензирование**, ввести подсказку.

Спрятать подсказки. Для чего выполнить **Показать исправления** и убрать галочку в пункте **Примечания**.

6. Сохранить документ под именем **Crossword.doc** в своей папке.

Практическое занятие № 18. Гиперссылки и закладки

Связывание документов гиперссылками и добавление в документ закладок

1. Запустите текстовый процессор MS Word.

2. Создайте в своей папке папку **Викторина**.

3. Создайте десять файлов и сохраните их в папке **Викторина**.

Файл 1 содержит, например, слово *Викторина* и *рисунок/эмблему*.

Файл 2: Хотите сыграть? [Да Нет](#).

Файл 3: *Вопрос 1*. На 100 рублей. Какая страна находится в Африке? Можно использовать файлы с загадками и создать более, чем один вопрос.

- [Бруней](#)
- [Бурунди](#)
- [Белиз](#)
- [Бутан](#)

Подсказки:

- [Мнение зала](#)
- [50:50](#)
- [Звонок другу](#)

Файл 4: А жаль! Ну, хоть познакомьтесь с [автором](#).

Файл 5: Автор: *Фамилия Имя Отчество*. Возможно использование файлов *visitka2.doc* или *nickname.doc*.

Файл 6: Ответ неверный. Ваш выигрыш 0 рублей. До свидания! [Познакомьтесь с автором!](#)

Файл 7: Это правильный ответ. Ваш выигрыш 100 рублей. Поздравляем! До свидания!
[Познакомьтесь с автором!](#)

Файл 8: Мнения в зале распределились следующим образом:

- 23%
- 37%
- 18%
- 22%

[Возврат к вопросу](#).

Файл 9: Остались два ответа:

- [Бурунди](#)
- [Белиз](#)

[Возврат к вопросу](#).

Файл 10: Ваш друг думает, что правильный ответ:

- [Бурунди](#)

[Возврат к вопросу](#).

4. Установите соответствующие гиперссылки. Сохраните все созданные документы в папке **Викторина**. Затем в тексте соответствующего документа выделите ключевое слово или объект (например, рисунок) и установите гиперссылку (**Вставка** → **Гиперссылка**). В появившемся диалоговом окне укажите имя файла, на который устанавливается ссылка.

5. Проведите тестирование.

6. **Добавление закладок**. Загрузите файл zagadki.doc.

Добавьте закладки к отгадкам (**Вставка** — **Закладка**). Задайте соответствующее имя закладке (в имени не должно быть пробелов). Нажмите **Добавить**.

Вернитесь в начало документа. Перейдите к первой закладке (**Вставка**→**Закладка**→**Имя закладки**→**Перейти**).

Попробуйте различные способы сортировки закладок (**Вставка**→**Закладка**→**Сортировать по**). Перейдите ко второй закладке.

Сохраните документ под именем **zakladka.doc** в своей папке.

Практическое занятие № 19. Таблицы в Word

Работа с таблицами

1. Запустите текстовый процессор MS Word.
2. **Форматирование таблиц.** Вставьте, заполните и оформите таблицу следующего вида:

Овощи, плоды и ягоды	Состав съедобной части, %			Калорийность, ккал / 100 г	Кислотность, %
	вода	углеводы	белки		
Картофель	78,0	21,0	2,0	91,0	5,5
Капуста белокочанная	90,0	5,0	1,8	29,1	6.1
Капуста цветная	91,0	5,0	2,8	30,5	5,8
Свекла столовая	86,5	10,0	1,5	48,0	5,2
Морковь	88,0	8,7	1,3	41,0	5,1
				239,6	

Рис. 414. Исходный вид таблицы

Создайте таблицу из 7 строк и 6 столбцов (**Вставка**→**Таблица**→**Вставить таблицу...**)(рис. 415).

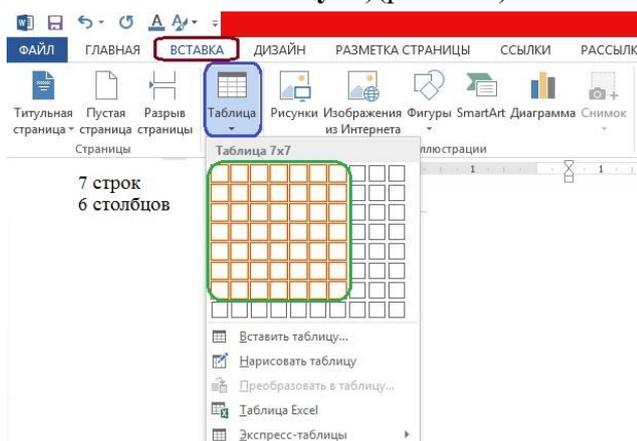


Рис. 415. Создание таблицы из 7 строк и 6 столбцов

Рис. 416. Вид созданной таблицы, подготовка к объединению трех ячеек

Выделите вторую, третью и четвертую ячейки в первой строке (рис. 416), установите курсор в выделенных ячейках, щелкните правой кнопкой мыши и выберите пункт меню **Объединить ячейки** (рис. 417).

Настройте ширину столбцов, используя мышь. Задайте выравнивание по центру или по левому краю для ячеек таблицы.

Используйте кнопку **Заливка** для красочного оформления таблицы.

Используйте кнопку **Внешние границы** и уберите ненужные границы.

Вставьте пустую строку после последней строки таблицы.

Заполните таблицу заголовков по горизонтали (первая и вторая строка) и вертикали (первый) столбец.

Заполните данными 2, 3, 4, 5 и 6 столбцы (рис. 418).

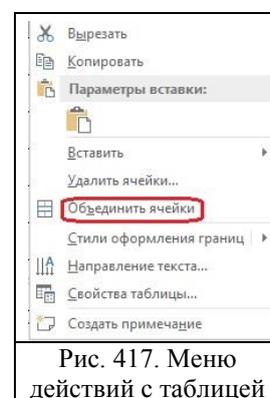


Рис. 417. Меню действий с таблицей

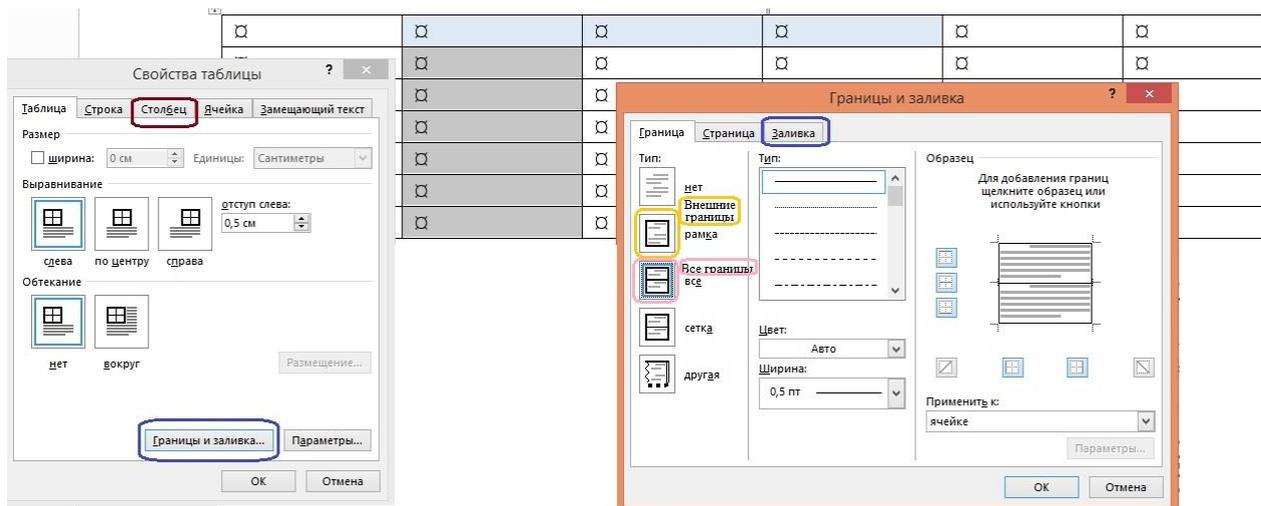


Рис. 418. Выпадающее меню **Свойства таблицы** и **Границы и заливка** для оформления таблицы

Есть специальная возможность работы с таблицами:

Вставка->Таблица->Вставить таблицу. Появляется окошко **Вставка таблицы**, в которой можно указать количество столбцов и строк таблицы. После появления таблицы появляется панель **Работа с таблицами**, с помощью которой можно выполнять все действия, которые выполняли выше (рис. 419).

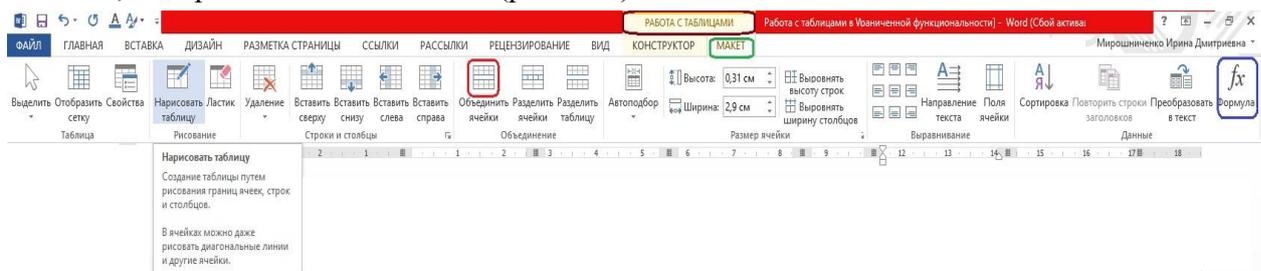


Рис. 419. Вставка **Работа с таблицами** и соответствующая этой вкладке панель с пунктами **Нарисовать таблицу**, **Объединить ячейки**, **Поля, Ячейки**, **Сортировка** и прочие действия с таблицами

Рассмотрите панель **Работы с таблицей**, в частности, поэкспериментируйте с применением пункта создания таблицы (рис. 420, 422) и использования пункта **Границы и заливки** (рис. 421).

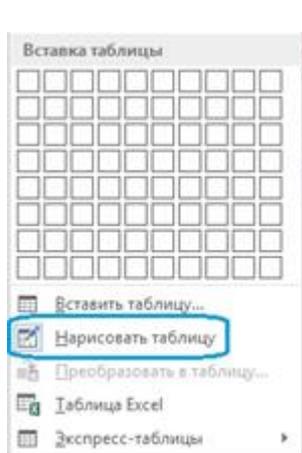


Рис.420. Нарисовать таблицу

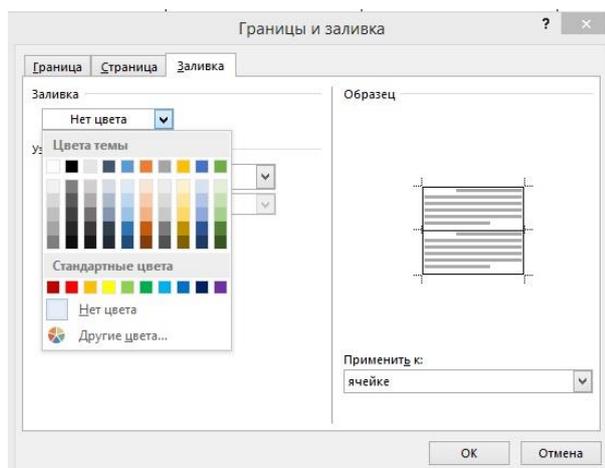


Рис. 421. Способы раскраски границ и заливки ячеек

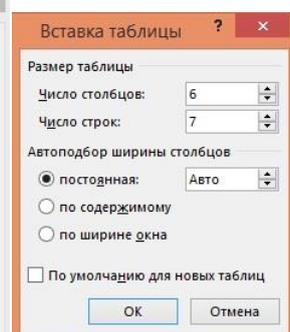


Рис. 422. Создание таблицы

Отсортируйте строки таблицы **по столбцу** «Овощи, плоды и ягоды»; **по столбцу** «Кислотность, %» (**Таблица — Сортировка**).

Вставьте после строки с цветной капустой следующую строку:

Хрен огородный	66,6	8,7	4,5	50,0	6,6
----------------	------	-----	-----	------	-----

Отсортируйте строки таблицы по столбцу «Овощи, плоды и ягоды» еще раз. Слева от столбца с кислотность вставьте столбец «Содержание железа в %».

Но, кроме того, на панели есть инструменты, расширяющие возможности работы с таблицами (рис. 424, 425). Одна из таких возможностей — ввод формул с помощью кнопки **Формула** (рис. 423).

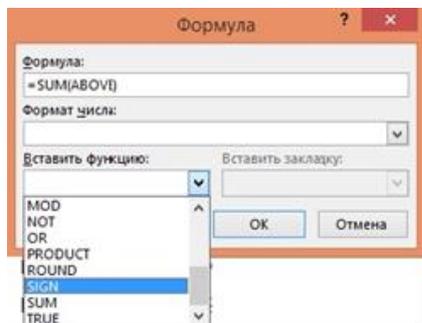


Рис. 423. Инструмент **Формула** на панели **Работа с таблицей**

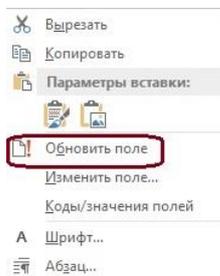


Рис. 424. Параметры вставки объекта

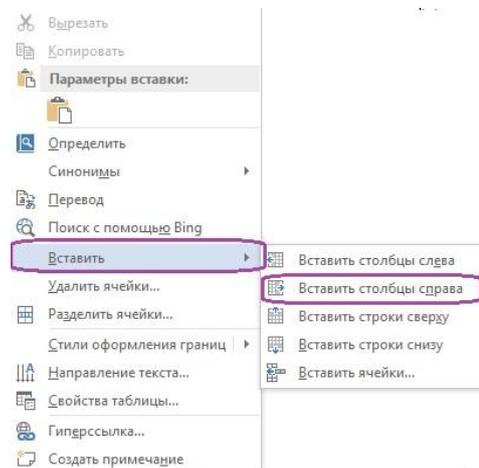


Рис. 425. Действия с таблицей

Подсчитайте итоговую сумму под 5-м столбцом «Калорийность»: используя кнопку **Формула**. Нажмите кнопку **Формула**, появится окно **Формула** (рис. 423). Установите курсор в итоговой ячейке, выберите в строке **Вставить функцию** требуемую формул из стандартных, в данном случае формулу **=SUM(ABOVE)**. В итоговой ячейке появится сумма всех ячеек в столбце.

Обратите внимание, что, если изменить какую-либо цифру в столбце «Калорийность», сумма автоматически не пересчитывается. Для ее пересчета надо, сначала выделить ячейку результата, затем щелкнуть по ней правой кнопкой мыши и выбрать пункт **Обновить поле** (рис. 424).

Слева от последнего столбца вставьте пустой столбец, используя пункт меню окна таблицы **Вставка таблицы**. Нарисуйте, соответственно, слева от таблицы ячейки таблицы

Справа от первого столбца вставьте пустой столбец, используя **Добавить столбец справа** (рис. 425).

Для этого подведите мышку к границе, разделяющей первый и второй столбцы. Рядом с границей появится плюсики в кружочке, нажмите его, появится справа столбец с шириной, равной ширине правого столбца.

Второй способ добавления столбца справа состоит в том, что курсор ставится в тот столбец, который будет левым в результате добавления, щелкните правой кнопкой мыши, появится контекстное меню, в котором выберите пункт меню **Вставить->Вставить столбцы справа** (рис. 425).

Сохраните документ под именем *table1.doc* в своей папке.

2. Изготовление визитки (2 способ).

Настройте параметры страницы так, чтобы ширина рабочего поля была 18 см.

Вставьте таблицу — 5 строк и 2 столбца (**Таблица** —> **Добавить** —> **Таблица**). Настройте высоту строк по 5 см (**Таблица** —> **Свойства таблицы**).

Занесите в первую ячейку данные о себе.

Выполните следующее оформление:

название факультета — размер **14, полужирный**, по центру;

фамилию, имя, отчество — размер 12, курсив;
 название группы — размер 10, полужирный по центру;
 адрес, телефон, электронная почта — размер 10, по правому краю.
 Значки конверта и телефона — в гарнитуре **Wingdings** соответственно **Shift+8** и **Shift+9**.
 После оформления скопируйте ячейку таблицы с визиткой в буфер обмена и вставьте ее
 в остальные девять ячеек.

Сохраните документ под именем *vizitka2.doc* в своей папке.

3. Оформление бланков.

Вставьте, заполните и оформите таблицы следующего вида:

 ФИЗКУЛЬТУРНЫЙ КОЛЛЕДЖ ГРУППА № _____ «__» _____ 202_ г.		<i>Преподавателю</i> <i>по информатике</i> <i>Мирошниченко Ирина Дмитриевна</i>
---	--	---

Заявка

Прошу обучить меня работе с новым приложением.

Гарантирую выполнение всех заданий.

Студент	<i>Группа 111</i>	<i>Иванов Олег Николаевич</i>
----------------	-------------------	-------------------------------

12 марта 2003 г.

Для оформления шапки и подписи используйте таблицы из трех столбцов. Границы таблиц можно убрать, используя **Таблица** —> **Скрыть сетку** (или другим способом, используя меню **Свойства таблицы->Границы->без границ**).

Для оформления таблицы можно использовать **Таблица** —> **Свойства таблицы** (задав конкретную ширину столбцов).

Символ  соответствует букве «ц» (русская раскладка), если выбран шрифт **Webdings**, размер **48**. Для других надписей используйте **Выравнивание ячеек**, гарнитуры Arial и Times New Roman, размер 14, полужирное и курсивное полужирное начертание.

Сохраните таблицу в файле *tables.doc*.

4. **Автоформат**. Загрузите файл *table1.doc*. Посмотрите возможные варианты, используя **Таблица** —> **Автоформат**. Выберите понравившийся формат и примените его. Воспользуйтесь **Таблица** —> **Автоподбор** —> **Автоподбор по содержимому**.

Сохраните документ под именем *table2.doc* в своей папке.

5. Использование формул.

Ячейкам таблицы назначаются номера:

- строки нумеруются натуральными числами,
- столбцы — латинскими буквами.

Загрузите файл *table2.doc*. Вставьте еще четыре строки после данных. Вычислите суммарное и среднее количество калорий и т.д., а также минимальное и максимальное значение по столбцам. Например, **Таблица** —> **Формула** —> **Sum(B3:B7)**, где **B3:B7** — последовательные ячейки **B3, B4, B5, B6, B7**

Сохраните документ под именем *table3.doc* в своей папке.

Список литературы

1. Практикум по информатике / Т. И. Немцова, Ю. В. Назарова. Ч. 1: Базовая компьютерная подготовка / ред. Л. Г. Гагарина - М.: Форум; М.: Инфра-М, 2011, 320 с.
2. Практикум по информатике / Т. И. Немцова, Ю. В. Назарова. Компьютерная графика и WEB-дизайн / ред. Л. Г. Гагарина - М.: Форум; М.: Инфра-М, 2020, 288 с.
3. Информатика. Задачник-практикум / Л. А. Залогова. - М. : БИНОМ. Лаборатория знаний, 2003
3. Задачник-практикум по информатике / Сафонов, Игорь Константинович. - СПб.: БХВ-Петербург, 2002. - 432с.

Ссылки

1. <https://support.microsoft.com/ru-ru>
2. <https://support.microsoft.com/ru-ru/word>
3. <https://support.microsoft.com/ru-ru/excel>

Оглавление

Информация и данные. Двоичное кодирование информации. Системы счисления	3
Информация.....	3
Кодирование информации.....	3
Системы счисления (способ кодирования чисел).....	3
Задание для самостоятельной работы.	6
Двоичное кодирование целых чисел	6
Перевод целых чисел из одной системы счисления в другую.....	7
Перевод дробных чисел из одной системы счисления в другую.	9
Сложение чисел в системах счисления.	11
Задания 1-3 для самостоятельной работы.....	12
Кодирование числовых и текстовых данных	13
Кодирование числовых данных	13
Кодирование текстовых (символьных) данных	16
Задание 4 (по системам счисления и кодированию).....	20
Кодирование графических изображений	21
Растровое кодирование.....	21
Кодирование цвета	22
Векторное кодирование	28
Задание 5 (по теме кодирование графической информации)	30
Кодирование звуковой и видеоинформации. Оцифровка звука.....	31
Инструментальное кодирование звука.....	33
Кодирование видеоинформации	34
Задание 6. (по теме кодирование видео информации)	35
Приемы и методы работы со сжатыми данными Алгоритм RLE. Алгоритм Хаффмана.....	36
Теоретические основы сжатия данных	36
Алгоритм RLE	37

Кодирование графического изображения	38
Алгоритм Хаффмана	39
Задание 7 для самостоятельной работы	43
Практическое занятие №1. Калькулятор	43
Приемы работы с инструментом Калькулятор	43
Контрольные вопросы для самопроверки	51
Задания 8 (<i>n</i> – номер студента по списку в журнале преподавателя)	51
Практическое занятие № 2. Paint	51
Приемы работы с графическим редактором Paint	51
Практическое занятие № 3. Блокнот	55
Приемы работы с текстовым редактором Блокнот	55
Практическое занятие № 4. WordPad	59
Приемы форматирования, внедрения и связывания объектов в текстовом процессоре WordPad	59
Практическое занятие № 5. MS Word	63
Создание и сохранение документов в текстовом процессоре MS Word	63
Практическое занятие № 6. Форматирование в MS Word	65
Шрифтовое и абзацное форматирование в текстовом процессоре MS Word	65
Практическое занятие № 7. Буквица, колонки	69
Дополнительные возможности форматирования и набора текста	69
Практическое занятие № 8. Списки	72
Word. Оформление списков	72
2. Оформление маркированным списком.	73
3. Оформление нумерованным списком.	74
4. Оформление многоуровневым списком (создание многоуровневого списка).	76
Практическое занятие № 9. Стили	81
Практическое занятие № 10. Шаблон	91
Стили и шаблоны (комплексная работа – проверка навыков)	91
Практическое занятие № 11. Графика в Word	94
Графические возможности	94
Практическое занятие № 12. Колонтитулы	101
Колонтитулы и дополнительные графические возможности	101
Практическое занятие № 13. MS Excel.	107
Знакомство с адресацией в MS Excel	107
Практическое занятие № 14. Поиск решения	110
Задачи линейного программирования и функция «Поиск решения»	110
Принципы решения задач оптимизации (управленческих и экономических задач)	112

I. Транспортная задача.....	113
II. Решение задачи о штате фирмы.....	117
III. Задача о производстве красок.....	122
IV. Расчет общей суммы скидок различных товаров.....	124
Практическая работа № 15. Решение уравнений в Excel.....	132
Приближенное решение уравнений (график и 4 метода).....	132
2.1. Метод простой Итерации.....	139
2.2. Метод Ньютона (касательных).....	144
2.3. Метод дихотомии (половинного деления).....	145
2.4. Метод хорд с итерационной формулой.....	147
Практическая работа № 16. Работа с векторами и матрицами.....	148
Некоторые матричные операции в MS Excel.....	148
Операции с векторами и матрицами.....	149
Скалярное произведение двух векторов.....	149
Обращение матрицы.....	150
Определитель матрицы.....	150
Дополнительные задания.....	151
Практическое занятие № 17. Составление кроссворда.....	151
Работа в Excel и Word.....	151
Практическое занятие № 18. Гиперссылки и закладки.....	154
Связывание документов гиперссылками и добавление в документ закладок.....	154
Практическое занятие № 19. Таблицы в Word.....	155
Работа с таблицами.....	155
Список литературы.....	159
Оглавление.....	159