

Санкт-Петербургский государственный университет  
Кафедра математической теории игр и статистических решений

# Выпускная квалификационная работа бакалавра

Рязанова Дарья Денисовна

«АВТОМАТИЗАЦИЯ ЛОГИСТИЧЕСКИХ ЗАДАЧ НА  
ПРОИЗВОДСТВЕ»

Направление: 01.03.02

Прикладная математика, фундаментальная информатика и основы  
программирования

Заведующий кафедрой  
д.ф.-м.н., профессор Петросян Л.А.

Научный руководитель  
д.т.н., профессор Буре В.М.

Санкт-Петербург  
2018г.

## Содержание

1	Введение . . . . .	4
2	Цель работы . . . . .	5
3	Обзор литературы . . . . .	5
4	Введение в предметную область . . . . .	6
4.1	Массовое производство . . . . .	8
4.2	Kanban, Just in Time . . . . .	9
4.3	Основные обозначения . . . . .	11
5	Информационные системы . . . . .	12
5.1	Система для автоматизации построения плана поставок и раз- грузок автокомпонентов на производство . . . . .	12
5.1.1	Постановка задачи . . . . .	12
5.1.2	Описание алгоритма . . . . .	13
5.1.3	Разработка системы . . . . .	15
5.1.4	Схема устройства базы данных . . . . .	16
5.1.5	Изменения в производственном процессе, полученные после внедрения системы . . . . .	18
5.2	Приложение для работы с базой данных излишек . . . . .	19
5.2.1	Постановка задачи . . . . .	19
5.2.2	Описание алгоритма . . . . .	19
5.2.3	Разработка системы . . . . .	21
5.2.4	Схема устройства базы данных . . . . .	21
5.2.5	Изменения в производственном процессе, полученные после внедрения системы . . . . .	22
5.3	Система автоматизированного построения планов для распре- деления деталей по линейкам . . . . .	22
5.3.1	Постановка задачи . . . . .	22
5.3.2	Описание алгоритма . . . . .	23
5.3.3	Разработка системы . . . . .	23

5.3.4	Изменения в производственном процессе, полученные после внедрения системы . . . . .	24
5.4	Приложение для построения планов литья деталей . . . . .	24
5.4.1	Постановка задачи . . . . .	25
5.4.2	Описание алгоритма . . . . .	25
5.4.3	Разработка системы . . . . .	30
5.4.4	Схема устройства базы данных . . . . .	30
5.4.5	Изменения в производственном процессе, полученные после внедрения системы . . . . .	31
6	Заключение . . . . .	31

## 1. Введение

В течение последних двух десятилетий стремление добиться преимущества над другими компаниями в сфере производства стало движущей силой для использования различных инновационных технологий. Для повышения эффективности компании используют самые различные методы, системы, позволяющие добиться максимальной прибыли. Одним из наиболее известных является принцип «*just-in-time*» (точно в срок) [1, 2]. Данный принцип послужил базой в разработке основных методологий в компании Toyota Motor и подразумевает, что все цеха производят необходимые детали к нужному времени, а запас материалов для их производства минимален. Обеспечение чёткой работы производства согласно упомянутому принципу — сложная задача, так как необходимо контролировать все процессы, обеспечивающие работу производства. Современная логистическая система на производстве базируется на следующих основных принципах: отказ от избыточных запасов, сокращение времени на транспортно-складские операции, снижение количества брака и предотвращение его возникновения, применение количественной и качественной гибкости производственных мощностей.

В целях обеспечения бесперебойного производственного процесса и минимизации влияния человеческого фактора является разумным использование автоматизированных решений, способных эффективно выполнять основные задачи, такие как составление планов, анализ статистических данных и оценка эффективности на их основе.

Автоматизация сейчас — один из наиболее эффективных способов достижения лучшего результата. Замена в производственной отрасли интеллектуального труда человека машинным приводит к повышению эффек-

тивности процессов, снижению затрат, обеспечению максимального уровня оперативности и гибкости. Универсальные компьютерные системы всех уровней, реализующие различные аспекты автоматизации процессов, крайне многочисленны. Из всего их многообразия достаточно сложно выделить из них те, что в полной мере удовлетворяли потребности конкретного производства. Таким образом, создание систем автоматизации, сочетающих в себе доступность для пользователя, многофункциональность, способность учитывать специфику конкретной производственной системы, является актуальной и важной задачей.

## **2. Цель работы**

Цель работы — разработка программного обеспечения, позволяющего автоматизировать производственные процессы, сократить временные затраты сотрудников на выполнение стандартных рабочих операций и повысить эффективность их выполнения.

## **3. Обзор литературы**

Для того, чтобы лучше усвоить, понять рассматриваемые далее процессы были изучены книги [3, 4], в которых описываются основные понятия и определения связанные с логистикой, в частности, с массовым производством, по какому принципу оно устроено, за счет чего регулируются разные процессы. Также в статьях [1, 5] описан упомянутый принцип «just-in-time», лежащий в основе устройства рассматриваемого производства, история его появления, важность для логистических систем, а также подходы для его обеспечения. В статьях [2, 6, 7] содержится информация о системе Канбан и ее развитии в современном обустройстве процессов: как она модифицируется со временем, как помогает обеспечению принципа описанного

выше. В качестве технической документации использовались [8, 9, 10].

#### **4. Введение в предметную область**

Логистика – процесс планирования и контроля эффективного с точки зрения снижения затрат потока сырья, материалов, незавершенного производства, готовой продукции, сервиса и связанной информации от точки зарождения до точки потребления (включая импорт, экспорт, внутренние и внешние перемещения) для целей полного удовлетворения требований потребителей [3]. Предметом логистики является комплексное управление всеми материальными и нематериальными (информационными, финансовыми, сервисными) потоками в системах. Основными объектами исследования в логистике являются: логистические издержки, информационный поток, логистическая система, логистическая функция, логистическая цепь, логистические операции, материальный поток и др. Основная цель логистики – обеспечить наличие необходимого продукта в необходимом количестве, в необходимом состоянии, в необходимом месте, в необходимое время и по подходящей потребителю цене с минимальными для предприятия издержками. Главное условие достижения цели – готовность сырьевых материалов, полуфабрикатов, товаров и их компонент к переходу в новое звено производственной цепи (монтажирование, комплектация заказов, отправление, доставка) при возникновении спроса внутри производства, либо за его пределами. Поставка материалов, сырья, готовой продукции “точно в срок” оказывает благоприятное влияние на функционирование всей экономической системы, позволяет существенно (иногда в 3-4 раза) сократить запасы на складах промышленных предприятий. Определим ряд необходимых понятий:

- Логистическая система – это организационно завершённая (структу-

рированная) экономическая система, которая состоит из элементов-звеньев, взаимосвязанных в едином процессе управления материальными и сопутствующими им потоками, причём задачи функционирования этих звеньев объединены общими целями организации бизнеса и (или) внешними целями.

- Логистические операции (функции) – действия, осуществляемые над материальными (или сопутствующими) потоками в логистической системе.
- Логистическая цепь – несколько звеньев, упорядоченных по какому-либо потоку. Полная логистическая цепь – упорядоченное по какому-либо потоку множество звеньев, по которым этот поток проходит от входа в логистическую систему до выхода из нее.
- Логистическая сеть – все звенья логистической системы, связанные между собой по материальным и сопутствующим им потокам. От понятия «логистическая система» логистическую сеть отличает отсутствие логистического менеджмента, реализующего целевую функцию системы. Очевидно, что в реальных организациях бизнеса полные логистические цепи (соответствующие определению) встречаются крайне редко. Поэтому концепцию общих издержек обычно связывают с понятием логистического канала.

Логистические системы бывают следующих видов:

1. «Тянущие» системы (pull systems) – системы, в которых последующий участок производства заказывает и изымает детали, сборочные единицы с предыдущего участка. Размещение заказов на пополнение запасов

материальных ресурсов или готовой продукции происходит, когда их количество в определенных звеньях логистической системы достигает критического уровня.

2. «Толкающие» системы (push systems) – системы, рассматривающие все звенья производства как единое целое. Ставка делается на планирование потока материалов. При этом недостаточно строго отслеживается спрос и обязательно наличие страховых запасов.

Рассматриваемый далее принцип «just in time» является самым популярным примером тянущих систем.

#### **4.1. Массовое производство**

Массовое производство характеризуется узкой номенклатурой и большим объемом выпуска изделий, непрерывно изготавливаемых в течение продолжительного времени на узкоспециализированных рабочих местах [4]. Логистическая концепция организации массового производства включает в себя следующие основные положения:

- 1) отказ от избыточных запасов;
- 2) отказ от завышенного времени на выполнение основных и транспортно-складских операций;
- 3) отказ от изготовления продукции, на которую нет спроса;
- 4) обязательное устранение брака;
- 5) устранение нерациональных внутризаводских перевозок.



## 4.2. Kanban, Just in Time

Одной из первых попыток практического внедрения концепции «точно в срок» явилась разработанная корпорацией Toyota Motor микрологистическая система Kanban (Канбан) (что в переводе с японского означает «карта»). В дальнейшем с помощью системы получилось революционизировать то, как корпорации и поставщики достигают максимальной эффективности при получении продуктов на сборочной линии. При правильном применении Канбан минимизирует количество отходов, предотвращает перепроизводство и обеспечивает быстрый ответ на изменения и проблемы. В рассматриваемой системе используется форма карточки заказов под названием Канбан [5]. Они бывают двух видов:

- «транспортный Канбан», который несут при переходе от одного процесса к предыдущему процессу
- «производственный Канбан» — используется для заказа продукции для последующих процессов. (дает команду предыдущим процессам производить детали для последующих процессов) Как это работает:

1. Рабочему-транспортнику с последующего этапа поступает сигнал забрать материалы со склада. Этот сигнал определен последующим процессом, а также одним из двух сигналов:

- сигналом о количестве собранных Канбанов перемещения;
- сигналом о периодическом промежутке времени.

Далее рабочий направляется на склад предыдущего процесса с пустыми поддонами и собранными Канбанами перемещения в виде листов

заказов, в которых указано, что и в каком количестве требуется для последующего процесса.

2. Детали, изготовленные на предыдущем процессе, складываются в поддоны, к каждому из которых прикрепляется Канбан, они помещаются на склад (этот процесс проходит независимо от процесса 1).
3. Транспортировщик собирает детали, указанные в Канбане перемещения (листе заказов), сверяет их параметры с производственными Канбанами, прикрепленным к деталям, затем заменяет один Канбан на другой.
4. Транспортировщик помещает производственный Канбан на «доску производства», что позже визуально подаст предыдущему процессу сигнал к производству, когда число Канбанов достигнет критических значений.
5. Транспортировщик доставляет необходимые детали вместе с Канбаном перемещения со склада на последующий этап производства.

Очевидно, что склад служит посредником между двумя процессами, существующими независимо друг от друга и обменивающимися деталями и информацией посредством Канбана. На Канбанах записывается информация о названии и номере детали, ее количестве, типе поддона, адресе склада, чтобы транспортировщик деталей имел полное представление о необходимых действиях. Правила Канбан также говорят операторам, что делать, когда у них есть проблемы и кому идти, когда эти проблемы возникают. Сегодня Toyota Motor продолжает использовать систему не только для управления затратами и расходами, но и для определения препятствий для потока и возможности для постоянного совершенствования [6].

Принцип «just in time» заключается в том, что все производственные подразделения завода, включая линии конечной сборки, снабжаются материальными ресурсами необходимых для выполнения заказа количестве и требуемому сроку, заданными потребителем. Таким образом, в отличие от традиционного подхода к производству, структурное подразделение не имеет общего жесткого графика производства, а оптимизирует свою работу в пределах заказа подразделения фирмы, осуществляющего операции на последующей стадии производственно-технологического цикла [7].

### 4.3. Основные обозначения

Введем следующие определения относящиеся к рассматриваемому предприятию:

- Takt Time ( $TT$ ) — промежуток времени между выпуском изделий (деталей, сборочных единиц) с последней операции или их запуском на первую операцию поточной линии;
- $Q_i$  — количество выпускаемых автомобилей типа  $i$ ,  $i = \overline{1, n}$ ;
- $n$  — количество моделей, производимых на предприятии;
- $q_j$  — количество деталей, содержащихся в контейнере  $j$ ;
- Shunt Time ( $ST$ ) — время доставки контейнера на предприятие со склада;
- Unloading Time ( $UT$ ) — время разгрузки контейнера;
- Док — место выгрузки комплектующих. Доки объединены в группы по четыре и находятся в разных частях производства;

- муда — это любая деятельность, которая потребляет ресурсы, но не создает ценности для клиента;
- дандори — время замены материала;
- каунтер (counter) — общий счетчик для всех автомобилей, выпускаемых за день.

## **5. Информационные системы**

В данной работе представлены описания разработанных мною информационных систем (приложений) для оптимизации различных задач на производстве. Все представленные временные и количественные данные искусственно просимулированы во избежание раскрытия коммерческой тайны.

### **5.1. Система для автоматизации построения плана поставок и разгрузок автокомпонентов на производство**

Для отдела логистики автокомпонентов была разработана система по автоматизации построения плана поставок и разгрузок на предприятии.

#### **5.1.1 Постановка задачи**

Имея исходные данные, необходимо в зависимости от планируемого количества выпуска каждой модели и  $TT$  построить план доставки и разгрузки компонентов на производство с учётом ограничения: одновременно могут разгружаться не более четырех контейнеров, так как количество погрузчиков на предприятии ограничено.

### 5.1.2 Описание алгоритма

На первом этапе строится план следования моделей автомобилей на линии сборки, так как именно для ее обеспечения и доставляются детали.

1. Вычисляем соотношение

$$Ratio_i = \frac{Q_i}{\sum_{i=1}^n Q_i}, \quad i = \overline{1, n}.$$

2. Вычисляем

$$Step_j = Ratio_i \cdot j, \quad j = \overline{1, Q_i}, \quad i = \overline{1, n}.$$

3. Ранжируем полученные значения (таблица 1).

Time	Counter	Model1	Model2
7:20:00	0	0	0
7:22:54	1	1	
7:25:48	2		1
7:28:42	3		2
7:31:36	4	2	
7:34:30	5		3
7:37:24	6		4
7:40:18	7		5
7:43:12	8	3	

Таблица 1. Пример производственного плана цеха сборки

Рассмотрим таблицу 2 – в ней отображены данные по контейнерам и сводная информация по ним. В таком виде она хранится в базе данных и доступна пользователю приложения для изменения.

id	id_model	name	id_dock	$q_j$	ShuntTime	UnloadingTime
1	1	H	7	40	39	30
2	1	W	2	70	46	45
3	1	Q	15	28	39	35
4	2	F	3	24	46	30
5	2	E	8	24	39	30
6	2	N	16	23	39	35

Таблица 2. Пример исходных данных

Рассмотрим алгоритм действий на примере: пусть известен набор деталей, необходимых на текущий день для производства, и место разгрузки контейнеров на территории предприятия. Например, компонент требуется F для модели 2 и содержится в контейнере в количестве 40 шт. Значит к тому моменту, когда будет выпущена 40-я по счету модель 2, например,  $counter = 80$ , контейнер с этими компонентами должен быть доставлен и разгружен.

$$X = \frac{(ShuntTime + UnloadingTime)}{TT},$$

где  $X$  — количество промежутков времени, за которые выполняются поставка и разгрузка комплектующих. Следовательно, заказывать контейнер необходимо при  $counter$ , равном  $(80 - X)$ . Аналогичным образом производится расчет для каждого компонента. Если разгрузка контейнера пересе-

кается с другими уже расставленными четырьмя, то разгрузка и доставка сдвигаются вверх, пока не будет выполнено условие, что на протяжении всего времени разгрузки этого контейнера происходят не более четырех разгрузок.

### 5.1.3 Разработка системы

Первым этапом, после построения плана следования автомобилей на конвейере, с помощью хранимой процедуры в таблицу вносится локальная разгрузка (разгрузка, которая соответствует фиксированному расписанию — рис. 1), и перерывы, так как разгрузка на их протяжении осуществляться не может. В том порядке, в котором компоненты хранятся в СУБД,

id_milkrun	id_Shift	Supplier	id_help	id_Docks	Start unloading	Finish unloading	comment1	comment2
1	1	local	1	9	08:20:00.0000000	09:00:00.0000000	милк ран	стекла
2	1	local	2	9	09:25:00.0000000	09:45:00.0000000	милк ран	АКБ
3	1	local	3	9	09:50:00.0000000	10:20:00.0000000	милк ран	шумоизоляция, проводка
4	1	local	4	9	12:30:00.0000000	13:10:00.0000000	милк ран	стекла
5	1	local	5	9	13:20:00.0000000	14:00:00.0000000	милк ран	шумоизоляция
6	1	local	6	9	15:00:00.0000000	15:20:00.0000000	разгрузка ковриков	-
7	1	local	1	1	10:00:00.0000000	10:30:00.0000000	милк ран	разгрузка шин
8	1	local	2	1	15:00:00.0000000	15:30:00.0000000	милк ран	разгрузка шин
9	2	local	7	9	17:20:00.0000000	18:00:00.0000000	милк ран	стекла
10	2	local	8	9	18:10:00.0000000	18:50:00.0000000	милк ран	шумоизоляция, проводка
11	2	local	9	9	22:20:00.0000000	23:00:00.0000000	милк ран	стекла
12	2	local	10	9	23:20:00.0000000	23:50:00.0000000	милк ран	шумоизоляция
13	2	local	3	1	19:10:00.0000000	19:40:00.0000000	милк ран	разгрузка шин
14	2	local	4	1	00:30:00.0000000	01:00:00.0000000	милк ран	разгрузка шин

Рис. 1. Пример расписания фиксированной разгрузки

согласно алгоритму они вносятся с учетом ограничения в итоговую таблицу, в которой отображено время начала и конца разгрузок и доставок автокомпонентов. Если необходимо добавить контейнер вручную, тогда достаточно перейти в соответствующую вкладку в главной форме (рис. 2). Пользователю в зависимости от выбранного им дока предлагается время, где возможно расположить контейнер, в зависимости от времени его разгрузки. Построенный план можно сохранить в виде Excel/xls-файла.

Данная система реализована в RADStudio 10.1 на языке C++. Для это-

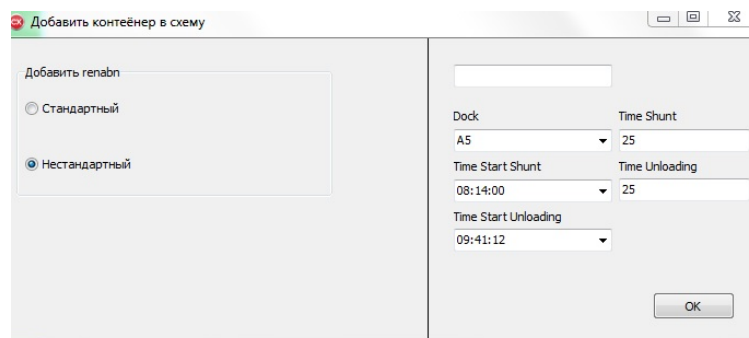


Рис. 2. Добавление контейнера

го использовались библиотеки `<Excel_XP_SRVR.h>` и `<OleServer.hpp>` [8], обеспечивающие связь с Excel. Главная таблица отображается в компоненте DBGrid. Данные по умолчанию запрещены для редактирования. Основные алгоритмы выполняются на стороне базы данных, в связи с возможностью быстрой обработки данных с помощью хранимых процедур.

#### 5.1.4 Схема устройства базы данных

Все построенные планы должны быть доступными для определенных пользователей и храниться в базе данных, а именно, MS SQL Server, ввиду его высокой производительности и безопасности данных. В базе данных хранятся таблицы: расписание фиксированных разгрузок, данные по контейнерам, времени их доставке и выгрузке, а также итоговая — с планом, отображаемым в основном окне приложения. Итоговая таблица содержит следующие столбцы: :

- 1) дата построения плана
- 2) время
- 3) каунтер для каждой модели



4) названия доков A1–A15

5) A1\_id–A15\_id

6) A1\_color–A15\_color

Пользователю видны только 2,3,4 группы столбцов. Первый столбец является идентификатором плана конкретного дня, группа 5 нужна для занесения в эти столбцы идентификатора разгрузки. В последние заносятся идентификаторы цвета, например 2 — в тот столбец, где происходит доставка на протяжении времени её осуществления, а 3 — на протяжении времени разгрузки. Таким образом, при прорисовке таблицы программа закрашивает в отображаемых столбцах обозначенные ячейки соответствующими цветами. Основные хранимые процедуры выполняют:

1. Вычисление плана сборки с учетом  $TT$  и расстановку фиксированных разгрузок, выполняемых строго по расписанию, хранящемуся в базе данных.
2. По набору комплектующих, который пользователь сформировал на текущий день, строится вспомогательная таблица, содержащая в себе время начала доставки и разгрузки для каждого компонента без учета ограничения.
3. Последовательно из таблицы пункта 2 в порядке, установленном пользователем, доставка и разгрузка контейнера заносятся в основной план. Если разгрузка пересекается еще с четырьмя другими, то разгрузка сдвигается вверх по времени, до того момента, как ограничение будет выполнено. Соответственно, сдвигается и доставка компонента. Если

разгрузка выходит за рамки текущего производственного дня, пользователю приходит уведомление, после которого ему необходимо вручную задать желаемое время ее исполнения.

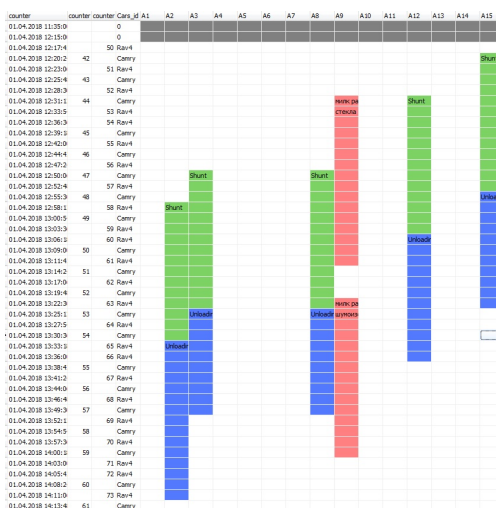


Рис. 3. Пример построенного в приложении плана

### 5.1.5 Изменения в производственном процессе, полученные после внедрения системы

Разработанная система позволила сократить временные затраты сотрудника с 90 минут до 7 и исключить вычислительные ошибки при построении данного плана. На рис. 3 отображен вариант построения плана поставок деталей на предприятие. Первый столбец — временная шкала рабочего дня, следующие два столбца обозначают каунтеры по каждой из моделей, третий столбец показывает название модели. Далее идут столбцы соответствующие, номерам доков, куда приходят контейнеры. Зелёным цветом отображается доставка контейнера, синим — его разгрузка. Серым

цветом выделены перерывы. Другая расцветка обозначает локальную разгрузку.

## **5.2. Приложение для работы с базой данных излишек**

Также для отдела логистики автокомпонентов был разработан программный продукт, позволяющий отслеживать количество излишков на момент окончания производственного дня и создавать инструкцию на повторную выгрузку контейнера.

### **5.2.1 Постановка задачи**

Детали, доставляемые на производство, идентифицируются по номеру Канбана и номеру контейнера, в котором они были доставлены. По окончании производственного дня оператор сканирует детали, не используемые за день. На основе этих данных создается CSV файл с излишками. На основе полученного csv-файла и шести Excel файлов сотрудникам приходилось вручную находить и отслеживать количество разных модулей в различных контейнерах, а после этого создавать. Была поставлена задача создания приложения, позволяющего пользователю без труда загружать эти файлы в систему и производить удобную корректировку данных.

### **5.2.2 Описание алгоритма**

Система заносит все известные данные в общую базу данных, хранящуюся на сервере. С помощью хранимых процедур происходит обработка данных:

- 1) подсчитываются одинаковые модули в каждом контейнере из файла по излишкам;

- 2) значение уникального номера детали подтягивается из соответствующего файла парт листа от производителя, первичным ключом которого является номер Канбана;
- 3) признаком для определения контейнера является то же поле – Канбан, но уже из берется другой таблицы, доставляемых деталей на производство;
- 4) формируется общая сводная таблица по разным излишкам.

Используя разработанное приложение, сотрудник может находить информацию по интересующему его параметру с помощью поиска и фильтра, редактировать данные вручную, если в этом есть необходимость, и выбирать контейнеры и модули, которые будут выданы на производственную линию.

№	Part No.	Part No. 1	Kanban	Module	Order lot	Q-ty boxes	Q-ty pieces	Container №
1	8273033P4000	82730-33P40-00	D253	SMR-346		7	1	7 HXLU8231395
2	853503314100	85350-33141-00	K408	SMR-438		3	1	3 HXLU1317669
3	813603305080	81360-33050-80	K270	SMR-179		8	3	24 HXLU8239884
4	5539442050C0	55394-42050-C0	F067	SMR-384		22	2	44 HXLU8231395
5	679233305000	67923-33050-00	C075	SMR-423		500	1	500 HXLU1317669
6	488203307000	48820-33070-00	K088	SCLT-107		28	1	28
7	434574801000	43457-48010-00	D053	SMR-358		8	2	16 HXLU1608978
8	909420327200	90942-03272-00	C457	WB-005		100	1	100 LCRUS760312
9	731784202080	73178-42020-80	Y373	SMR-206		56	1	56 RSTU4002991
10	426031275000	42603-12750-00	C025	SMR-217		40	1	40 WJNJ9910950
11	909420321300	90942-03213-00	C447	WB-008		200	2	400 LCRUS760312
12	909420321300	90942-03213-00	C447	WB-009		200	1	200 LCRUS760312
13	909420321300	90942-03213-00	C447	WB-006		200	2	400 LCRUS760312
14	909420327200	90942-03272-00	C457	WB-002		100	3	300 YMLU8152364
15	909420321300	90942-03213-00	C447	WB-007		200	3	600 LCRUS760312
16	909420321300	90942-03213-00	C447	WB-005		200	3	600 LCRUS760312
17	821824206100	82182-42061-00	F333	SMTRAV-010		10	1	10 RSTU4000751
18	909420321300	90942-03213-00	C447	WB-002		200	2	400 YMLU8152364
19	909420321300	90942-03213-00	C447	WB-003		200	5	1000 GLDU7155453
20	9094702F4700	90947-02F47-00	O156	SMTRAV-026		40	1	40 YMLU8152364

Рис. 4. Главная форма приложения

На рис 4. представлена главная форма с общей сводной таблицей, где указаны для каждого Канбана его парт номер, модуль, которому он соответствует, количество в контейнере и номер контейнера, в котором он располагается.

### **5.2.3 Разработка системы**

Программа также разработана на языке C++ в Embracadero Rad Studio, в качестве базы данных используется Microsoft SQL Server. Компонент, используемый для отображения таблиц — DBGrid, связанный с базой данных через Data Source, где выбран клиентский тип курсора, это означает, что для удобного подключения разными сотрудниками подразделения к разработанной системе данные переносятся на компьютер пользователя, а не открыты напрямую через сервер. Такой тип устройства взаимодействия с данными, позволяет производить изменения данных на стороне клиента, а уже после отправлять данные на сервер, что обеспечивает устойчивость данной системы.

При загрузке данных система проверяет файл на тот материал, находящийся в загружаемом файле. Если формат данных не соответствует заявленному, система откажет в загрузке такого файла, чтобы избежать неправильного соотнесения файла впоследствии, при обработке данных хранимой процедурой.

### **5.2.4 Схема устройства базы данных**

Для хранения данных созданы таблицы, в которых хранятся данные по загружаемым файлам. При добавлении файла в систему данные заносятся с соответствующей датой, а информация предыдущего дня не удаля-

ется, чтобы в любой момент можно было осуществить резервное копирование. Для того чтобы реализовать быструю фильтрацию по нескольким параметрам, вызывается хранимая процедура, которая принимает на вход название таблицы, строку, хранящую названия полей, по которым хотим отфильтровать значения, и строку со значениями этих полей.

#### **5.2.5 Изменения в производственном процессе, полученные после внедрения системы**

Временные затраты сокращены до 3-х минут на загрузку и обновление данных, создание готовой инструкции на выгрузку деталей. При необходимости пользователь может выгрузить все отредактированные данные в удобном для него формате CSV/EXCEL.

### **5.3. Система автоматизированного построения планов для распределения деталей по линейкам**

Для отдела логистики была разработана информационная система, позволяющая составлять план распределения деталей по линейкам с паллетами.

#### **5.3.1 Постановка задачи**

Имеется файл CSV хранящий в себе номера Канбанов, которые в определенный день должны поступить на линию в нужном объеме. В нем приходилось выделять детали по поставщикам, удалять те, которые не распределяются по паллетам и другие, которые должны пройти контроль качества пометать отдельно. Необходимо было создать удобное приложение, которое будет выполнять действия за сотрудников соответствующего подразделения.

### 5.3.2 Описание алгоритма

На вход программе подается дата, план Getsudo (план, который был просчитан за два предыдущих месяца) и план, учитывающий конкретные возможности производства на текущий день. Программа выбирает детали, относящиеся к этой дате, и распределяет их по линейкам. Известно, что при стандартном дне, линеек должно быть 16 (заданное постоянное число). Рассмотрим подробнее алгоритм:

1. Вычисляем общий шаг каунтера для каждой из шестнадцати линеек, т.е. за какой промежуток будет выдана полностью каждая из линеек:

$$len = \frac{GetsudoPlan}{16}.$$

2. Вычисляем количество линеек:

$$N = \frac{Plan}{len}.$$

Если  $N > 16$ , тогда берем детали следующего дня, а если  $N < 16$ , то оставшиеся линейки переходят на следующий день.

Система разработана для быстрого формирования Excel файла нужного формата, а именно — линейки должны отображаться блоками по 4, детали разных поставщиков выделяются разными цветами, а детали на качество выделяются фоном.

### 5.3.3 Разработка системы

Программа реализована с помощью Microsoft Visual Studio на языке C# и связана с базой данных (рис. 5) [9]. Для визуальных компонентов

были выбраны модули Metro [10]. Для работы с Excel и базой данных были написаны два основных класса, чтобы избежать повторяющихся действий в тексте программы. Для этого использовались библиотеки Microsoft такие как, Interop.Excel, Office.Core, VisualBasic, System.Data.SqlClient. Чтобы подключение к базе данных было скрыто от пользователя, строка подключения к базе данных прописана в исходном коде, а не в файле конфигурации.

Рис. 5. Главная форма

### 5.3.4 Изменения в производственном процессе, полученные после внедрения системы

Разработанное приложение позволяет исключить ошибки, возникающие при пересчете нового плана, с учётом фактических значений. Время создания плана сотрудниками было сокращено до 5 минут.

### 5.4. Приложение для построения планов литья деталей

Система разработана для производственного цеха литья пластиковых деталей, поставляемых на линию сборки.



### 5.4.1 Постановка задачи

На производстве работают в штатном режиме две машины литья пластиковых частей: бамперов, ip-панелей и др. Они отливаются поочередно в количестве, необходимом для обеспечения производства. Для каждого типа детали известно время её литья в секундах и модель автомобиля, для которой она отливается. Детали делятся на категории, например передние бампера и задние — категория 1, ip-панели — 2, При смене категории необходима смена материала. Время смены пресс-формы —  $t_d$ . Время смены материала —  $t_m$ . Необходимо минимизировать простой при условии, что детали не будут отливаться большими блоками, так как место для готовой продукции также ограничено.

### 5.4.2 Описание алгоритма

Пользователь вводит значение плана на сегодня и корректирует временные параметры, если это необходимо. После этого пользователю необходимо выбрать какие детали будут отливаться на одной или обеих машинах (рис. 6). Так как последовательность отливаемых деталей напрямую влияет на итоговое производственное время, у пользователя есть возможность поменять последовательность следования деталей в циклах. На (рис. 7) представлена форма, в которой последовательными «кликами» по блокам с названиями деталей пользователь задает порядок их следования в одном цикле. Нажатием кнопки на перенесенный блок пользователь может переместить его на первоначальное место. При необходимости весь порядок можно изменить на первоначальный «Cancel».

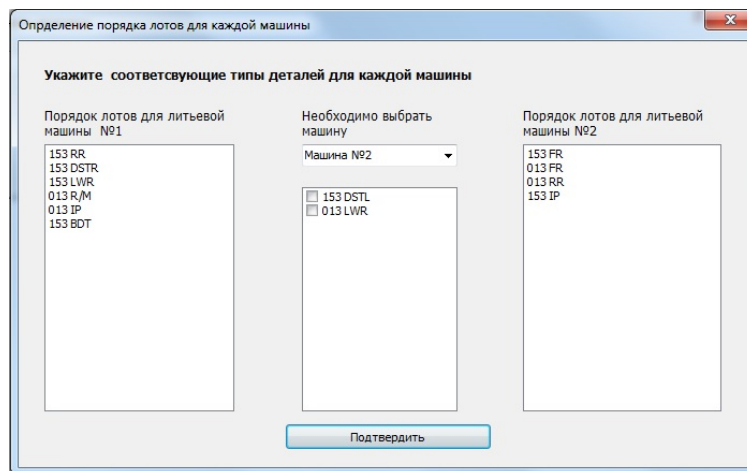


Рис. 6

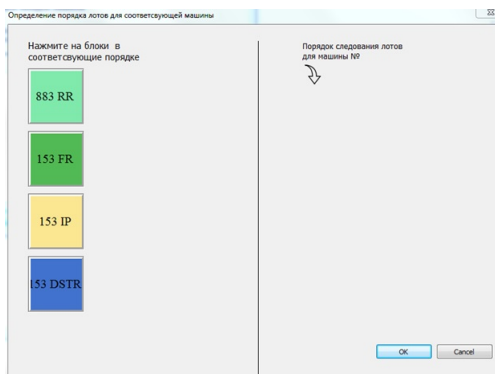


Рис. 7

Задав циклы для каждой из машин, пользователь нажимает на кнопку «построить план». Далее отображается окно, (рис. 8), на котором отображено оптимальное количество отливаемых деталей в одном лоте. Вычисление оптимального плана можно описать следующим алгоритмом.

1. Время литья каждой из  $n$  выбранных деталей умножается на план по производству автомобиля, которому отливаемый компонент соответствует:

$$T_{ij} = t_i * plan_j, \quad i = \overline{1, n}, \quad j = \overline{1, 2}.$$

2. Общее время литья выбранных деталей вычисляем по следующей формуле:

$$T = \sum_{i=1}^n T_i.$$

3. Задаемся начальным значением количества смен материала  $qm$ :

$$qm = n * 2.$$

4. Определение времени на смену всех пресс-форм:

$$T_d = WT - T - qm * t_m.$$

5. Определение суммарного количества отливаемых лотов:

$$Q = \frac{T_d}{t_d}.$$

6. Количество отливаемых лотов для каждого типа деталей:

$$Q_{ij} = \frac{plan_j}{Q}, \quad i = \overline{1, n}, \quad j = \overline{1, 2}.$$

7. Пересчитываем количество смен материала, проверяя категории соседних лотов:

$$qm = \begin{cases} 1, & \text{соседние лоты одной категории;} \\ 0, & \text{иначе.} \end{cases}$$

8. Возвращаемся к шагу 4.

Так продолжается до тех пор, пока вычисленное число количеств замены материала не совпадет с вычисленным на предыдущем шаге алгоритма.

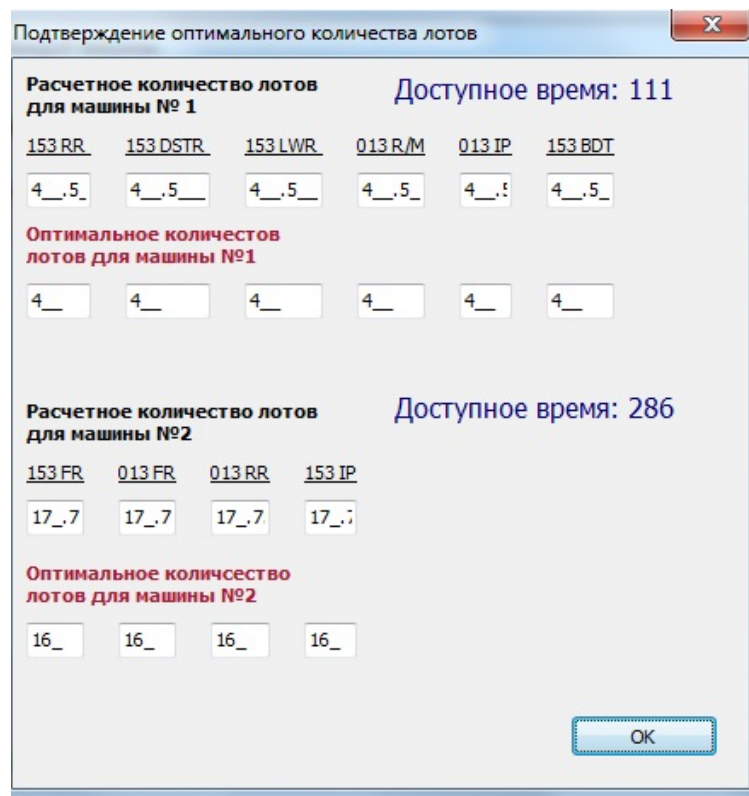


Рис. 8

На финальном этапе в главной форме отображаются цветные блоки (рис. 9), масштаб которых соотносится с временными затратами на литьё данного лота. Блоки могут перемещаться внутри компонентов Scrollbox, по каждому блоку нажатием правой кнопки мыши можно просмотреть информацию по этому лоту: время начала, время конца, количество деталей в лоте (рис. 10). Пользователь может изменить количество деталей, после изменения система пересчитывает и отображает актуальные данные по требуемому лоту. Также пользователь имеет возможность добавить или удалить лот, сдвинуть или изменить время перерыва (рис. 11).

При ручном изменении плана — порядка или количества деталей в лотах, их удаления/добавления, пользователю необходимо нажать на кнопку

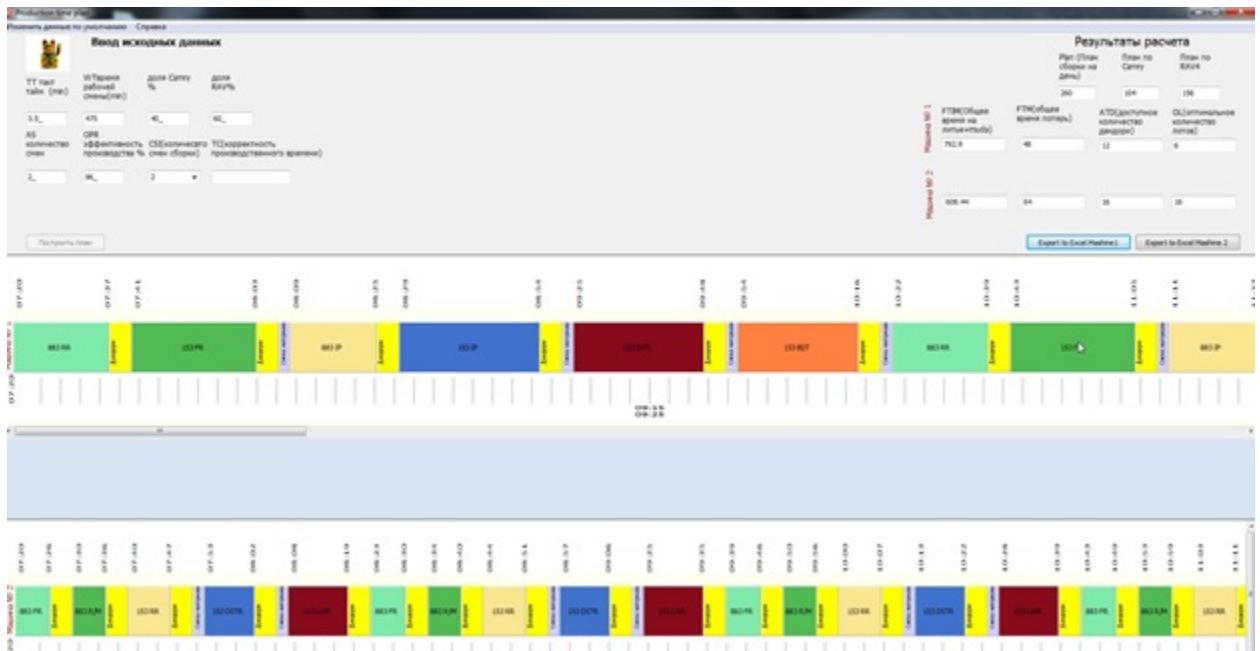


Рис. 9. Вариант построенного плана

«Машина № », после этого система пересчитывает и создает новые временные интервалы. Если при отправке плана на печать суммарное число деталей по всем лотам не совпадает с планом, система оповещает пользователя.

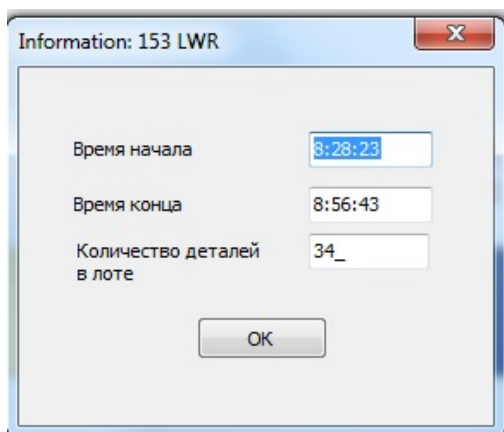


Рис. 10. Окно редактирования лота.

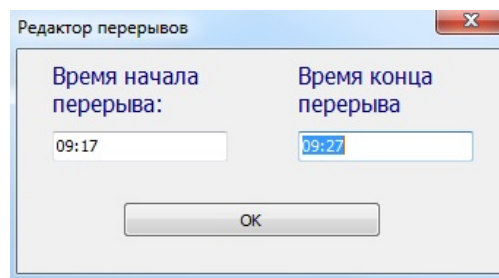


Рис. 11. Окно редактирования перерывов.

### 5.4.3 Разработка системы

Программа также разработана на языке C++ в Embracadero Rad Studio. В качестве базы данных используется Ms SQL Server. возможных форм динамические визуальные компоненты создаются при открытии любой из форм и уничтожаются при ее закрытии. Такие компоненты необходимы, так как наборы отливаемых деталей могут изменяться.

### 5.4.4 Схема устройства базы данных

В базе данных созданы таблицы с данными по деталям, которые заносятся через вкладку главной формы (рис. 12). Также создана таблица, в которую заносится дата, значения плана по каждой из модели, времени рабочего дня, строка с названиями деталей, строка с оптимальными значениями плана.

Название детали	153 FR	153 RR	153 DSTL	153 DSTR	153 LWR	013 FR	013 RR	013 R/M	013 IP	153 IP	153 BDT	013 LWR
Действительный СТ(Сек)	49.1_	45.5_	54.____	54.____	50.____	50.5_	56.____	50.5_	52.____	58.____	50.____	46.8_
Серв-е детали; шт/д.	0_	0_	0_	0_	0_	0_	0_	0_	0_	0_	0_	0_
Процент окраски	100	100	100	100	100	100	100	100	100	100	100	100

Рис. 12. Форма корректировки данных по лотам

#### 5.4.5 Изменения в производственном процессе, полученные после внедрения системы

Созданный программный продукт позволяет сократить время построения плана сотрудниками соответствующего цеха. Система динамично и гибко реагирует на любые ручные изменения и помогает пользователю построить наиболее подходящий вариант плана производства пластиковых частей.

## 6. Заключение

В данной работе были автоматизированы следующие процессы на производстве: планирование поставок и разгрузок автокомпонентов, создание инструкций на разгрузку контейнера с излишками, создание планов распределения деталей по линейкам и построение плана литья пластиковых частей. Разработанные программы позволяют значительно сократить временные затраты на составление планов и повысить качество планирования на производстве. Каждая из разработанных систем учитывает особенности конкретного предприятия, что позволяет строить рациональные решения в рассматриваемых процессах.

В дальнейшем исследовании построения рациональных решений планируется расширить возможности программ и проводить имитационное

моделирование, которое позволит выявить наиболее слабые места разработанных алгоритмов, что даст возможность модифицирования программ для еще более продуктивной работы.



## Список литературы

1. Sugimori Y., Kusunoki K., Cho F., Uchikawa S. Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system // International Journal of Production Research. 1997. P. 553–564.
2. Axsater S., Rosling K. Ranking of generalised multi-stage Kanban policies // European Journal of Operational Research. 1999. No 3. P. 560–567.
3. Тихомирова А. Н., Сидоренко Е. В. Математические модели и методы в логистике: Учебное пособие. М.: НИЯУ МИФИ, 2010. С. 320.
4. Peter Nyhuis, Hans-Peter Wiendahl Fundamentals of Production Logistics // Springer-Verlag Berlin Heidelberg, 2009. P. 39–52
5. Just-in-Time — Philosophy of complete elimination of waste [Электронный ресурс] URL: [http://www.toyota-global.com/company/vision\\_philosophy/toyota\\_production\\_system/just-in-time.html](http://www.toyota-global.com/company/vision_philosophy/toyota_production_system/just-in-time.html)
6. John M. Gross., Kenneth R. McInnis. Kanban Made Simple : Demystifying and Applying Toyota's Legendary Manufacturing Process // AMACOM. 2003.
7. Malik Ghallab, Dana S., Nau, Paolo Traverso. Automated Planning : Theory and Practice // Elsevier Science & Technology. 2004.
8. Vcl.OleServer.TOleServer [Электронный ресурс] URL: <http://docwiki.embarcadero.com/Libraries/Tokyo/en/Vcl.OleServer.TOleServer>
9. Руководство по языку C# [Электронный ресурс] URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>

10. Библиотека компонентов [Электронный ресурс] URL: <https://github.com/MahApps/MahApps.Metro>