

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет»

Специальность Астрономия (011501)
Направление Математика и механика

Бахарев Константин Вадимович

Разработка и создание базы данных наблюдений спутников
системы DORIS для решения геодинимических задач

Дипломная работа

Допущена к защите.

Зав. кафедрой:

д.ф.-м.н., К.В. Холшевников

Научный руководитель:

к.ф.-м.н., В.Б. Титов

Консультант:

к.ф.-м.н., М.В. Васильев

Рецензент:

к.ф.-м.н., В. Л. Горшков

Санкт-Петербург, 2016

SAINT-PETERSBURG STATE UNIVERSITY

Main Field of Study Astronomy(011501)
Area of Specialisation mathematics and mechanics

Konstantin Bakharev

Developing and creating observation database of DORIS
system satellites for geodynamic problem solution

Graduation Thesis

Admitted for defence.

Head of the chair:

(Professor) K. V. Kholoshevnikov

Scientific supervisor:

(Ph.d.) V.B. Titov

Adviser:

(Ph.d.) M.V. Vasiliev

Scientist Reviewer:

(Ph.d.) V.L. Gorshkov

Saint-Petersburg, 2016

Оглавление

1	Тезисы	4
2	Введение	5
3	Система спутников DORIS	6
4	Программный комплекс	10
4.1	Общая схема	10
4.2	Рабочий цикл программы	11
5	Постановка задачи	13
6	Модуль базы данных	14
6.1	База данных наблюдений спутников DORIS	14
6.2	База данных эфемерид DORIS	16
6.3	Службы для работы с интернетом	17
7	Модуль тестирования	19
8	Заключение	20
	Приложение. Листинг программы	21

Глава 1

Тезисы

В дипломной работе рассматривается задача построения базы данных и модуля тестирования для программного комплекса, который работает с данными DORIS. Главным результатом проделанной работы является реализация:

1. Модуля программного комплекса, отвечающего за агрегацию, хранение, обновление и первичную обработку данных,
2. Метода позволяющий проводить сравнение финальных результатов программы с результатами других ученых.

Глава 2

Введение

В середине 90-х годов в эксплуатацию были запущены несколько глобальных навигационных спутниковых систем. Российская ГЛОНАСС, американское GPS, французская DORIS. Глобальные навигационные спутниковые системы оказались полезны в различных областях научной деятельности, в частности в астрометрии и геодезии.

Обработкой данных полученных от систем ГЛОНАСС и GPS занимается огромное количество групп, а данные системы DORIS в России обрабатывает малое количество групп, возможно это связано с тем, что система обработки данных от International DORIS Service является платной и с закрытым исходным кодом, что мешает использовать свои модели движения спутника.

В институте прикладной астрономии был запущен проект создания программного комплекса, обрабатывающий данные системы спутников DORIS.

Глава 3

Система спутников DORIS

Система спутников DORIS (Doppler Orbitography and Radiopositioning Integrated by Satellite) – доплеровская орбитографическая радиопозиционная интегрированная спутниковая система, была создана в начале 90-х годов французской компанией Dassault Electronique по заказу CNES (Французское географическое агенство*).

Система состоит из двух частей: наземного сегмента (наземные станции) и космического сегмента (спутники с антеннами, принимающие информацию от наземных станций). Существует несколько различных спутников, каждый из них выполняет собственную задачу.

В настоящее время, на Земле работает 58 наземных станций DORIS (рис. 3.1), расположенных так, чтобы в каждый момент времени каждый спутник находился в поле зрения как минимум одной станции (Некоторые зоны остаются без покрытия рис. 3.2). Около половины станций расположена на островах или прибрежных зонах, равномерно заполняя оба полушария. Наземные станции DORIS состоят из радиомаячка, антенны, датчиков давления, температуры и влажности.

Наземная сеть станций состоит из нескольких типов маячков: постоянные, главные и случайные маячки. Постоянные отвечают за возможность точного определения орбит спутников. Три главных маячка (Toulouse, Kourou, Harthebeesthoek) относятся к сети постоянных маячков, но также отвечают за обеспечение синхронизации системы к международному атомному времени (International Atomic Time). Случайные маячки устанавлива-

ются временно, для решения геофизических задач (измерения дрейфа материков).

Маячки передают сигнал на двух частотах: 2036.25 МГц и 401.25 МГц. Последняя частота настроена передавать сообщения, содержащие идентификатор станции, информацию о времени, метеорологическую информацию, служебную информацию.



Рис. 3.1: Расположение маячков на поверхности Земли

Кроме наземной сети станций существует множество спутников. Каждая пара спутника и наземная сеть – являются миссией, решающие свои задачи. В данный момент работают около полудюжины миссий: SPOT-5, JASON-2, JASON-3, CRYOSAT-2, HY-2A, SARAL, SENTINEL-3A.

- Основной задачей миссии SPOT-5 является создание фотоснимков поверхности Земли, для коммерческих целей.
- Миссии JASON, HY-2A уточняют уровень поверхности океана, течения и перемещения воздушных масс.
- Информация о станциях на земле, которые отвечают за синхронизацию.
- Миссии JASON, HY-2A уточняют уровень поверхности океана, течения и перемещения воздушных масс.

Cryosat-2 stations visibilities Elevation 5°

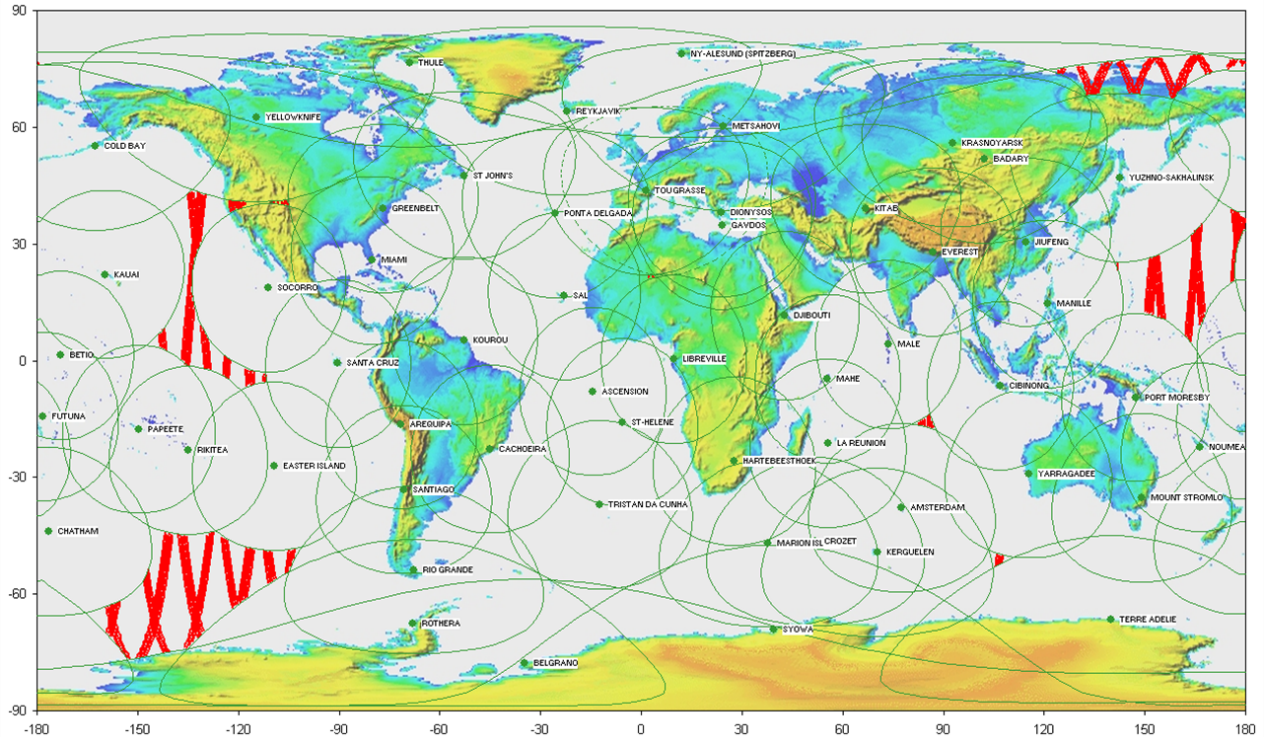


Рис. 3.2: Покрытие зон видимости наземных станций по отношению к спутнику CRIOSAT-2

- CRYOSAT исследуют в полярные льды, фотографии содержат детали с точность до 1,5 сантиметров.
- SARAL исследует морскую погоду, составляет карту океанов, следит за климатом и континентальным льдом.
- SENTINEL-3А следят за температурой на поверхности Земли, отслеживают пожары и эффекты глобального потепления.

Список запущенных спутников с их параметрами:

Спутник	Старт миссии	p (км)	T (мин)	e	Конец миссии
SPOT-2	22-Jan-1990	802	100.9	0.002014	29-Jul-2009
POSEIDON	10-Aug-1992	1322	112	0.000095	9-Oct-2005
SPOT-3	26-Sep-1993	819	101.2	0.001872	14-Nov-1996
SPOT-4	24-Mar-1998	791	100.9	0.001392	24-Jun-2013
JASON-1	7-Dec-2001	1328	112	0.000778	21-Jun-2013
ENVISAT	1-Apr-2002	785	100.6	0.000419	9-May-2012
SPOT-5	4-May-2002	825	101.4	0.000069	-
STPSat-1	9-Mar-2007	558	95.8	-	7-Oct-2009
JASON-2	20-Jun-2008	1328	112	0.000702	-
CRYOSAT-2	8-Apr-2010	707	99.4	0.001407	-
HY-2A	15-Aug-2011	963	104.4	0.001170	-
SARAL	25-Feb-2013	706	99.59	0.001165	-
JASON-3	17-Jan-2016	1328	112	0.000095	-
SENTINEL-3A	16-Feb-2016	802	100.96	0.000300	-

Здесь p – перицентрическое расстояние орбиты спутника, T - орбитальный период, e – эксцентриситет орбиты.

Обработывая данные полученные от спутников можно получить множество результатов:

- Координаты станций, содержащие временные ряды изменений координат наземных станций
- Эфемериды орбит спутников
- Движение геоцентра
- Параметры вращения Земли, движение полюса
- Коррекции ионосферы

Глава 4

Программный комплекс

4.1 Общая схема

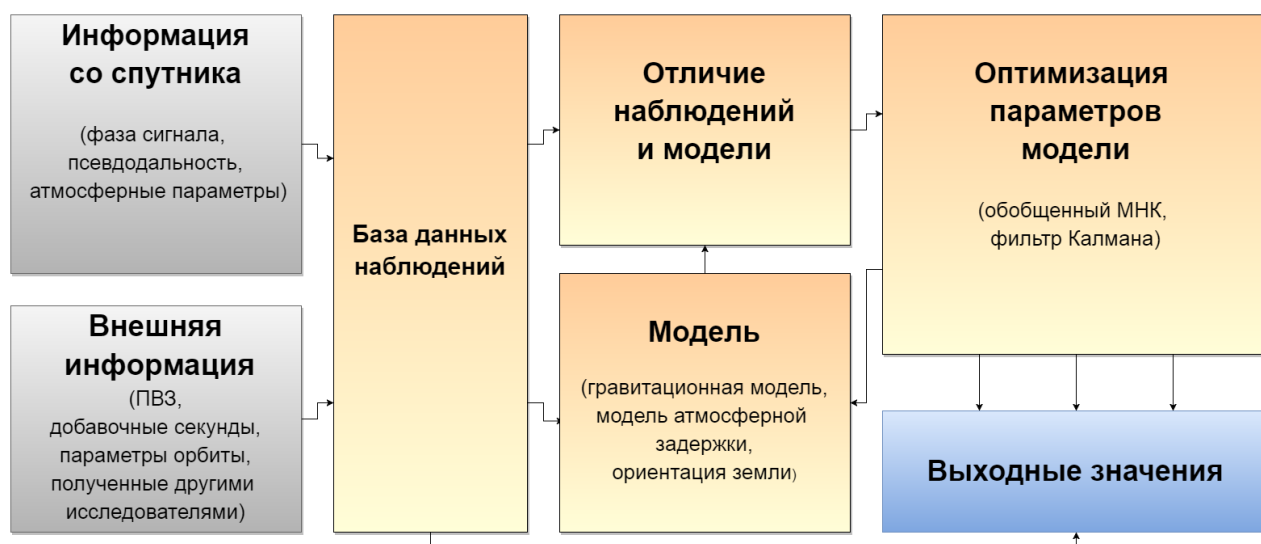


Рис. 4.1: Общая схема программного комплекса

Информация со спутника – информация, которую собирает аппарат DORIS, публикуемая по адресу <ftp://cddis.nasa.gov/doris/data/>. Хранится в виде архивов, содержащих данные, полученные в течение одних суток. Эти данные выкладываются в общий доступ с задержкой в двое суток.

Внешняя информация – информация, которая необходима для работы модели или тестирования. К ней относятся добавочные секунды (leap seconds), ПВЗ (параметры вращения Земли), эфемериды спутников DORIS. (ссылки)

База данных – модуль, который отвечает за получение, хранение и обновление данных.

Модель – часть программного комплекса, которая отвечает за моделирование движения спутника в зависимости от многих параметров.

Модуль оптимизации модели – модуль, отвечающий за оптимизацию параметров в зависимости от разницы между данными, получаемыми в модели и данными, полученными из обработки наблюдений. В качестве побочного продукта оптимизации модели получают улучшенные положения спутников, ПВЗ, и многие другие параметры.

Модуль тестирования – модуль, который аппроксимирует данные полученные другими исследовательскими группами и сравнивает их с результатами, разрабатываемого программного комплекса.

4.2 Рабочий цикл программы

1 Запуск программного комплекса. Осуществляется с целью обучения модели на «исторических данных»;

2 Основной цикл:

2.1. Обновление данных:

2.1.1. Сохранение,

2.1.2. Распаковывание,

2.1.3. Добавление в базу данных.

2.2. Итерационный процесс:

2.2.1. Модель предсказывает положение спутника,

2.2.2. Сравнение невязок (реальные значения – значения, полученные из моделирования),

2.2.3. При больших невязках происходит оптимизация параметров модели,

2.2.4. Возврат к началу итерационного процесса.

2.3. Сохранение данных итерационного процесса;

3 Сравнение полученных результатов в ходе работы программного комплекса ИПА с результатами других групп ученых.

Глава 5

Постановка задачи

Целью данной работы является обеспечение возможности использования данных произведенных системой спутников DORIS в программном комплексе.

Для достижения поставленной цели были сформулированы задачи, приведенные ниже.

- Разработать и запрограммировать модуль базы данных на языке программирования C++, содержащий информацию полученную со спутников.
- Разработать и запрограммировать модуль тестирования на языке программирования C++, производящий проверку результатов комплекса и результаты других исследователей.
- Разработать и запрограммировать дополнительные службы, заполняющие базу данных новой информацией.

Описание решения поставленных задач приведено в последующих главах данной работы.

Глава 6

Модуль базы данных

Для решения задачи создания модуля базы данных в рамках данной работы были выделены подзадачи перечисленные ниже:

- Создание базы данных наблюдений спутников DORIS
- Создание базы данных обработанных эфемерид DORIS
- Создание базы данных остальных параметров

6.1 База данных наблюдений спутников DORIS

Для работы программного комплекса в первую очередь необходима информация, накапливаемая антенной установленной на спутнике. Данные используются для уточнения параметров модели.

Данные получаемые системой спутников DORIS, хранятся на сервере [link на хранилище], в архивированных текстовых файлах, в формате RINEX 3.0 [link на документацию]. Получением и обновлением этих архивов занимаются службы, описанные в 6 главе.

Общая схема базы данных представленная на рис. 6.2. Далее приводится более подробное описание классов обозначенных в этой схеме.

DORIS data хранится в формате RINEX 3.0, поэтому необходимо в первую очередь создать обработчик этих данных, в формат который можно будет свободно использовать нужные параметры из этого файла. Обработчик назовем ParserHeaderRINEX, он будет создавать структуру данных, хранящую

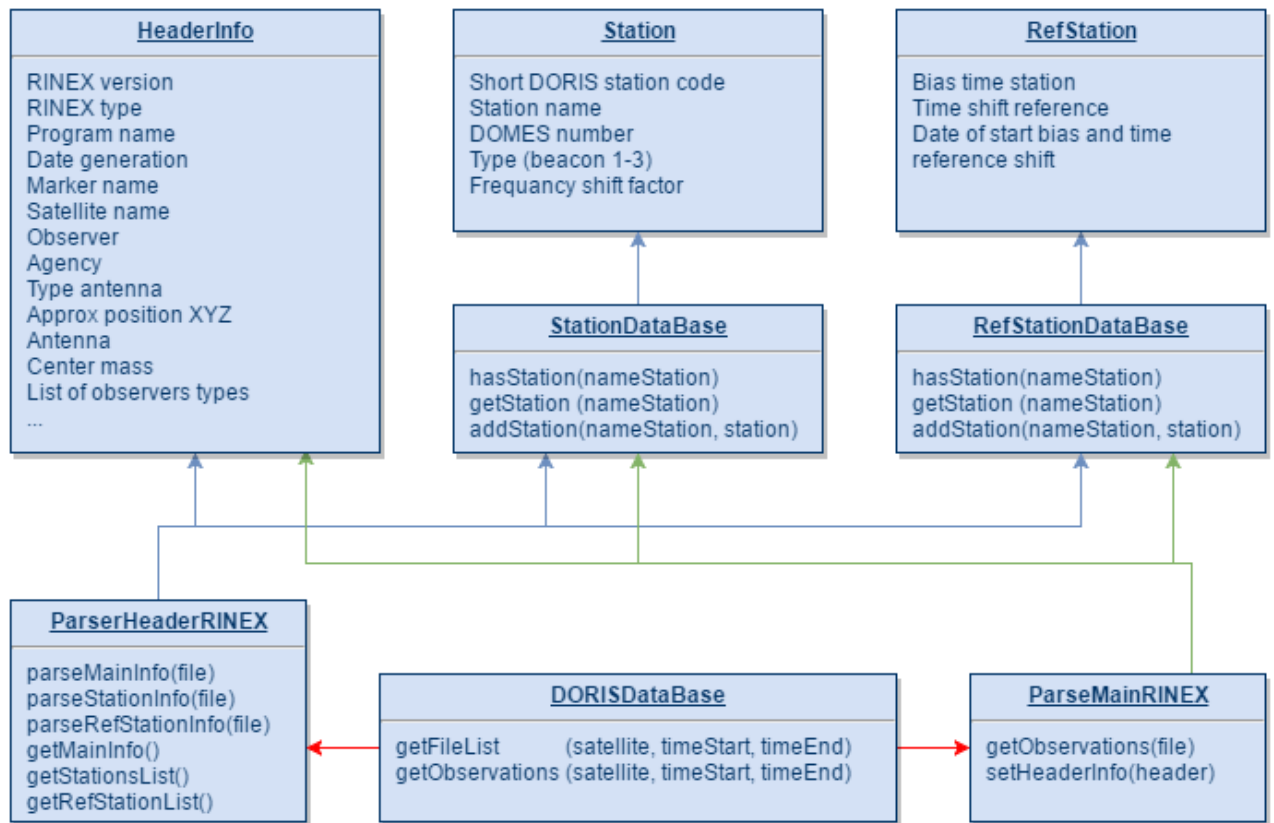


Рис. 6.1: Схема базы данных наблюдений

в себе все поля, перечисленные в заголовке. Заголовочный файл состоит из нескольких частей:

- Общая информация о спутнике
- Информация о станциях на земле
- Информация о станциях на земле, которые отвечают за синхронизацию.

Составные части программы:

- HeaderInfo - структура данных, содержащая общую информацию о спутнике
- Station - структура данных, содержащая общую информацию о станции на Земле (имя, тип маячка)
- RefStation - структура данных, содержащая дополнительную информацию о станции синхронизации (сдвиг по времени, производная сдвига в

секунду, момент времени от которого происходит отсчет сдвига и производной)

- `StationDataBase/RefStationDataBase` - класс, хранящий информацию о все станциях используемых в данной RINEX файле. Позволяет узнавать, получать, добавлять станции.
- `ParserHeaderRINEX` - вспомогательный класс, позволяющий получить всю информацию из заголовка RINEX файла.
- `ParseMainRINEX` - вспомогательный класс, который позволяет обработать информацию о наблюдениях из главной части RINEX файла.
- `DORISDataBase` - основной класс, который позволяет абстрагировать всю работу чтения и первичной обработки RINEX файлов, позволяет добавлять новые файлы для обработки и получать данные.

6.2 База данных эфемерид DORIS

Данные обработанные другими группами хранятся в формате SP3. Формат SP3 проще чем формат RINEX, поэтому для чтения таких файлов не надо использовать сложные методы. Из-за того, что данные хранятся на временной сетке с большим шагом, то требуется уметь приближать данные между узлами. Я использовал метод интерполяции для приближения этих значений, в качестве степени многочленов были использованы полиномы 8 степени. Обоснование того, что этой степени хватает и что лишняя информация не теряется будет приведено в следующей главе. Составные части модуля:

- `EphemerisDataBase` - основной класс, отвечающий за получения координат и скоростей. Между узлами используется приближение полиномом 8 степени.
- `ParserSP3` - класс, позволяющий получить изначальную сетку вычисленных значений эфемерид.

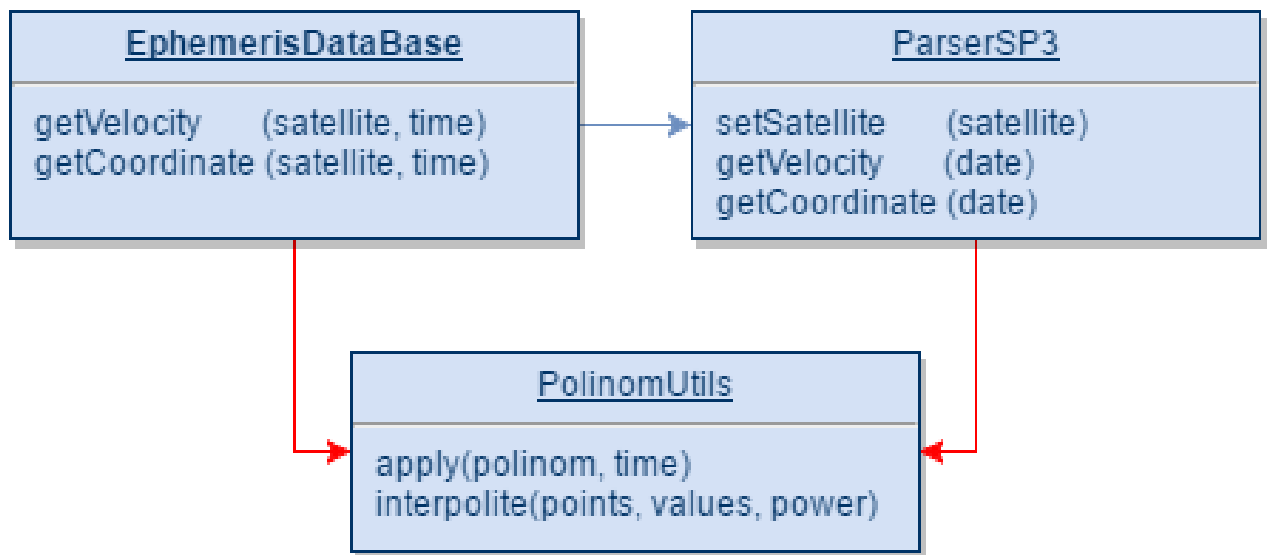


Рис. 6.2: Схема базы данных эфемерид

- PolinomUtils - вспомогательный класс, содержащий две функции, подстановку значения в полином и конструирование полинома от значений в точках.

6.3 Службы для работы с интернетом

Для систематического обновления данных требуется создать службу, которая сможет периодически получать данные из интернета. Так как данные лежат в архивах на ftp-сервере, а в языке программирования C++ работа с сетью реализована не очень удобным способом, то реализовать службу было бы удобнее используя другой язык программирования. На эту роль больше всего подходит Python. В нем работа с сетью интернет реализована проще всего.

Для обновления данных спутников DORIS требуется каждый день получать архив с новыми данными, каждый архив содержит в себе данные за сутки об одном спутнике. Из этого вытекает, что служба должна уметь получать не просто архив с данными, но и понимать, что за данные она получает (рис с данными DORIS).

Для обновления обработанных данных эфемерид спутников требуется создать другую отдельную службу, так как путь по которому хранятся данные сильно отличается (рис с данными sp3).

После получения архивов и загрузки их на жесткий диск, требуется распаковать эти архивы, так как тип архива является специализированным для Unix систем и алгоритм разорхивирования формата не реализован в библиотеках языка Python, то потребовалось использовать дополнительное программное обеспечение, чтобы получить содержимое архива. Выбор пал на программу 7z, которая позволяет вызывать себя из консоли системы Windows.

Схема работы:

- Каждая служба по таймеру проверяет сайт и получает последний архив.
- Используя внешнюю программу архивы превращаются в текстовые данные формата SP3 или RINEX.
- Файлы складываются в специальное место на диске, откуда программный комплекс умеет их доставать.

Дополнительно имеется возможность ручного получения данных, для этого необходимо запустить Python программу с аргументами, которые сообщат наблюдения какого спутника и на какую дату вы хотите получить.

Глава 7

Модуль тестирования

Глава 8

Заключение

В рамках данной работы были достигнуты результаты:

- Написан модуль базы данных эфемерид и наблюдений
- Созданы службы автоматического обновления данных
- Предложен и обоснован метод тестирования результатов

Приложение. Листинг программы

DownloadUtil.cpp

```
#include <cstdlib>
#include <string>
#include <sstream>

#include "DownloadUtils.h"

namespace DownloadUtils
{
void unzipAll(string satellite) {
std::stringstream ss;
ss << "./UtilsScripts/unzip.py" << " " << satellite;
system(ss.str().c_str());
}

void downloadRaw(string satellite) {
std::stringstream ss;
ss << "./UtilsScripts/rinex_downloader.py" << " " << satellite;
system(ss.str().c_str());
}

void downloadRaw(string satellite, int year) {
std::stringstream ss;
ss << "./UtilsScripts/rinex_downloader.py" << " " << satellite
    << " " << year;
system(ss.str().c_str());
}

void downloadRaw(string satellite, int year, int day) {
std::stringstream ss;
```

```

ss << "./UtilsScripts/rinex_downloader.py" << " " << satellite
    << " " << year
    << " " << day;
system(ss.str().c_str());
}

```

```

void downloadProcessed(string satellite) {
std::stringstream ss;
ss << "./UtilsScripts/sp3_downloader.py" << " " << satellite;
system(ss.str().c_str());
}

```

```

void downloadProcessed(string satellite, int year) {
std::stringstream ss;
ss << "./UtilsScripts/sp3_downloader.py" << " " << satellite
    << " " << year;
system(ss.str().c_str());
}

```

```

void downloadProcessed(string satellite, int year, int day) {
std::stringstream ss;
ss << "./UtilsScripts/sp3_downloader.py" << " " << satellite
    << " " << year
    << " " << day;
system(ss.str().c_str());
}
};

```

```
// DownloadUtil.h
```

```
#pragma once
```

```
namespace DownloadUtils
```

```

{
void downloadRaw(string satellite);
void downloadRaw(string satellite, int year);
void downloadRaw(string satellite, int year, int day);

void downloadProcessed(string satellite);
void downloadProcessed(string satellite, int year);
void downloadProcessed(string satellite, int yers, day);

void unzip(string satellite);
};

// PolinomUtils.h
#pragma once

#include "geometry"
#include <vector>

using geometry::Point3d;

namespace PolinomUtils
{
Point3d apply(const std::vector<Point3d>& polinom, double t);
std::vector<Point3d> interpolate(const std::vector<double>& points,
                                const std::vector<Point3d>& values);
};

// PolinomUtils.cpp
#include "PolinomUtils.h"

namespace PolinomUtils
{
Point3d apply(const std::vector<Point3d>& polinom, double t) {

```

```

Point3d result = {0.0, 0.0, 0.0};
for (int i = 0; i < polinom.size(); ++i) {
result.x = result.x * t + polinom[i].x;
result.y = result.y * t + polinom[i].y;
result.z = result.z * t + polinom[i].z;
}
return result;
}

{
    // вычисляем результирующий y
    double q = (x-MasX[0]) / step; // см. формулу
    double result = MasY[0]; // результат (y)

    double mult_q = 1; // произведение из q*(q-1)*(q-2)*...*(q-n)
    double fact = 1; // факториал

    for ( int i = 1; i < n; i++ )
    {
        fact *= i;
        mult_q *= (q-i+1);

        result += mult_q/fact * dy[i][0];
    }

    // освобождаем dy
    for ( int i = 0; i < n; i++ ) delete[] dy[i];
    delete[] dy;

    return result;
}

```



```

}
std::vector<Point3d> interpolate(const std::vector<double>& points,
                               const std::vector<Point3d>& values) {
std::vector<std::vector<Point3d> > grid(points.size());
for (int i = 0; i < grid.size(); ++i) {
grid[i].resize(points.size(), {0.0, 0.0, 0.0});
}

    for (int i = 0; i < n; i++) {
        grid[0][i].x = points[i].x;
        grid[0][i].y = points[i].y;
        grid[0][i].z = points[i].z;
    }

    for (int i = 1; i < n; i++) {
        for (int j = 0; j < n-i; j++) {
            grid[i][j].x = dgrid[i-1][j+1].x - grid[i-1][j].x;
            grid[i][j].y = dgrid[i-1][j+1].y - grid[i-1][j].y;
            grid[i][j].z = dgrid[i-1][j+1].z - grid[i-1][j].z;
        }
    }
return result;
}
};
// Geometry.h
#pragma once

namespace geometry
{
struct point3d {
double x, y, z;
};
};

```

```

// ParseSP3.cpp

#include "ParseSP3.h"
#include <string>

namespace ParseSP3
{
void read(std::ifstream &fin, std::vector<Point3d>& c,
         std::vector<Point3d>& v,
         std::vector<Time>& times) {
std::string s, name, tmp;
int year, month, day, hour, minute;
double second;
double px, py, pz;
double vx, vy, vz;

// header reading
while (s != "*") {
fin >> s;
}
// main part
while (s != "EOF") {
fin >> year >> month >> day >> hour >> minute >> second;
time::time_stamp time(year, month, day, hour, minute, second);
times.push_back(convert(time));

fin >> name;
fin >> px >> py >> pz >> tmp;
c.push_back(geometry::point3d(px, py, pz));

fin >> name;

```

```

fin >> vx >> vy >> vz >> tmp;
v.push_back(geometry::point3d(vx / 10.0, vy / 10.0, vz / 10.0));
fin >> s; // char "*"
}
}
};

// служба загрузки эфемерид
import os.path,urllib,sys

def download(path, name, end, end2, url):
    if (not os.path.isfile(path + name + end)):
        open(path + name + end, 'wb').write(urllib.urlopen(url + name + en

def exist(year, day):
def get_begin_year(s):
return 2000 + int(s[10:12])

def get_begin_day(s):
return int(s[12:15])

def get_end_year(s):
return 2000 + int(s[17:19])

def get_end_day(s):
return int(s[19:22])

for path in os.listdir("data/"):
if get_begin_year(path) <= year and get_begin_day(path) <= day and get_
return True
return False

```

```
# get needed year and day
year, days = tuple(int(x) for x in sys.argv[1:])
# check that file exist
if exist(year, days):
print("file exists")
else:
path = "data/"
name = "ssaja201.b14173.e14183.DGS.sp3.001"
url = "ftp://cddis.gsfc.nasa.gov/doris/products/orbits/ssa/ja2/"
download(path, name, ".Z", ".Z", url)
print("download complete")
```

Литература

- [1] Т.Н. Сотникова, К.В. Холшевников, М.С. Джазматиде.
К выводу уравнений движения в оскулирующих элементах.
Вестник Санкт-Петербургского университета, том 1(59), выпуск 2,
стр.340-344 (2014).
- [2] В.В. Витязев и др.
Небесные и земные координаты: Учебное пособие.
- СПб., Изд-во С.-Петербурде. ун-та, 2011. -304 с.
- [3] К.В. Холшевников, В.Б. Титов
Задача двух тел: Учебное пособие.
- СПб., Изд-во С.-Петербурде. ун-та, 2007. -180 с.
- [4] DORIS Tables of Data and Products [Электронный ресурс].
— Режим доступа:
<http://ids-doris.org/data-products/tables-of-data-products.html>
- [5] DORIS system [Электронный ресурс].
— Режим доступа:
<http://www.aviso.altimetry.fr/en/techniques/doris/doris-system/ground-beacons.html>
- [6] DORIS Documentation [Электронный ресурс].
— Режим доступа:
<http://ids-doris.org/report/documentation.html>
- [7] Paul R. Spofford et. al.
The National Geodetic Survey Standard GPS Format SP.
Dickerson, Maryland, USA.

- [8] G. Petit, B. Luzum.
IERS Conventions (2010).
Frankfurt am Main, 2010.
- [9] Paul R. Spofford et. al.
The National Geodetic Survey Standard GPS Format SP.
Dickerson, Maryland, USA.
- [10] Т.Н. Сотникова, К.В. Холшевников, М.С. Джазматни.
К выводу уравнений движения в оскулирующих элементах.
Вестник Санкт-Петербургского университета, том 1(59), выпуск 2,
стр.340-344 (2014).
- [11] В.В. Витязев с соавт.
Небесные и земные координаты: Учебное пособие.
- СПб., Изд-во С.-Петербур. ун-та, 2011. -304 с.
- [12] К.В. Холшевников, В.Б. Титов
Задача двух тел: Учебное пособие.
- СПб., Изд-во С.-Петербур. ун-та, 2007. -180 с.
- [13] В.В. Белецкий
Очерки о движении космических тел
— 2-е изд., перераб. и доп. — Москва: Наука, 1977. — 432 с.
- [14] Н.Г. Дубошин
Небесная механика. Основные задачи и методы
— 2-е изд., перераб. и доп.— Москва: Наука, 1968. — 799 с.
- [15] К. Мюррей, С. Дермотт
Динамика Солнечной системы, перевод под ред. И.И.Шевченко.
— Москва: Физматлит, 2009. — 588 с.
- [16] Мюррей, К., Дермотт, С. Динамика Солнечной системы / перевод под
ред. И.И.Шевченко. — Москва: Физматлит, — 2009. — 588 с.

- [17] C. Marchal.
The three-body problem — 1990.
- [11] E. Hairer, G. Wanner.
Analysis by Its History, Undergraduate Texts in Mathematics.
— New York, 1996.
- [8] J. Danby.
Fundamentals of Celestial Mechanics, 2nd Revised
& Enlarged Edition Hardcover, 1988.
- [18] А.А. Самарский. Теория разностных схем.
М: Наука, 1977