

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по учебной практике
Тема: Спутники

Студент гр. 3343



Коршков А.А.

Руководитель

Бахарев Д.С.

Санкт-Петербург

2025

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Коршков А. А.

Группа 3343

Тема практики: Спутники

Задание на практику:

Разработать приложение с GUI для отображения информации о спутниках на основе данных TLE.

Программа должна отобразить следующую статистику:

- Общее количество спутников, данные по которым загружены;
- Дата самых старых данных из загруженных;
- Разбитое по годам количество запущенных спутников;
- Разбитое по градусам (с точностью до одного градуса) количество спутников с разным наклоном орбиты.

В программе должна быть возможность сохранения полученной статистики в текстовый файл в произвольном виде.

Сроки прохождения практики: 25.06.2025 – 08.07.2025

Дата сдачи отчета: 08.07.2025

Дата защиты отчета: 08.07.2025

Студент



Коршков А.А.

Руководитель

Бахарев Д.С.

АННОТАЦИЯ

Целью практики является создание программы для отображения информации о спутниках на основе данных TLE. В ходе практики осваиваются технологии фреймворка QT, принципы построения GUI интерфейсов, умение работать с форматом TLE, извлекать из него необходимую информацию и язык программирования C++. В рамках практики создаётся интерфейс для окна с информацией о спутниках, с возможностью. Полученные навыки помогают понять принципы построения понятного интерфейса, и использовать в реальных проектах.

СОДЕРЖАНИЕ

Введение.....	5
1. Выполнение работы	7
1.1. Описание использованных технологий.....	7
1.2. Полученные результаты.....	8
Заключение	10
Список использованных источников	11
Приложение А. Снимки экрана программы.....	12
Приложение Б. Отзыв о прохождении практики.....	13

ВВЕДЕНИЕ

Цель практики – разработать на C++ с использованием Qt Widgets приложение для загрузки, синтаксического анализа (парсинга) и визуализации статистики по TLE-данным, закрепив навыки работы с GUI, сетевыми запросами и обработкой данных.

Задачи:

1. Написать парсер для файлов TLE формата
 - 1.1. Создать структуру для хранения записи об одном космическом объекте.
 - 1.2. Создать структуру, которая будет хранить необходимую обработанную информацию, в том числе и набор полученных объектов.
 - 1.3. Создать функции, реализующие загрузку и обработку информации из файлов. Применить язык регулярных выражений для проверки соответствия данных с шаблоном TLE формата.
 - 1.4. Реализовать загрузку и обработку информации из сетевого хранилища.
 - 1.5. Реализовать функцию проверки контрольной суммы для определения целостности данных.
2. Создать GUI для главного окна и окна с информацией о спутниках.
 - 2.1. Реализовать в главном окне необходимый функционал для загрузки из файлов или из сети.
 - 2.2. Информационное окно должно в себе содержать в понятном виде данные согласно заданию практики.
 - 2.3. Пользователь должен иметь возможность создать ещё окно с уже другими данными.
 - 2.4. Реализовать кнопки сохранения информации в файл и копирования в буфер обмена
3. Подключить парсер к логике работы GUI приложения
 - 3.1. В главном окне реализовать слоты, которые будут обрабатывать сигналы нажатия с кнопок и выполнять загрузку информации из файла или из сети.

3.2.В окне с информацией о спутниках добавить слоты, которые будут реализовывать логику сохранения информации в файл и копирования, а также сигналы, которые будут обращаться к функциям загрузки из сети или файла главного окна.

1. ВЫПОЛНЕНИЕ РАБОТЫ

1.1. Описание использованных технологий

В ходе выполнения учебной практики были использованы следующие технологии:

1. ОС Windows 11 и Fedora Linux

1.1. В ходе практики использовалась операционная система Windows 11 как основная ОС для создания и использования данной программы.

1.2. Также была использована операционная система Fedora Workstation 42 для создания и тестирования программы под семейство операционных систем на базе ядра Linux.

2. Фреймворк QT5/QT6

2.1. Фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++.

2.2. Для разработки программы был использован традиционный подход на основе Qt Widgets, который использует систему виджетов. Данный подход является очень проверенным для создания качественных настольных приложений, где в первую очередь важна функциональность.

2.3. Для создания интерфейсов использовались .ui файлы в формате XML, которые компилировались в C++, которые настраиваются с помощью «Сигналов-Слотов» и реализуют логику работы.

2.4. В концепции QT Widgets используется связка «Сигнал-Слот», где сигналом является какое-то действие в программе, а слот является перехватчиком данного сигнала и реализует необходимый функционал.

2.5. Основным для разработки использовалась версия QT 6.8.3 LTS, чтобы создать современный дизайн и использовать более современный стек технологий.

2.6. Также поддерживается QT 5.15.17 LTS для совместимости с более старыми версиями операционных систем и дистрибутивов.

3. Система сборки CMake

3.1.CMake — кроссплатформенное программное средство автоматизации сборки программного обеспечения из исходного кода. Он позволяет создать стандартные файлы для сборки, чтобы системные инструменты сборки скомпилировали программу под каждую ОС.

4. Язык программирования C++

4.1.Для разработки приложения с концепцией Qt Widgets необходимо использование языка C++, который позволяет точно работать с объектами, реализовывать логику окон и их внутренних объектов и форм, а также связывать сигналы и слоты между объектами.

5. Git

5.1.Git — распределенная система контроля версий, необходимая для совместной работы и управления исходным кодом.

5.2.Использовался конфигурационный файл .gitignore для исключения скомпилированных артефактов и локальных настроек IDE.

1.2. Полученные результаты

В результате прохождения учебной практики были получены следующие результаты:

- Реализована сборка приложения сборка под Windows 11 и Fedora Workstation 42 с помощью CMake.
- Создан адаптивный GUI на базе QtWidgets, автоматически подстраивающийся под размер окна:
 - ◆ Отображение общей статистики (число спутников, дата самой старой записи).
 - ◆ Таблицы с группировками по диапазонам наклона и по годам запуска.
 - ◆ Примеры интерфейса и скриншоты — в приложении А.
- Реализован надёжный парсер с валидацией формата через регулярные выражения и контрольными суммами, с распределением данных по диапазонам наклона и годам.
- Получен опыт в создании адаптивного интерфейса под разный размер экрана

- Исходный код находится в публичном репозитории Github:
<https://github.com/KorzikAlex/Satellites>
- Отзыв о прохождении учебной практики в приложении Б.

ЗАКЛЮЧЕНИЕ

В ходе прохождения практики были освоены основные принципы создания и построения графических приложений на базе фреймворка QT. Было написано приложение, которое позволяет получить TLE данные из файла или из сетевого хранилища и вывести необходимую информацию, с возможностью её сохранения в файл. Все поставленные учебные задачи выполнены в полном объеме. Полученные знания и навыки соответствуют современным требованиям к построению специализированных графических приложений и могут быть применены в дальнейшей профессиональной деятельности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. TLE — двухстрочный набор данных. Формат // Space-п URL: <https://spacepi.space/wiki/article/tle-dvustrochnyj-nabor-dannyh-format/> (дата обращения: 25.06.2025).
2. TLE // Википедия URL: <https://ru.wikipedia.org/wiki/TLE> (дата обращения: 25.06.2025).
3. TLE // Wikipedia (English Version) URL: <https://en.wikipedia.org/wiki/TLE> (дата обращения: 25.06.2025).
4. Qt Documentation // Qt Group URL: <https://doc.qt.io/> (дата обращения: 25.06.2025).
5. Main // Qt Wiki URL: <https://wiki.qt.io/> (дата обращения: 25.06.2025).
6. Руководство по языку программирования C++ // Metanit URL: <https://metanit.com/cpp/tutorial/> (дата обращения: 26.06.2025).
7. Руководство по программированию с помощью фреймворка Qt и языка C++ // Metanit URL: <https://metanit.com/cpp/qt/> (дата обращения: 26.06.2025).
8. Celestrak: Current GP Element Sets // Celestrak URL: <https://celestrak.org/NORAD/elements/> (дата обращения: 30.06.2025).
9. Regex101: build, test and debug // Regular expressions 101 URL: <https://regex101.com/> (дата обращения: 27.06.2025).
10. QML vs Qt Widgets – detailed comparison // Scythe Studio URL: <https://scythe-studio.com/en/blog/qml-vs-qt-widgets-detailed-comparison> (дата обращения: 25.06.2025).

ПРИЛОЖЕНИЕ А.
СНИМКИ ЭКРАНА ПРОГРАММЫ

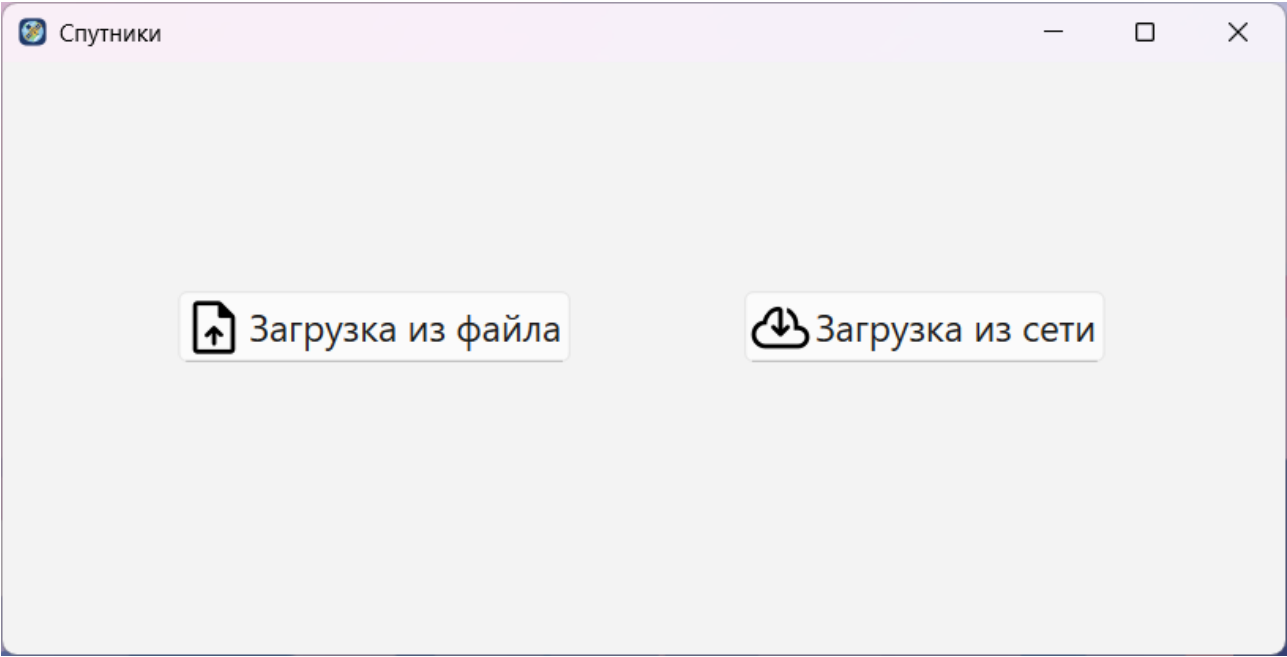


Рисунок 1 - Главное окно программы

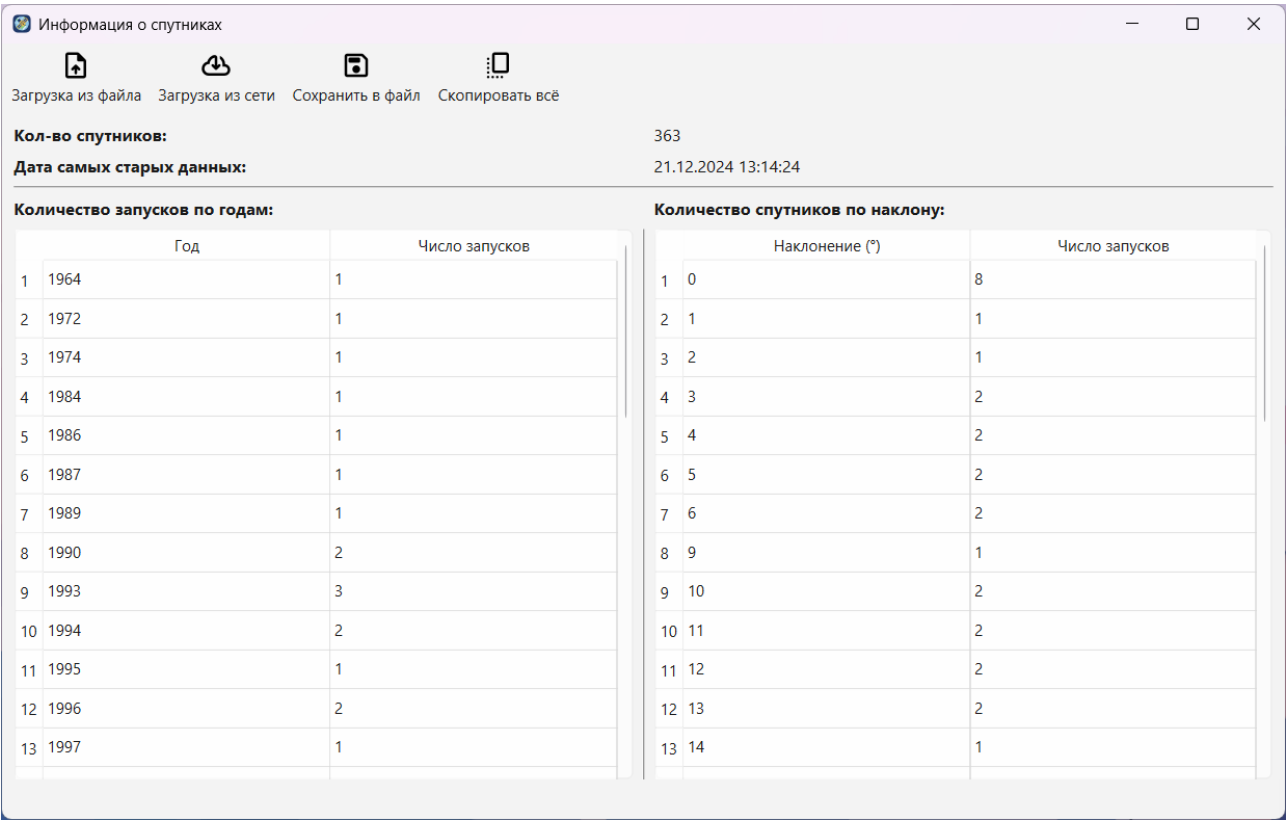


Рисунок 2 - Окно с информацией о спутниках

ПРИЛОЖЕНИЕ Б.
ОТЗЫВ О ПРОХОЖДЕНИИ ПРАКТИКИ