**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. Данные могут быть либо записаны в текстовый файл, либо находиться в интернете по заданному адресу.  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 наклонением 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 в TLE формате. 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, c возможностью сохранить информацию в файл или скопировать информацию в буфер обмена. Полученные навыки помогают понять принципы построения понятного интерфейса, работать с данными спутников, и использовать в реальных проектах.

**содержание**

|  |
| --- |
|  |

[Введение 5](#_Toc202556045)

[1. Выполнение работы 6](#_Toc202556046)

[1.1. Описание использованных технологий 6](#_Toc202556047)

[1.2. Полученные результаты 7](#_Toc202556048)

[Заключение 9](#_Toc202556049)

[Список использованных источников 10](#_Toc202556050)

[Приложение А. Снимки экрана программы 11](#_Toc202556051)

[Приложение Б. Отзыв о прохождении практики 12](#_Toc202556052)

# **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. Создать структуру для хранения записи об одном космическом объекте.
   2. Создать структуру, которая будет хранить необходимую обработанную информацию, в том числе и набор полученных объектов.
   3. Реализовать функции для загрузки и обработки информации из файлов и из сетевого хранилища. Применить язык регулярных выражений для проверки соответствия данных с шаблоном TLE формата.
   4. Реализовать функцию проверки контрольной суммы для определения целостности данных.
2. Создать GUI для главного окна и окна с информацией о спутниках.
   1. Реализовать в главном окне необходимый функционал для загрузки из файлов или из сети.
   2. Информационное окно должно в себе содержать в понятном виде данные согласно заданию практики.
   3. Реализовать кнопки сохранения информации в файл и копирования в буфер обмена.
3. Подключить парсер к логике работы GUI приложения
   1. В главном окне реализовать слоты, которые будут обрабатывать сигналы нажатия с кнопок и выполнять загрузку информации из файла или из сети.
   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 от 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. В ходе практики использовалась операционная система Windows 11 как основная ОС для создания программы и использование её на большинстве устройств.
   2. Также была использована операционная система Fedora Workstation 42 для создания программы под семейство операционных систем на базе ядра Linux.
2. Фреймворк Qt5/Qt6
   1. Фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++.
   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.
   3. Для создания интерфейсов использовались ui-файлы в формате XML, которые представляют собой шаблон для GUI, к которому подключается различная логика.
   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.
   5. Основной версией для разработки является Qt 6.8.3 LTS, чтобы создать современный дизайн и использовать более современный стек технологий.
   6. Также поддерживается сборка под Qt 5.15.17 LTS для совместимости с более старыми версиями операционных систем Windows и дистрибутивов на основе ядра Linux.
3. Система сборки CMake
   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++
   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
   1. Git – распределенная система контроля версий, необходимая для совместной работы и управления исходным кодом. С её помощью можно сохранять результат в удалённый репозиторий и переносить разработку и тестирование на другую платформу.

## 1.2. Полученные результаты

В результате прохождения учебной практики были получены следующие результаты:

* Реализована сборка приложения под Windows 11 и Fedora Workstation 42 с помощью CMake.
* Создан адаптивный GUI на базе QtWidgets, автоматически подстраивающийся под размер окна, в котором выводиться следующая информация
  + 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 контрольные суммы, с распределением данных по диапазонам наклона и годам.
* Получен опыт в создании адаптивного интерфейса под разный размер экрана.
* Исходный код программы 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-π 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 С++ // 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**

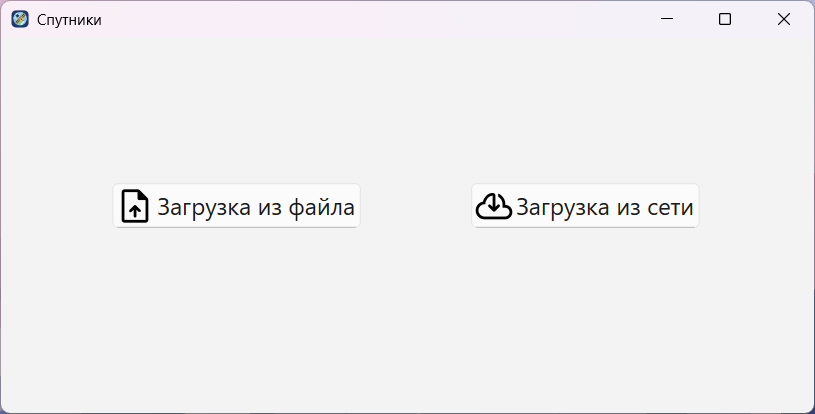


Рисунок 1 - Главное окно программы

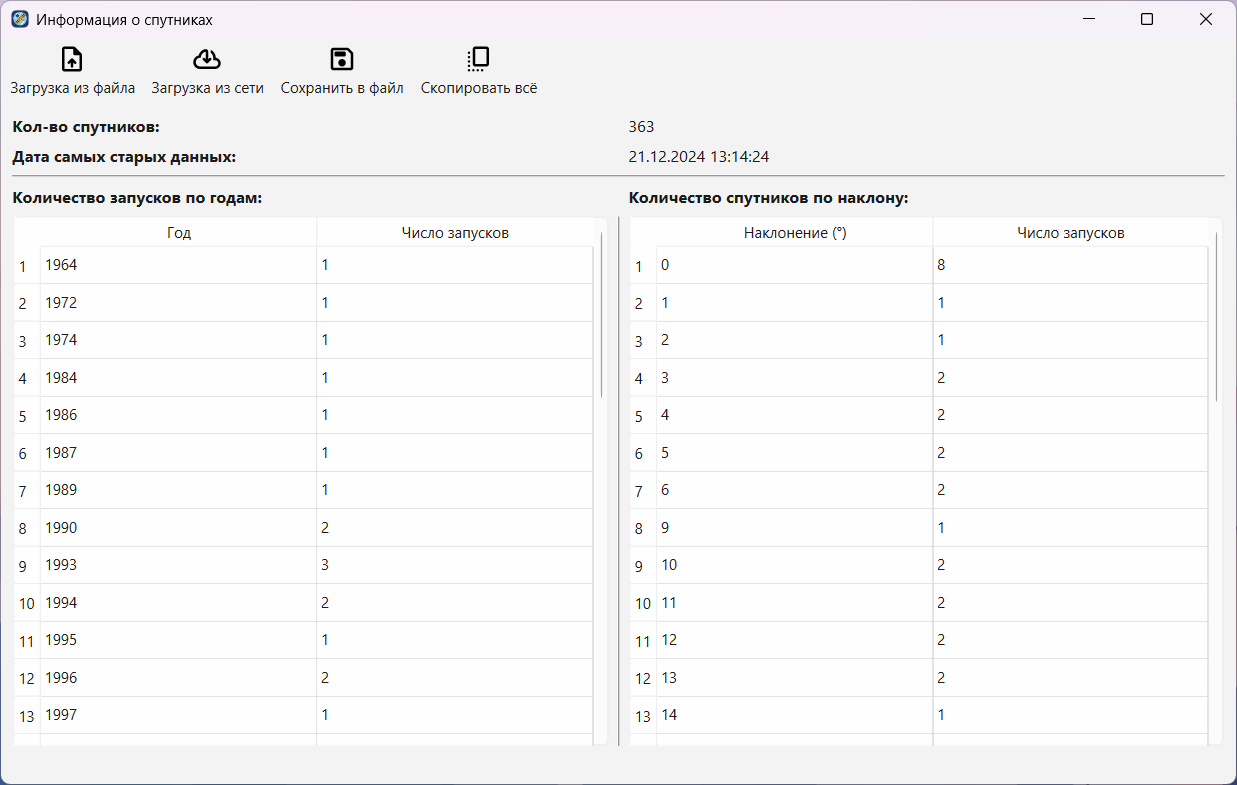


Рисунок 2 - Окно с информацией о спутниках

# **ПРИЛОЖЕНИЕ Б. ОТЗЫВ О ПРОХОЖДЕНИИ ПРАКТИКИ**

