## Муниципальное бюджетное общеобразовательное учреждение

## «Средняя общеобразовательная школа № 6 п. Новый Надеждинского района»

Итоговый проект

На тему: «Анализ оценок Сетевого Города при помощи программирования»

Выполнил: Гримме. М. А.

Ученик 9 “А” класс

Руководитель проекта:

Журавлёва О.В.

П. Новый 2023-2024 год

# Оглавление

1. Введение

* Актуальность темы
* Цель
* Задачи
* Методы
* Расшифровки

1. Основная часть
   1. Теоретическая часть

* Поиск и сбор информации
* Кратко об языке программирования Python
* Выбор и подготовка структуры проекта
* Выбор библиотеки для работы с Firefox
* Выбор библиотеки для работы с Telegram API
  1. Практическая часть

1. Установка Python
2. Установка библиотек
3. Создание репозитория на Github
4. Написание кода
   1. Код в parcer.py
   2. Создание СУБД и БД в DataBase.py
   3. Создание бота в BotFather
   4. Создание UI в tg\_bot.py
5. Тестирование
6. Заключение
7. Список использованной литературы

# Введение

## **Актуальность темы**

Активное внедрение информационно-коммуникационных технологий, является важным фактором создания системы образования, отвечающей требованиям информационного общества и процессу реформирования традиционной системы образования.

Информационные технологии сегодня играют важнейшую роль в обеспечении взаимодействий между людьми, а также практически во всех системах формирования, передачи и распространения информации.

Автоматизация рутинных задач с помощью информационных технологий может облегчить и ускорить повседневные задачи. Философия программирования как раз таки основана на автоматизации всяческих процессов и задач, с которыми легко может справится современный компьютер. А внедрение такой философии в образовательный процесс, способствует развитию системы образования и упрощению повседневных задач.

## **Цель**

Целью данного проекта является - упрощение, с последующей автоматизацией одной из рутинных задач в образовательном процессе, при помощи программирования и информационных технологий.

## **Задачи**

1. Сбор и поиск информации о взаимодействии браузера с языком программирования Python.
2. Подготовка, выбор и создание структуры программного проекта.
3. Создание программы в соответствии со структурой программного проекта
4. Тестирование программы
5. Компиляция программы

Под программным проектом имеется в виду, практическая часть проекта, а именно код написанный на выбранном языке программирования. Структурой же данного проекта является компьютерная папка со специализированными файлами.

## **Методы**

1. Программирование и написание скриптов на языке программирования Python
2. Сохранение прогресса программирования на открытую интернет платформу Github
3. Компьютер с установленной системой Windows для написания, тестирования создания структуры проекта
4. Интернет браузер Mozilla Firefox и Geckodrive для взаимодействия с языком программирования Python
5. Интегрированная среда разработки Visual Studio Code
6. Telegram API.
7. Свой опыт и знания в области программирования на Python

## 

## **Расшифровки**

API — описание способов взаимодействия одной компьютерной программы с другими.

Telegram — популярный интернет-мессенджер

ЯП — язык программирования

Python — один из языков программирования

Mozilla Firefox (или Firefox) — интернет браузер

Geckodrive — API между Python и Firefox

IT — (с англ.) Информационные Технологии

UI - (User Interface) с англ. Пользовательский интерфейс

GitHub - это веб-платформа для хостинга и управления проектами на основе системы контроля версий Git.

Git - это распределенная система контроля версий, позволяющая отслеживать изменения в файлах, управлять проектами и сотрудничать с другими разработчиками.

Библиотека — это набор связанных модулей, объединенных вместе для решения определенной задачи или предоставления определенного функционала. Библиотеки можно представить как расширение базового функционала Python, которое помогает разработчикам решать различные задачи без необходимости писать все с нуля.

# Основная часть

# Теоретическая часть

## **Поиск и сбор информации**

Для того чтобы начать проект, требовалось заполнить пробелы в знаниях и выбрать некоторые инструменты для работы. А также узнать как эти инструменты работают. Чтобы оптимизировать данную задачу, был составлен план сбора информации:

1. Язык Программирования Python - краткое описание языка и методов работы на нём
2. Кратко об Объектно-Ориентированном программировании и о его принципах
3. Структуры проектов - их виды и обоснования выбора
4. Визуализация структуры проекта
5. Библиотека Selenium - её документация и способы работ
6. Библиотека MatPlotLib - вывод графиков
7. Библиотека Aiogram - работа с Telegram API

**Язык Программирования Python**

Python — это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении (ML). Разработчики используют Python, потому что он эффективен, прост в изучении и работает на разных платформах. Программы на языке Python можно скачать бесплатно, они совместимы со всеми типами систем и повышают скорость разработки.

Платформы Python — это наборы пакетов и модулей. Модуль — это набор связанного кода, а пакет — это набор модулей. Разработчики могут использовать платформы Python для более быстрого создания приложений Python, поскольку им не нужно беспокоиться о низкоуровневых деталях (например, скорости обмена данных в веб-приложении) или том, как Python ускоряет работу программы. Python имеет два типа платформ:

* Платформа с полным стеком включает почти все, что требуется для создания крупного приложения.
* Микроплатформа – это базовая платформа, предоставляющая минимальные функциональные возможности для создания простых приложений Python. Также она предоставляет расширения, если приложениям требуются более сложные функции.

## 

## 

## **Объектно-ориентированное программирование**

Объектно-ориентированное программирование (ООП) — это подход, при котором программа рассматривается как набор объектов, взаимодействующих друг с другом. У каждого есть свойства и поведение. Если постараться объяснить простыми словами, то ООП ускоряет написание кода и делает его более читаемым.

Идеология объектно-ориентированного программирования (ООП) разрабатывалась, чтобы связать поведение определенного объекта с его классом. Людям проще воспринимать окружающий мир как объекты, которые поддаются определенной классификации (например, разделение на живую и неживую природу).

До ООП в разработке использовался другой подход — процедурный. Программа представляется в нем как набор процедур и функций — подпрограмм, которые выполняют определенный блок кода с нужными входящими данными. Процедурное программирование хорошо подходит для легких программ без сложной структуры. Но если блоки кода большие, а функций сотни, придется редактировать каждую из них, продумывать новую логику. В результате может образоваться много плохо читаемого, перемешанного кода.

В программировании объекты — это структуры данных: пользователь, кнопка, сообщение. У них, как и у реальных предметов, могут быть свойства: цвет, содержание или имя пользователя. А чтобы объединить между собой объекты с похожими свойствами, существуют классы.

Класс — это «шаблон» для объекта, который описывает его свойства. Несколько похожих между собой объектов, например профили разных пользователей, будут иметь одинаковую структуру, а значит, принадлежать к одному классу. Каждый объект — это экземпляр какого-нибудь класса.

## **Принципы ООП**

Абстракция

Абстрагирование — это способ выделить набор наиболее важных атрибутов и методов и исключить незначимые. Соответственно, абстракция — это использование всех таких характеристик для описания объекта. Важно представить объект минимальным набором полей и методов без ущерба для решаемой задачи.

Инкапсуляция

Каждый объект — независимая структура. Все, что ему нужно для работы, уже есть у него внутри. Если он пользуется какой-то переменной, она будет описана в теле объекта, а не снаружи в коде. Это делает объекты более гибкими. Даже если внешний код перепишут, логика работы не изменится.

Инкапсуляция помогает с легкостью управлять кодом. Выше мы сказали, что для обращения к объекту не нужно понимать, как работают его методы. Начальнику разработчика Ивана не обязательно знать, как именно он программирует: главное — чтобы выполнялись поставленные задачи.

Внутреннее устройство одного объекта закрыто от других: извне «видны» только значения атрибутов и результаты выполнения методов.

Наследование

Можно создавать классы и объекты, которые похожи друг на друга, но немного отличаются — имеют дополнительные атрибуты и методы. Более общее понятие в таком случае становится «родителем», а более специфичное и подробное — «наследником».

Полиморфизм

Полиморфизм (polymorphism) — это понятие из объектно-ориентированного программирования, которое позволяет разным сущностям выполнять одни и те же действия. При этом неважно, как эти сущности устроены внутри и чем они различаются.

## **Структуры проекта**

В программировании существует несколько основных видов архитектур проекта:

Монолитная архитектура:

* Программа разрабатывается как единое целое, где весь код обычно находится в одном приложении или кодовой базе.
* Проста в начале разработки, но может быть сложной для масштабирования и поддержки по мере роста проекта.

Клиент-серверная архитектура:

* Проект разделяется на клиентскую (пользовательскую) и серверную (серверную) части.
* Клиенты обращаются к серверу для получения данных или выполнения операций.

Многоуровневая архитектура:

* Проект разделен на несколько уровней (например, представление, бизнес-логика, доступ к данным), каждый из которых выполняет определенные функции.
* Обеспечивает легкость масштабирования и изменения различных частей проекта независимо друг от друга.

Микросервисная архитектура:

* Проект состоит из небольших, независимых сервисов, каждый из которых отвечает за определенную функциональность.
* Обеспечивает высокую гибкость и масштабируемость, но требует сложной инфраструктуры и управления.

Событийно-ориентированная архитектура:

* Проект построен вокруг обмена сообщениями между компонентами, которые реагируют на события или изменения состояния.
* Обеспечивает легкость интеграции новых компонентов и отслеживания изменений в системе.

Это основные виды архитектуры проекта в программировании, каждый из которых имеет свои преимущества и недостатки, и выбор зависит от конкретных требований и характеристик проекта.

## **Выбор структуры проекта**

После сравнения и анализа структур проекта, мною была выбрана многоуровневая структура проекта, так как она больше всего подходит для данного проекта.

Выбор многоуровневой структуры проекта обоснован несколькими преимуществами, которые она предоставляет:

Модульность и разделение ответственности:

Многоуровневая структура позволяет разделить проект на отдельные уровни, каждый из которых отвечает за конкретные аспекты приложения (например, представление, бизнес-логика, доступ к данным). Это способствует улучшению модульности кода и позволяет легко изменять или заменять отдельные компоненты без влияния на другие части проекта.

Упрощение тестирования и отладки:

Каждый уровень может быть протестирован и отлажен независимо от других уровней. Это облегчает обнаружение и исправление ошибок, а также обеспечивает более высокую степень надежности всего проекта.

Улучшение переиспользования кода:

Хорошо спроектированные компоненты на разных уровнях могут быть переиспользованы в различных частях проекта или даже в других проектах. Это способствует повышению производительности разработки и снижению затрат на создание нового функционала.

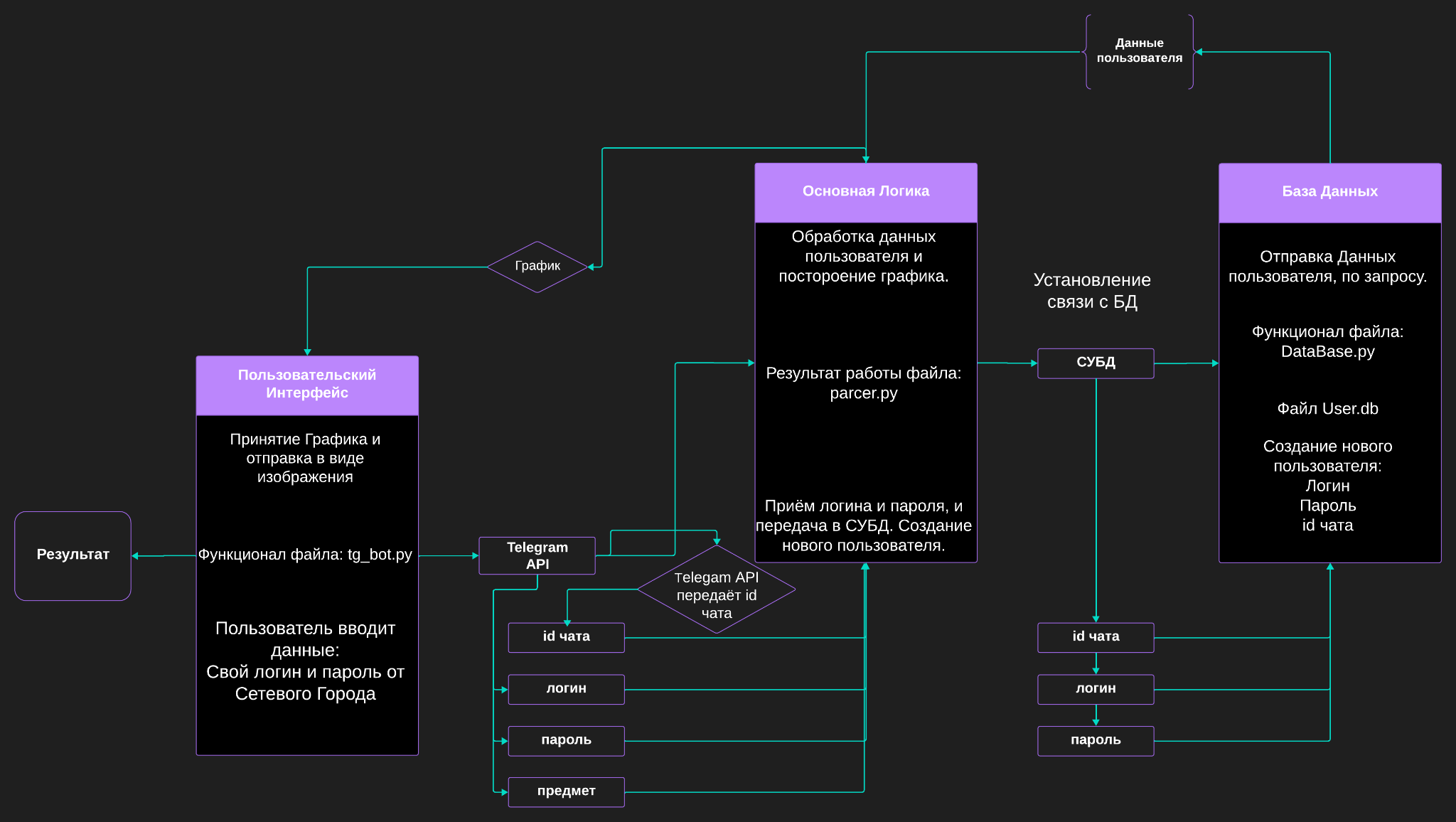
Снижение зависимости и повышение связности:

Многоуровневая структура позволяет уменьшить зависимость между различными компонентами проекта и повысить связность (связь) внутри каждого уровня. Это способствует созданию более гибкой и расширяемой архитектуры.

Исходя из этих преимуществ, многоуровневая структура проекта является хорошим выбором для моего проекта.

## **Визуализация структуры проекта**

После выбора структуры проекта, и для наглядной демонстрации работы кода, была составлена следующая блок-схема которая описывает работу программы:



*Первый уровень* проекта это - Пользовательский интерфейс (UI), который принимает от пользователя данные из сетевого города, после при помощью Telegram API, к ним добавляется id чата, и всё это дело передаётся во второй уровень.

*Второй уровень* обрабатывает запрос, и перенаправляет его в СУБД(которая является *третьим уровнем* проекта), которая создаёт нового пользователя в базе данных.

После этого, пользователю выдают сообщение о завершении регистрации.

Потом пользователь в первом уровне, выбирает предмет и четверть по которой он хочет построить график. Все эти данные первый уровень сообщает второму уровню, которые обрабатывает запрос, согласуясь при этом с третьим уровнем, и возвращает результат первому уровню в виде изображения графика, который первый уровень отправляет пользователю, либо выдаёт ошибку.

## 

## 

## **Библиотека Selenium и её выбор**

Выбор библиотеки Selenium на Python для парсинга сайтов можно обосновать следующими причинами:

Интерактивное взаимодействие с веб-страницами: Selenium предоставляет возможность автоматизированного взаимодействия с веб-страницами, включая клики по элементам, заполнение форм, отправку запросов и т. д. Это полезно для парсинга сайтов, где данные требуется получать через динамически генерируемый контент или после выполнения действий пользователя.

Поддержка различных браузеров: Selenium поддерживает различные браузеры, такие как Chrome, Firefox, Safari, Edge и другие. Это означает, что можно тестировать и парсить сайты в разных браузерах без необходимости изменения кода.

Множество возможностей: Библиотека Selenium предоставляет широкий спектр возможностей для управления веб-страницами, включая поиск элементов по CSS селекторам, XPath, работу с куки, обработку JavaScript и многое другое. Это делает её мощным инструментом для парсинга даже самых сложных веб-страниц.

Простота использования с Python: Selenium прекрасно интегрируется с Python и имеет удобный API. Пользователи Python могут легко создавать и запускать автоматизированные скрипты для парсинга сайтов с использованием Selenium.

Активное сообщество и документация: Selenium является одной из самых популярных библиотек для автоматизации веб-тестирования, что означает наличие обширного сообщества пользователей и множества ресурсов, таких как документация, форумы и учебные материалы, которые могут помочь в решении возникающих вопросов.

В целом, Selenium на Python является мощным инструментом для парсинга веб-сайтов, который обеспечивает гибкость, надежность и возможность автоматизации взаимодействия с веб-страницами.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **Библиотека MatPlotLib**

Использование библиотеки Matplotlib на Python предоставляет ряд значительных преимуществ для визуализации данных и создания графиков. Ниже перечислены основные преимущества:

Простота использования: Matplotlib предоставляет простой и интуитивно понятный интерфейс для создания графиков. Благодаря этому, даже новички могут быстро начать использовать библиотеку для визуализации своих данных.

Красивый дизайн графиков: Matplotlib предоставляет возможность настройки почти всех аспектов графиков, включая цвета, шрифты, маркеры, размеры и т. д. Это позволяет создавать профессионально выглядящие графики, которые эффективно передают информацию.

Интерактивность: В последних версиях Matplotlib добавлены возможности для создания интерактивных графиков, позволяющих пользователю взаимодействовать с данными непосредственно на графике. Это включает в себя возможность масштабирования, перемещения и подсветки данных.

Интеграция с другими библиотеками: Matplotlib хорошо интегрируется с другими библиотеками Python для анализа данных, такими как NumPy, Pandas и SciPy. Это позволяет создавать графики на основе данных, хранящихся в этих библиотеках, и легко комбинировать различные инструменты для анализа данных.

Большое сообщество и документация: Matplotlib - одна из самых популярных библиотек для визуализации данных на Python. Существует обширное сообщество пользователей и множество документации, примеров кода, учебных ресурсов и форумов поддержки, которые могут помочь в решении возникающих вопросов.

Matplotlib является мощным инструментом для создания высококачественных графиков и визуализации данных на Python.

## **Библиотека Aiogram**

Библиотека Aiogram представляет собой мощный инструмент для создания чат-ботов на платформе Telegram с использованием языка программирования Python. Её принципы работы и преимущества могут быть обоснованы следующим образом:

Асинхронность: Aiogram основана на асинхронном программировании с использованием asyncio в Python. Это означает, что чат-боты, созданные с помощью Aiogram, могут обрабатывать множество запросов одновременно и эффективно реагировать на события без блокировки потоков.

Простота использования: Aiogram предоставляет простой и понятный интерфейс для создания чат-ботов. Её API легко понять и использовать, что позволяет быстро разрабатывать и развертывать чат-ботов на платформе Telegram.

Многофункциональность: Aiogram обладает широким набором функций, позволяющих создавать разнообразные чат-боты. Она поддерживает отправку текстовых сообщений, медиафайлов, клавиатур, опросов, управление подписками и многое другое.

Гибкость и расширяемость: Aiogram предоставляет множество возможностей для настройки и расширения функциональности чат-ботов. Она поддерживает использование плагинов, обработчиков событий, и других механизмов, что позволяет адаптировать чат-бота под конкретные потребности проекта.

Открытый и активно развивающийся проект: Aiogram является открытым проектом с открытым исходным кодом, что позволяет разработчикам вносить свои вклады в развитие библиотеки и создавать свои собственные расширения. Благодаря активному сообществу поддержки, Aiogram постоянно обновляется и совершенствуется.

Обширная документация и примеры кода: У Aiogram есть обширная документация, содержащая подробные инструкции и примеры кода, что делает процесс обучения и использования библиотеки более простым и удобным для новых пользователей.

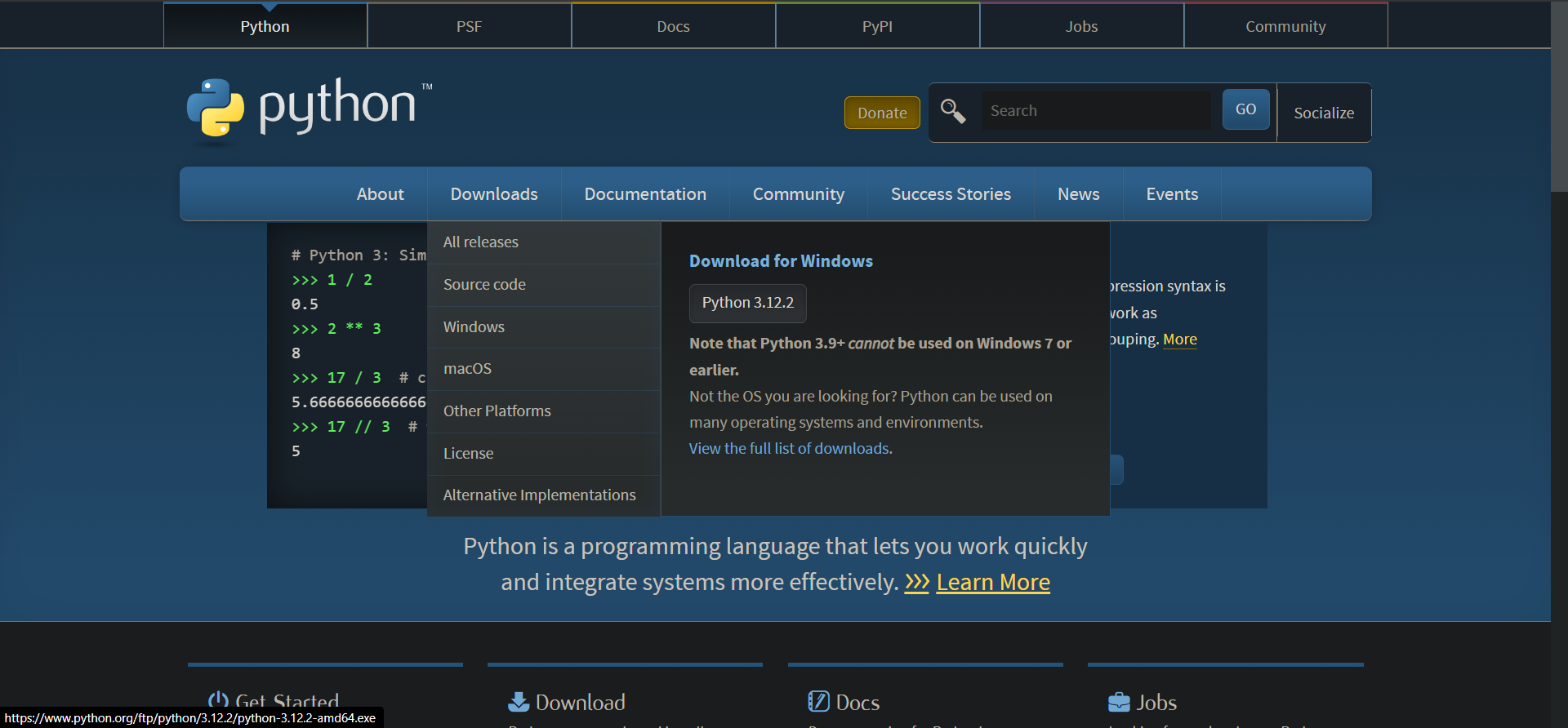
Библиотека Aiogram представляет собой удобный инструмент для создания чат-ботов на платформе Telegram на языке программирования Python, который обладает простотой использования, гибкостью и широким набором функций. И идеально подходящий для моего проекта.

# Практическая часть

## **Установка Python**

Установка языка программирования Python разделена на 2 этапа:

1. Скачивание установщика: Первый шаг - загрузить установщик Python с официального веб-сайта Python (https://www.python.org/). На сайте представлены версии Python для различных операционных систем.



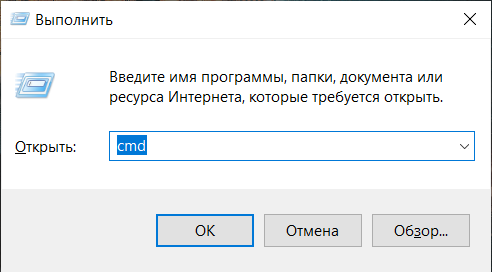
### сайт: https://www.python.org/

1. Запуск установщика: После загрузки установщика следует запустить его, дважды щелкнув по скачанному файлу. На Windows это обычно .exe файл.

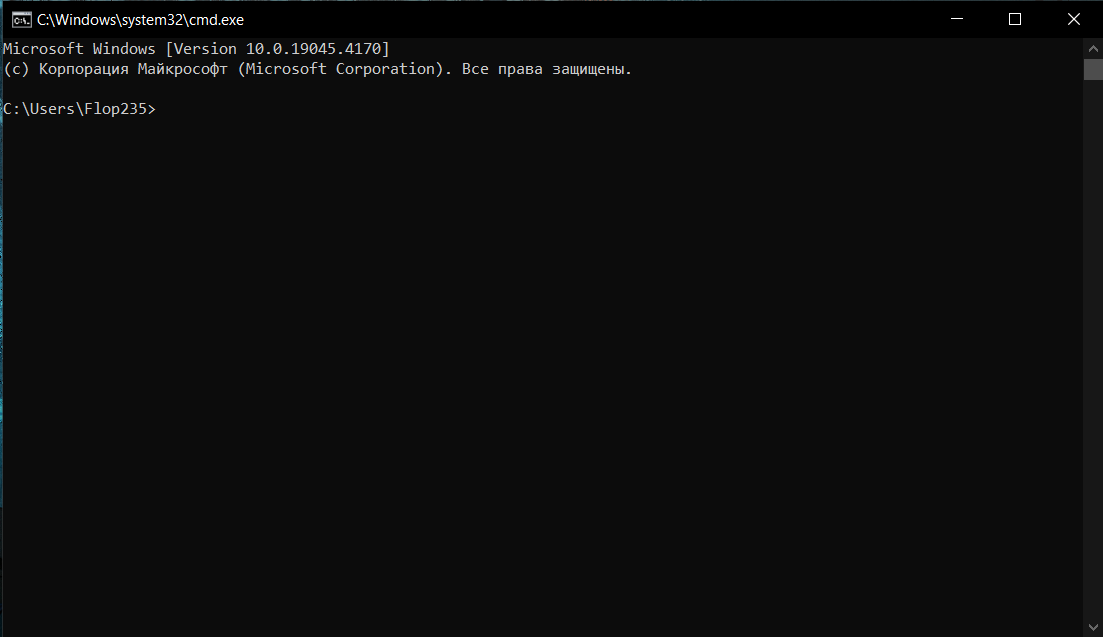
## **Установка библиотек**

Установка библиотек в языке программирования Python, осуществляется очень просто:

1. Открытие командной строки: Чтобы открыть командную оболочку(командную строку) в операционной системе Windows, требуется зажать сочетание клавиш *Win+R*, а в открывшемся окне прописать команду: *cmd*.



### Вызов командной оболочки в Windows

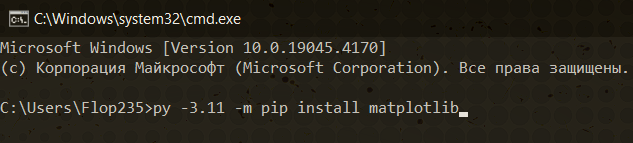


### Командная строка в Windows

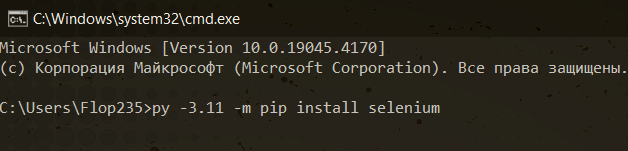
1. Ввод команды: Далее, в командной строке требуется ввести следующую команду:

**py -[версия ЯП Python] -m pip install [название библиотеки**]

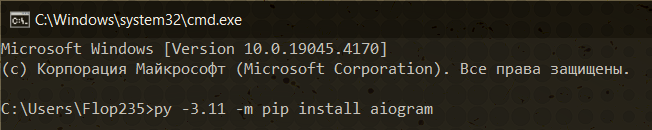
Команды для установки определённых библиотек:



### Команда для установки библиотеки matplotlib



### Команда для установки библиотеки selenium



### Команда для установки библиотеки aiogram

## **Создание Github репозитория**

Использование GitHub представляет собой важный инструмент для разработчиков и проектных групп в различных областях. GitHub предоставляет мощные средства для управления версиями кода. Это позволяет отслеживать изменения в коде, возвращаться к предыдущим версиям, создавать ветки для разработки новых функций и объединять их обратно в основную ветку. Тем самым он становится отличным дополнением к моему проекту и обеспечивает гарантию сохранности и читаемости кода.

1. Регистрация: для начала нужно зарегистрироватся на сайте: <https://github.com/>

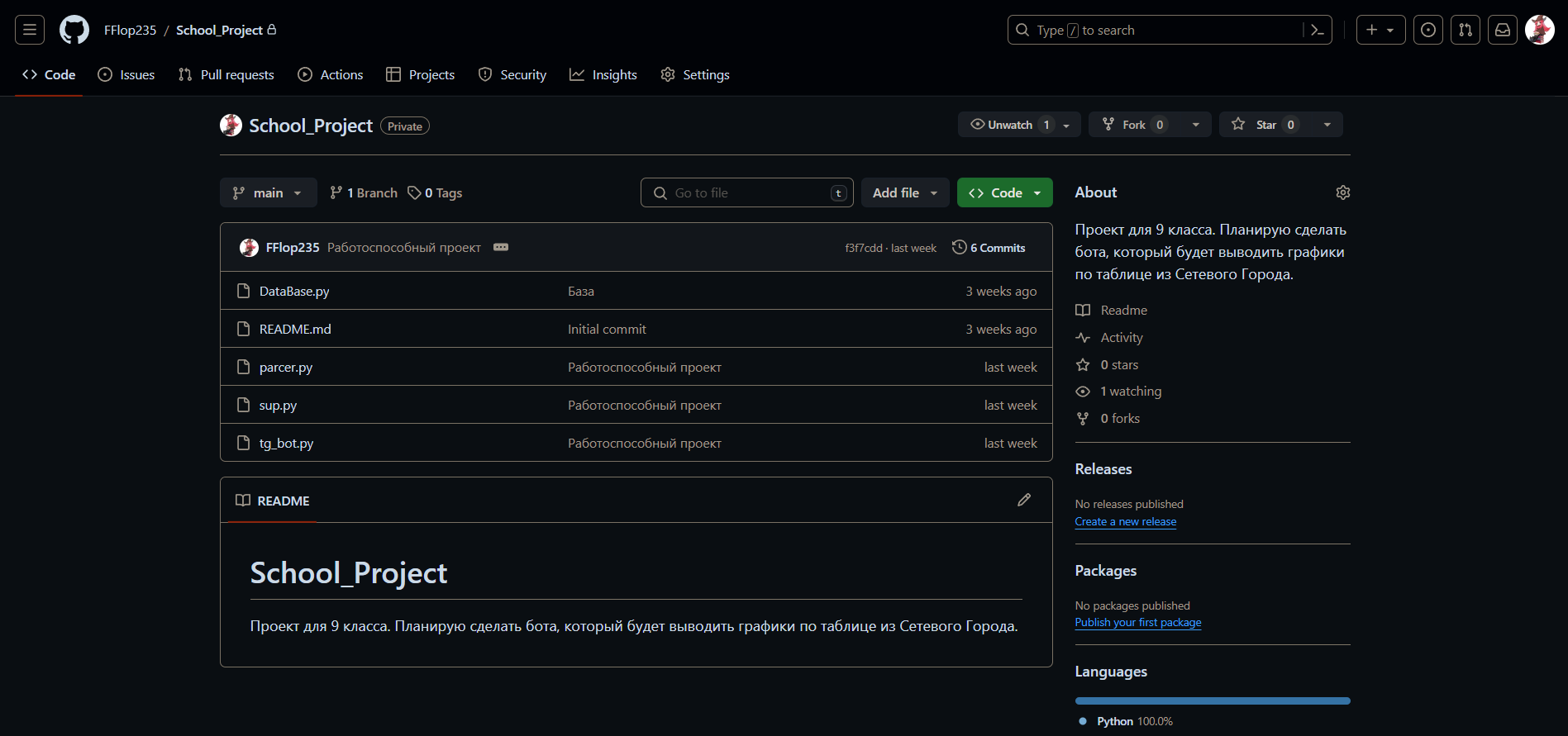
У меня уже есть аккаунт на Github, так что переходим к следующему шагу.

1. Создание репозитория: нужно создать отдельное место “репозиторий” на Github чтобы начать работу.



Выбираем название, нужные параметры и готово .

Так выглядит мой репозиторий:

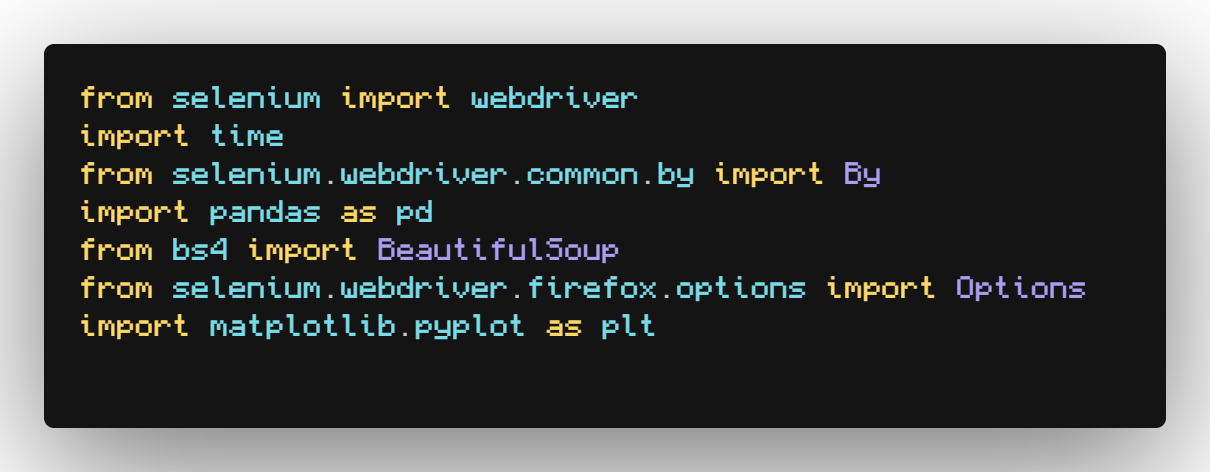


# **Написание кода**

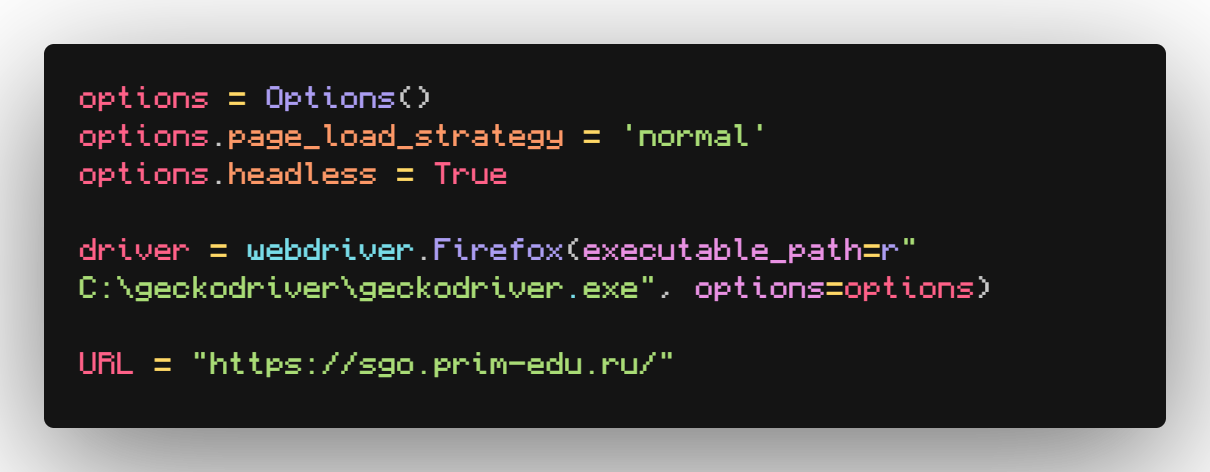
В данной главе будет преведены некоторые части кода с объяснением их работы. Объяснение будет во многом обобщенно и некоторый подробности будут опущены. Будут приведены изображения некоторых функций кода, а также их описание.

## **Вход в сетевой город и парсинг таблицы**

Первым в файле parcer.py по правилам программного этикета и здравого смысла описывается то, какие библиотеки мы будем использовать в данном файле:



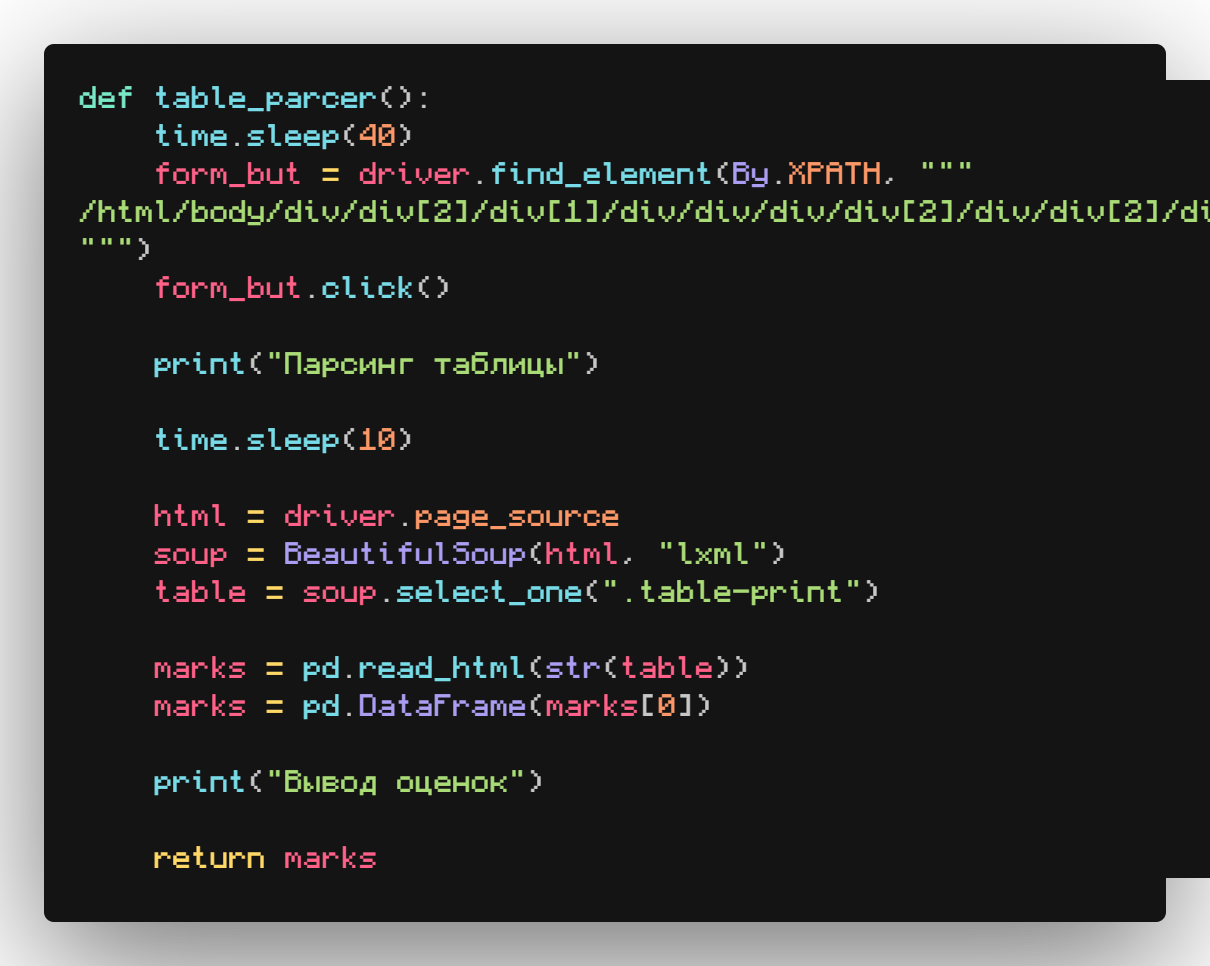
Затем идёт объявление глобальных переменных (переменных к которым будут обращаться из разных функций), в данном случае это переменная URL - ссылка на сетевой город, driver - экземпляр класса из библиотеки selenium (библиотека selenium будет отвечать за вход и управление действиями в сетевом городе)



Функция входа в сетевой город. Данная Функция будет вводить логин и пароль при авторизации:

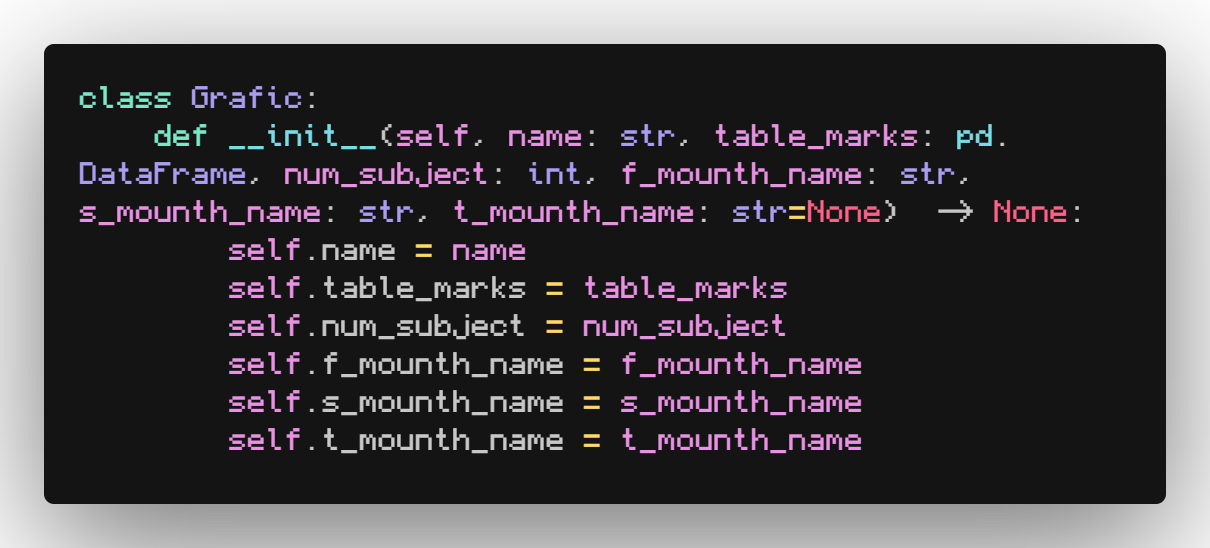


Далее от сетевого города требуется только получить таблицу с оценками, в мо1м проекте это реализовано так:



## **Построение графиков**

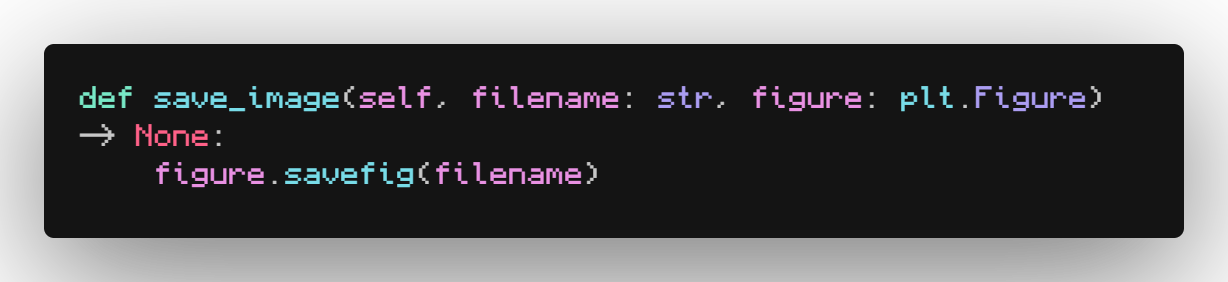
По полученной таблице требуется построить график, поэтому создаём отдельный класс для этого. Первым делом фильтруем таблицу от “УП”, “Н” и других “нечисел”. А потом по отфильтрованной таблице выводим график, который позже сохраняем.



### Инициализация класса Grafic



### Фильтрация таблицы

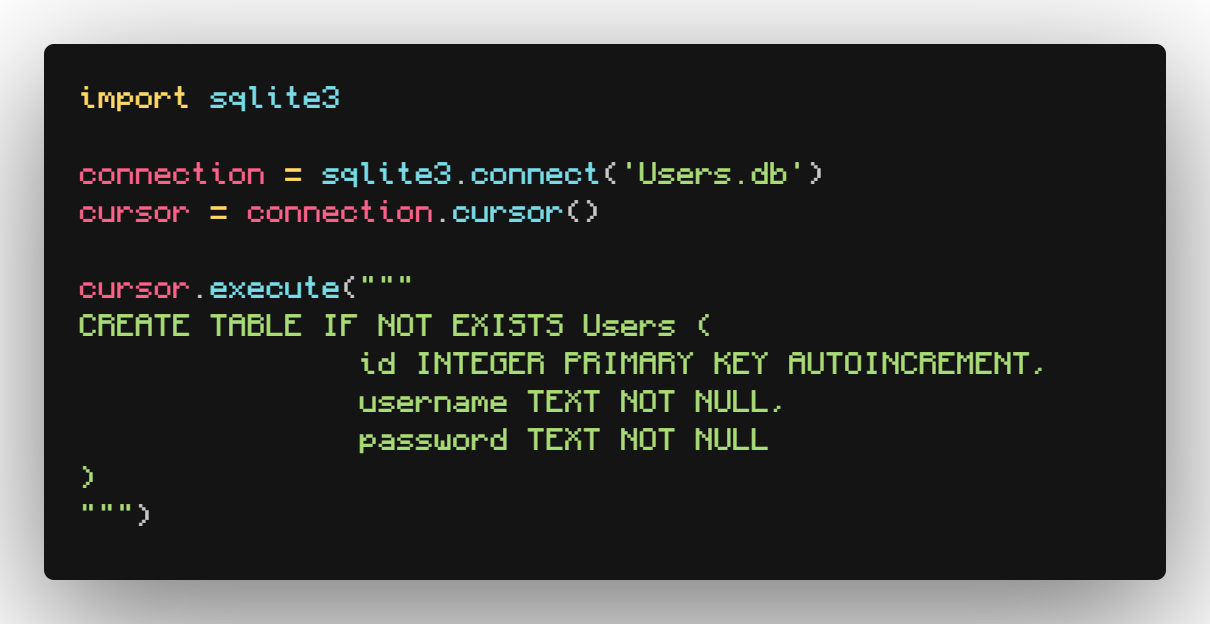
Сохранение изображения



### Создание графика

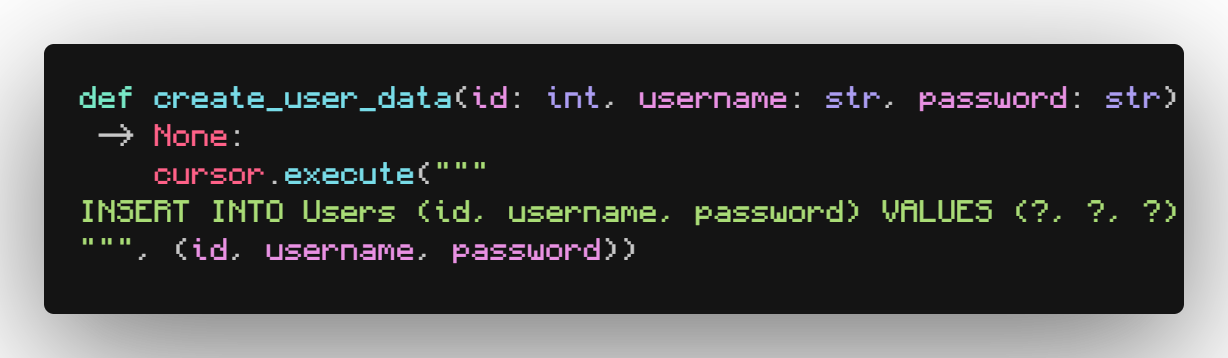
## **Создание СУБД и БД в DataBase.py**

Для работы с пользовательскими данными требуется база данных, и система управления ею (СУБД). С помощью библиотеки sqlite3 и языка программирования Python мы можем создать БД, и работать с ней. Делается это следующим образом:



### Создание базы данных с помощью ЯП Python

Теперь нам нужна функция добавления пользователя в БД, я реализовал её следующим образом:



На вход принимаются id, логин и пароль от сетевого, которые позже записываются в базу данных.

## **Создание бота в BotFather**

BotFather - это официальный бот от Telegram, который используется для создания и управления ботами на платформе Telegram. Основные возможности и функции BotFather:

Создание нового бота: BotFather позволяет создавать новых ботов, предоставляя им уникальные имена и токены доступа. Для этого пользователь может использовать команду /newbot и следовать инструкциям от BotFather.

Управление настройками бота: BotFather позволяет настраивать различные параметры и функции созданных ботов, такие как название бота, описание, изображение профиля и другие.

Получение токена доступа: После создания бота BotFather выдает токен доступа, который используется для аутентификации бота при взаимодействии с API Telegram.

Управление командами бота: BotFather позволяет настраивать пользовательские команды для бота. Пользователь может добавлять новые команды, удалять или изменять существующие.

Получение информации о боте: BotFather предоставляет различные сведения о созданных ботах, такие как их имя, токен доступа, список команд и другие параметры.

Поддержка и обновления: BotFather регулярно обновляется и поддерживается командой Telegram, что гарантирует его надежность и соответствие последним стандартам и требованиям.

BotFather является неотъемлемым инструментом для разработчиков, желающих создать и управлять ботами на платформе Telegram, и обеспечивает простой и удобный способ настройки и администрирования ботов.

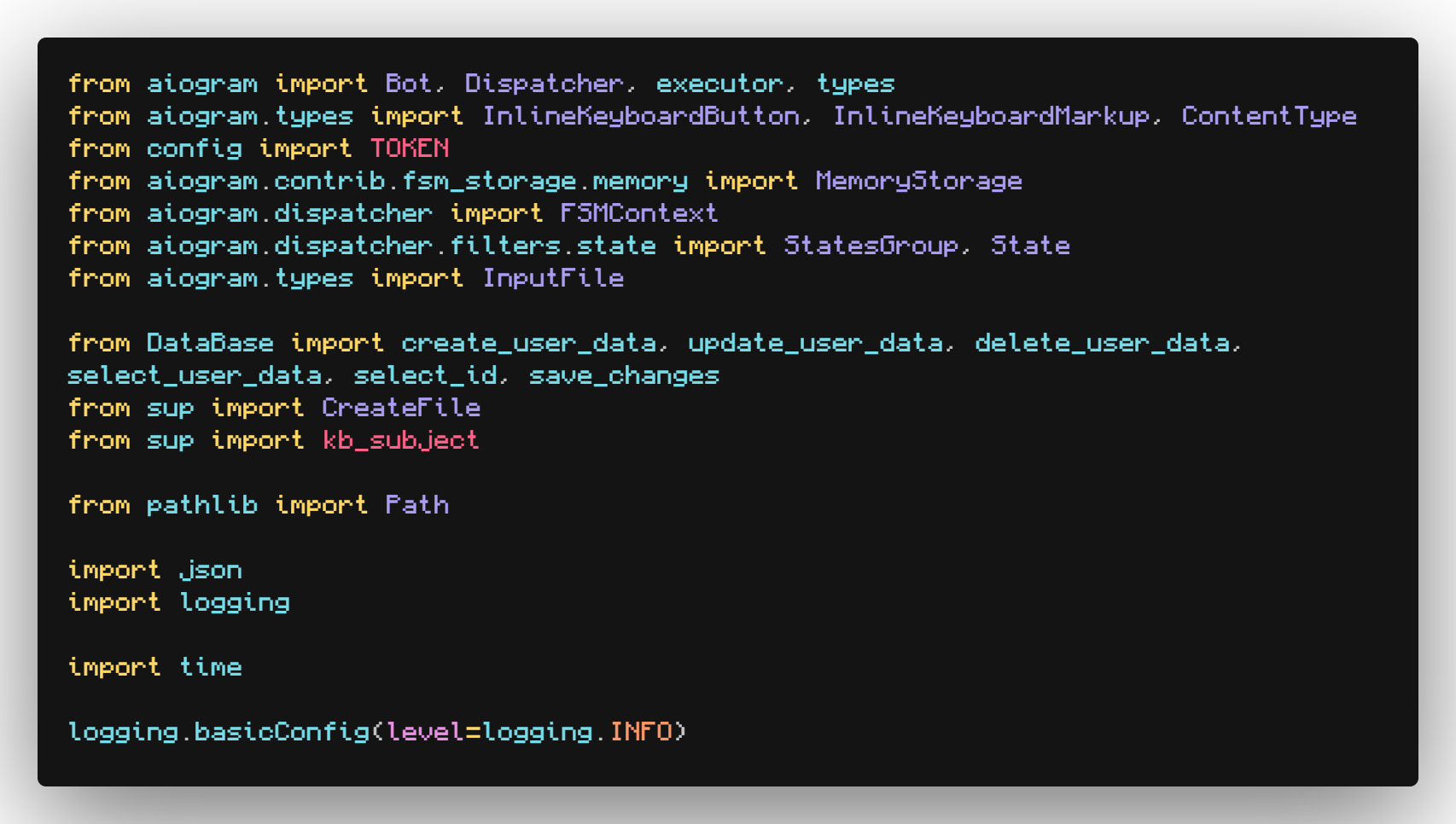
Создаём бота с помощью команды /newbot и копируем токен доступа, который записываем в файл config.py. Теперь всё готово для работы с Telegram.

## 

## **Создание UI (Telegram бота)**

После создания Data уровня и основной логики, требуется создать пользовательский интерфейс. Я выбрал использовать Telegram бота в качестве UI, так как у меня достаточно знаний и навыков для этого. Также Telegram является популярным мессенджером, который используют ~900 млн. человек.

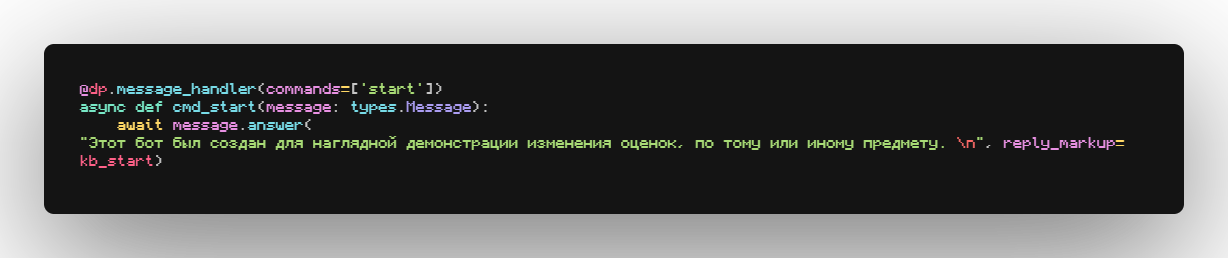
Первое что нужно сделать это импортировать библиотеки:



После импортирования библиотек, требуется создать глобальные переменные dp(dispatcher, диспетчер), bot(сокр. от слова Robot) и storage(с англ. Хранилище). Переменная dp в aiogram используется для организации обработки сообщений и управления поведением бота в соответствии с входящими событиями. Она является основным инструментом для разработки функционала бота и обеспечивает его корректное и эффективное взаимодействие с пользователями. Storage позволяет боту сохранять и управлять состояниями пользователей во время взаимодействия с ними. Например, если пользователь начинает диалог с ботом и вводит определенные данные, storage может сохранить эту информацию и использовать ее позже для персонализированного ответа или выполнения других действий.

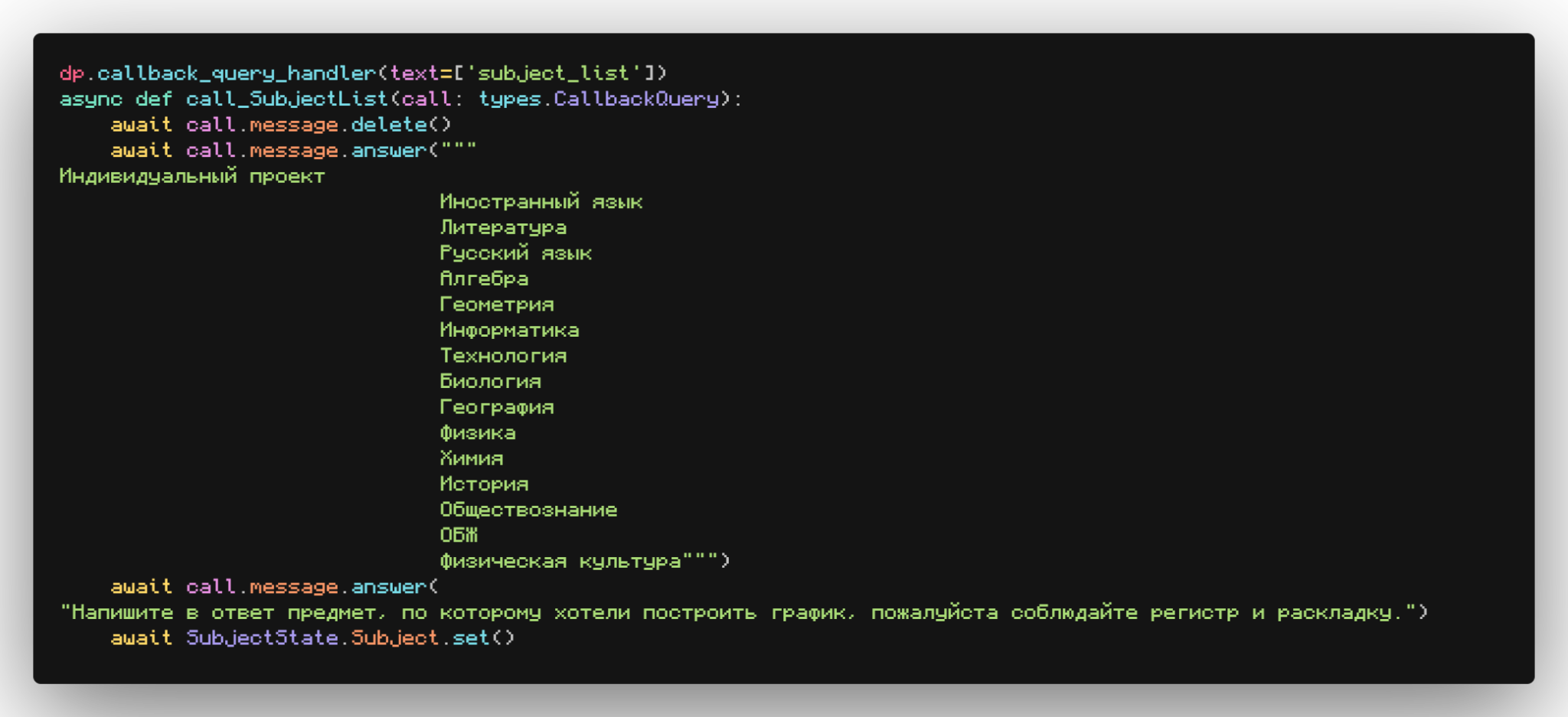


Далее следует обработать функцию старта бота, когда пользователь будет запускать нашего бота, вызовется функция cmd\_start(), в которой будет выводится сообщение о том для чего нужен бот, и какой у него функционал. После этого функция должна будет предлагать выбор пользователю, на регистрацию или на создание графика если пользователь уже зарегистрирован, а после перенаправлять его на регистрацию или иное действие. Для этого были созданы две кнопки, в которых реализован данный выбор.



### Функция cmd\_start()

После регистрации пользователя, бот должен вывести список предметов и спросить пользователя о том по какому предмету и какой четверти нужно вывести график. Для такой задачи, была реализована функция call Subject List(), реализована она следующим образом:



### Функция call\_SubjectList()

### 

## 

## 

## 

## 

## 

## 

## 

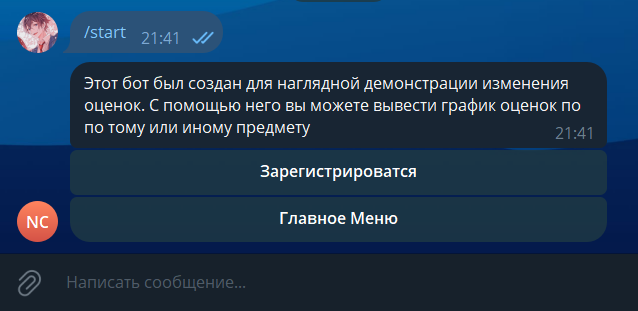
## 

## 

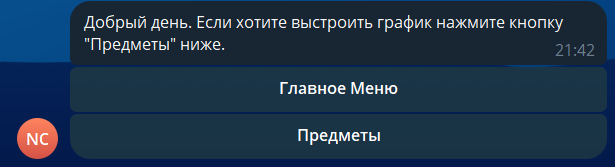
## **Тестирование**

Тестирование кода является неотъемлемой частью разработки программного обеспечения и играет важную роль в обеспечении его качества, надежности и производительности. Оно позволяет выявлять ошибки, улучшать качество кода и повышать доверие пользователей к программному продукту. Тестирование помогает выявить ошибки и дефекты в коде. Это позволяет разработчикам исправить проблемы на ранних стадиях разработки, что экономит время и ресурсы на исправление более серьезных проблем в будущем. Тестирование помогает улучшить качество программного обеспечения, так как оно позволяет проверить работоспособность и соответствие кода требованиям и спецификациям. Это способствует созданию более надежного и стабильного продукта. Хорошо протестированное программное обеспечение внушает доверие пользователей. Они чувствуют себя уверенно, используя приложение, которое прошло обширное тестирование и демонстрирует надежную работу.

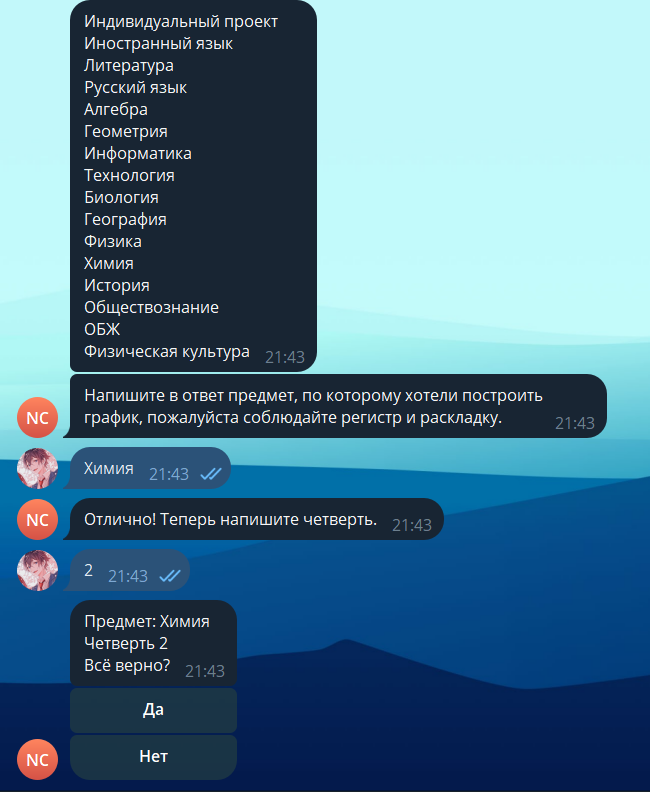
В моём проекте комплексное тестирование кода не нужно, так как он не особо большой. Но проверка работоспособности должно быть. Поэтому будут прикреплены изображения работы бота и вывода им графиков.



### Первое сообщение бота



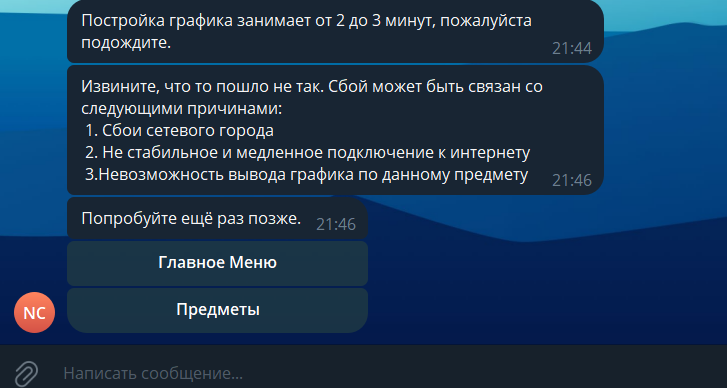
Главное меню



Список предметов и выбор



### Результат



Сообщение в случае ошибки

Как мы видим бот работоспособен и прекрасно выполняет свои задачи, а в случае ошибки выводит соответствующие сообщения с инструкциями.

# Заключение

Цель в виде упрощения одного из образовательных процессов посредством автоматизации при помощи интернет технологий и программирования, достигнута. Написана программа на языке программирования Python, которая выполняет функции визуализации в виде графика, и автоматизации процесса представления изменения оценок, и быстрому доступу к информации. Когда пользователь заходит в Сетевой Город, ему придётся потратить около 5-6 минут только на вход и получение необработанной таблицы с оценками, а визуализировать данные он не сможет. Моя программа же автоматизирует вход в Сетевой Город и строит графики изменения оценок по всем предметам, тем самым упрощая и ускоряя доступ к информации, что приводит к оптимизации такого образовательного процесса как контроль за успеваемостью. В ходе изучения документации по библиотекам python, создание архитектуры проекта, написание программы, цель проекта была достигнута. Данный проект наглядно показывает как интернет технологии могут быть полезны и даже необходимы для процесса образования. Правильное внедрение их в образовательную систему, способствует улучшению не только качества образования, за счёт быстрого контроля над успеваемостью, но и стимулированию интереса к интернет технологиям и программированию у участников образовательного процесса. Такое внедрение будет одним из ключевых факторов для развитие технологий и экономического состояния страны в будущем, так как такие технологии используются для автоматизации разных задач не только в образовательных процессах но и в экономических, технических, научных, социологических и т.д. Автоматизация - это большая экономия ресурсов и времени для компаний или образовательных организаций. Данные проект - наглядные пример автоматизации какой либо задачи в образовательном процессе, для экономии времени и некоторых ресурсов. Она была достигнута путём создания программы при помощи IT технологий, таких как язык программирования и др. Это и показывает их дальнейшую необходимость в будущем.

# Список использованной литературы

1. Шевко Н.Р., Турутина Е.Э. - Внедрение информационных технологий в образование: достаточность или необходимость? -<https://cyberleninka.ru/article/n/vnedrenie-informatsionnyh-tehnologiy-v-obrazovanie-dostatochnost-ili-neobhodimost>

2. Software Freedom Conservancy - Selenium документация библиотеки

<https://www.selenium.dev/documentation/>

3. aiogram Team with Sphinx and @pradyunsg's Furo - Aiogram документация библиотеки

<https://docs.aiogram.dev/en/latest/>

4. Задорожный С.С., Фадеев Е. П. - Объектно-ориентированное программирование на языке Python

<https://cmp.phys.msu.su/sites/default/files/%D0%9E%D0%9E%D0%9F_%D0%BD%D0%B0_Python_%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D0%BE%D0%B5%20%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%B8%D0%B5_var7.pdf>

5. Власова Е.А., Новожилова О.Е. - Внедрение современных технологий в образовательный процесс

<https://cyberleninka.ru/article/n/vnedrenie-sovremennyh-tehnologiy-v-obrazovatelnyy-protsess>

6. Иван Карманов @ivankarmanow - За границей Hello World: полный гайд по разработке Telegram ботов с помощью Python и Aiogram 3.

<https://habr.com/ru/articles/732136/>

7. gwaap (Пользователь форума zelenka) - Aiogram. Разбираем FSM простыми словами

<https://zelenka.guru/threads/3769612/>

8. pandas via NumFOCUS, Inc. Hosted by OVHcloud. Created using Sphinx 7.2.6. - документация библиотеки pandas

<https://pandas.pydata.org/docs/>

9. John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team; 2012–2024 The Matplotlib development team. Created using Sphinx 7.2.6. - документация библиотеки matplotlib

<https://matplotlib.org/stable/index.html>

10. Python Software Foundation. - Документация по языку программирования Python

https://docs.python.org/3/