МИНИСТЕРСТВО ПРОСВЯЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Муниципальное бюджетное общеобразовательное учреждение

«Средняя общеобразовательная школа №64 г. Владивостока»

Использование функционала чат-ботов, как составной части

цифровой образовательной среды образовательной организации

Автор работы:

ученик 10 класса Глухова В.Д.

Научный руководитель:

учитель информатики МБОУ «СОШ №64» Руденко Е.О.

г. Владивосток

2023

Оглавление

[Введение 3](#_Toc130128991)

[1. Теоретическая часть 4](#_Toc130128992)

[1.1. Telegram, как один из самых популярных мессенджеров, реализующих поддержку ботов 4](#_Toc130128993)

[1.2. Python как инструмент гибкой настройки чат-ботов 5](#_Toc130128994)

[1.3. Использование библиотек с целью ускорения процесса разборки 5](#_Toc130128995)

[2. Практическая часть 7](#_Toc130128996)

[2.1. Подготовительный этап, создание эхо-бота 7](#_Toc130128997)

[2.2. Реализация основной логики чат-бота 9](#_Toc130128998)

[2.3. Настройка поставки информации о расписании уроков из информационной системы образовательной организации 11](#_Toc130128999)

[Заключение 13](#_Toc130129000)

[Источники 14](#_Toc130129001)

# Введение

Все чаще чат-боты применяются для решения бытовых задач человека. Чат-боты – это небольшие приложения, которые полностью работают в приложении Telegram. Пользователи взаимодействуют с ботами через гибкие интерфейсы, которые могут поддерживать любые задачи или услуги. Такие боты можно настраивать с помощью языков программирования [1].

Организации доступа к актуальному расписанию уроков используя удобный и привычный канал для обучающихся позволит оперативно узнать об изменении в расписании и эффективно планировать учебную деятельность.

Целью проекта является создание бота для своевременной доставки информации о расписании уроков, удобного и привычного обучающимся.

Задачами проекта являются:

1. изучить теоретические и практические аспекты предметной области;
2. создать инструмент, позволяющий быстро получать расписание уроков.
3. организовать автоматическую поставку данных из информационной системы образовательной организации.

# Теоретическая часть

Социальные сети стали частью нашей повседневной жизни, уже сложно найти человека, который бы не применял мессенджеры для общения или решения своих бытовых задач. Одним из самых популярных приложений, который входит в топ 5 скачиваемых бесплатных приложений в мире, является Telegram [4].

## Telegram, как один из самых популярных мессенджеров, реализующих поддержку ботов

Telegram – это приложение для обмена сообщениями, ориентированное на скорость и безопасность. С помощью Telegram можно отправлять сообщение, фотографии, видео и файлы любых форматов. Так же там можно создавать группы или каналы для трансляции информации на неограниченную аудиторию. Неотъемлемой частью Telegam так же являются боты [5].

Боты Telegram – это специальные учетные записи, для настройки которых не требуется дополнительный номер телефона. Эти учетные записи служат интерфейсом для кода, запущенного на сервере. Боты способны обрабатывать входные данные и запросы способами, недоступными учетным записям пользователей, но есть несколько различий между ботом и обычным пользователем. У ботов нет статусов "последний раз видели" или "онлайн" – вместо этого они показывают ярлык "бот" в чате. У них ограниченное облачное хранилище – старые сообщения могут быть удалены сервером вскоре после их обработки. Так же боты не могут первыми начинать разговоры с пользователями. Пользователь должен либо добавить их в группу, либо сначала отправить им сообщение. По умолчанию боты, добавленные в группы, видят только соответствующие сообщения в чате. И наконец боты никогда не едят, не спят и не жалуются (если специально не запрограммировано иное).

Логику поведения ботов можно описывать, используя специализированные конструкторы или языки программирования. Для того, чтобы установить связь Telegram-бота и программы, моделирующей работу чат-бота используется Telegram API.

API (Application Programming Interface) – это программный интерфейс приложения, набор способов и правил, по которым различные программы общаются между собой и обмениваются данными. Все эти взаимодействия происходят с помощью функций, классов, методов, структур, а иногда констант одной программы, к которой обращаются другие) [6].

## Python как инструмент гибкой настройки чат-ботов

Для гибкой настройки работоспособности бота следует использовать языки программирования, например, 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.

К преимуществам Python относят легкую читаемость, поскольку язык имеет базовый синтаксис, похожий на синтаксис английского языка. Так же Python можно легко сочетать с другими языками программирования, например, Java, C, C++. В интернете доступно множество ресурсов для изучения Python, что каждый может научиться писать на языке самостоятельно.

## Использование библиотек с целью ускорения процесса разборки

Библиотеки – сборники модулей и различных компонентов, которые облегчают процесс создания новой программы. В зависимости от поставленной задачи, применяются различные комплексы и элементы кода. Современные разработчики должны использовать библиотеки, так они сокращают количество ошибок [7].

При написании софта, существует множество стандартных элементов: кнопки, поле с поиском, проигрыватели. Разработчику нет необходимости отдельно расписывать каждый элемент с нуля. При подключении модулей, часть кода переносится в новую программу. В зависимости от языка программирования, различают количество библиотек. Чем их больше, тем популярнее язык.

Современный софт написан большим количеством кода. Использование стандартных элементов с нуля увеличивает время создания программы. Также если разработчик не использует отдельные компоненты, код программы становится объёмным и массивным. К преимуществам использования библиотек относят экономию времени, структурирование информации, облегчение разработки и уменьшение количества ошибок в коде.

# Практическая часть

## Подготовительный этап, создание эхо-бота

Для непосредственного создания Telegram-бота в первую очередь необходимо воспользоваться другим ботом, который помогает создавать собственные «приложения» «@BotFather».

Этот бот выдает токен на нового, полностью настраиваемого в этом же чате бота.

Токен – это его уникальный идентификатор, предоставляющий полный контроль над ботом, именно поэтому при создании бота его лучше хранить в отдельном файле, чтобы никто не возымел контроль над вашим ботом.

Получив токен и настроив описание бота через «@BotFather» стоит сначала проверить его на работоспособность, а для этого можно написать простого эхо-бота, который будет как попугай повторять все отправленные пользователем сообщения [8].

Первым делом стоит записать токен, полученный от ранее упомянутого бота, в отдельный файл, который мы назовем “config”.

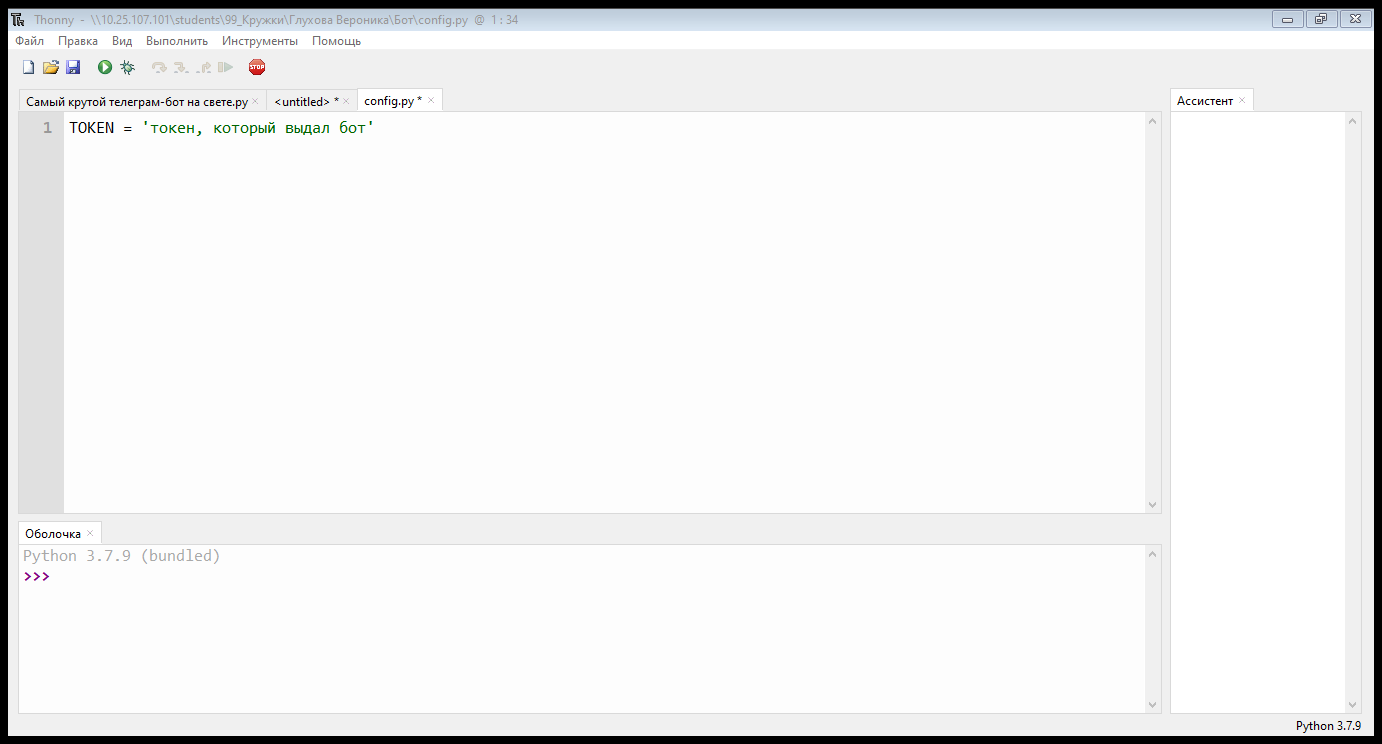


Рисунок 1. Хранение токена

Далее необходимо импортировать библиотеку “Telebot” и файл с токеном [9]. С целью упрощения процедуры импорта необходимо, чтобы файл с токеном бота находился в том же каталоге, что и основной файл приложения.

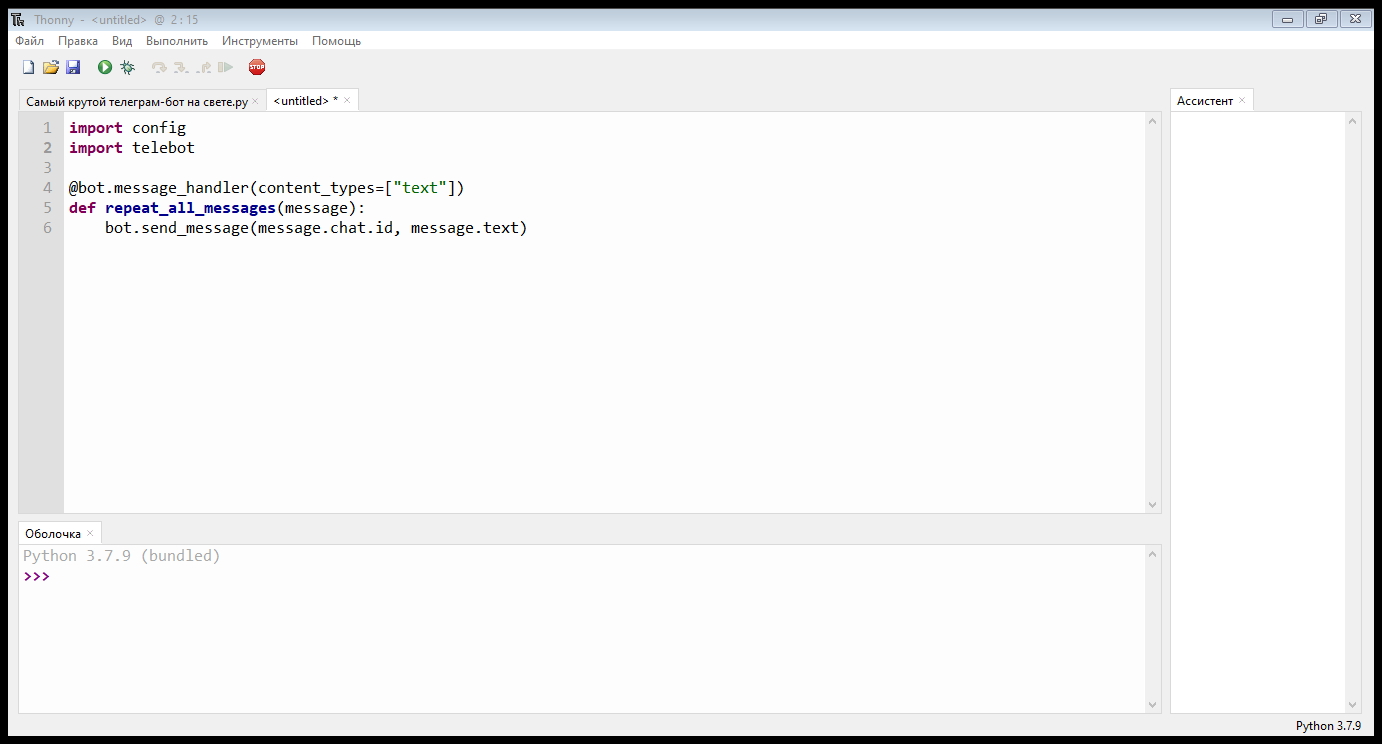


Рисунок 2. Импорт необходимых библиотек

Следующим этапом является описание работы обработчика, с помощью которого программа будет считывать все сообщения, отправленные пользователями.

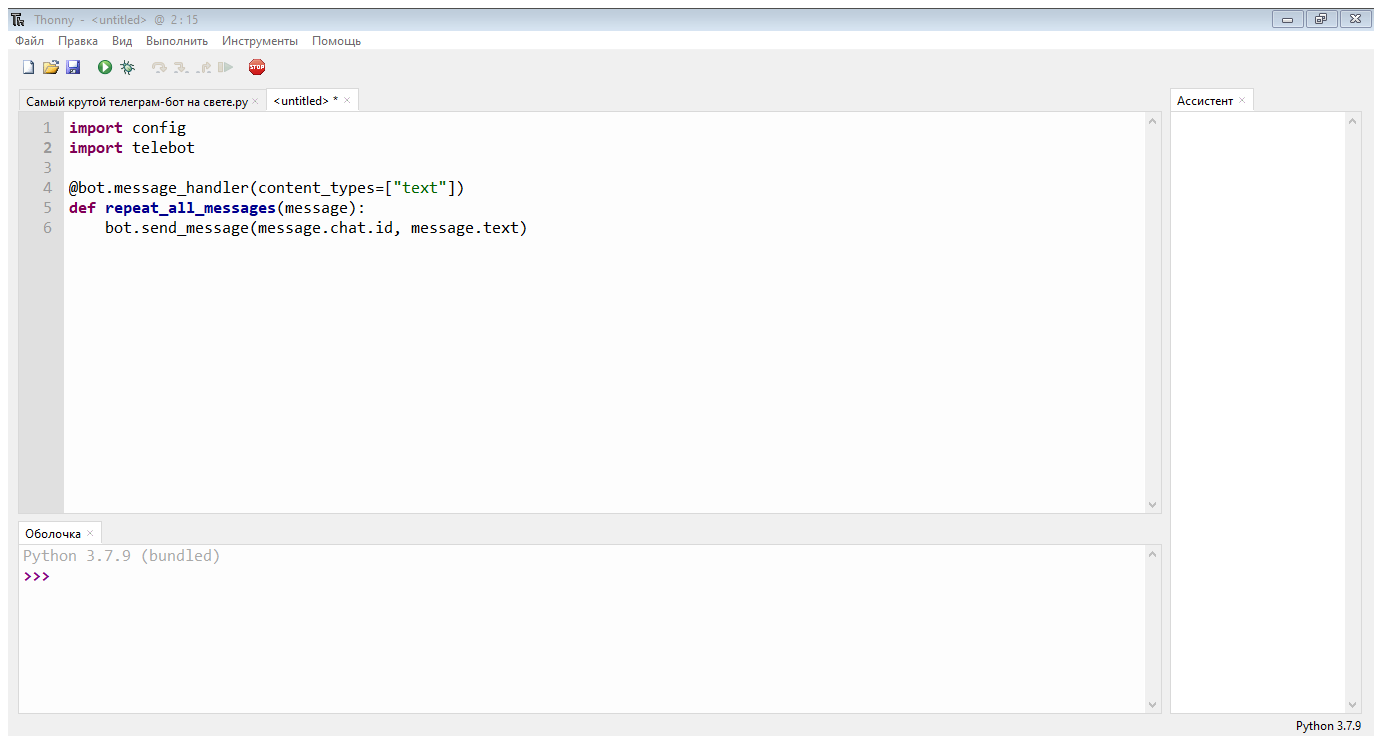


Рисунок 3. Обработчик текста

С помощью декоратора [10] после символа “@” мы указываем какой тип данных мы обрабатываем, в данном случае это любое текстовое сообщение. Сразу после объявления декоратора создается функция [11], в которой мы узнаем от какого пользователя приходит текст, и отправляем то, что нам прислал пользователь, тем самым проверяя работоспособность бота.

Для того, чтобы бот всегда отвечал на сообщения, надо запустить функцию, которая будет стараться не дать боту остановиться.

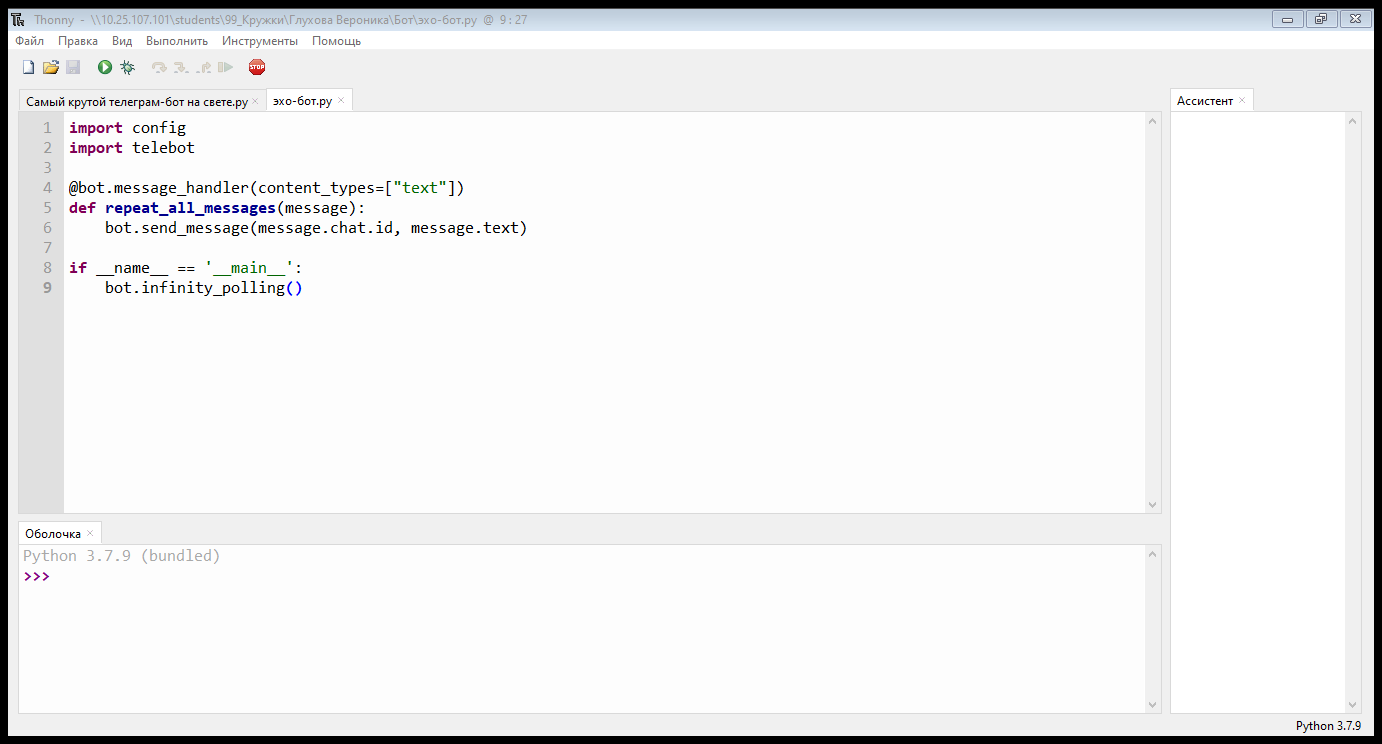


Рисунок 4. Защита бота от падения

Готовый эхо-бот представлен на рисунке 5, результат работы на рисунке 6.

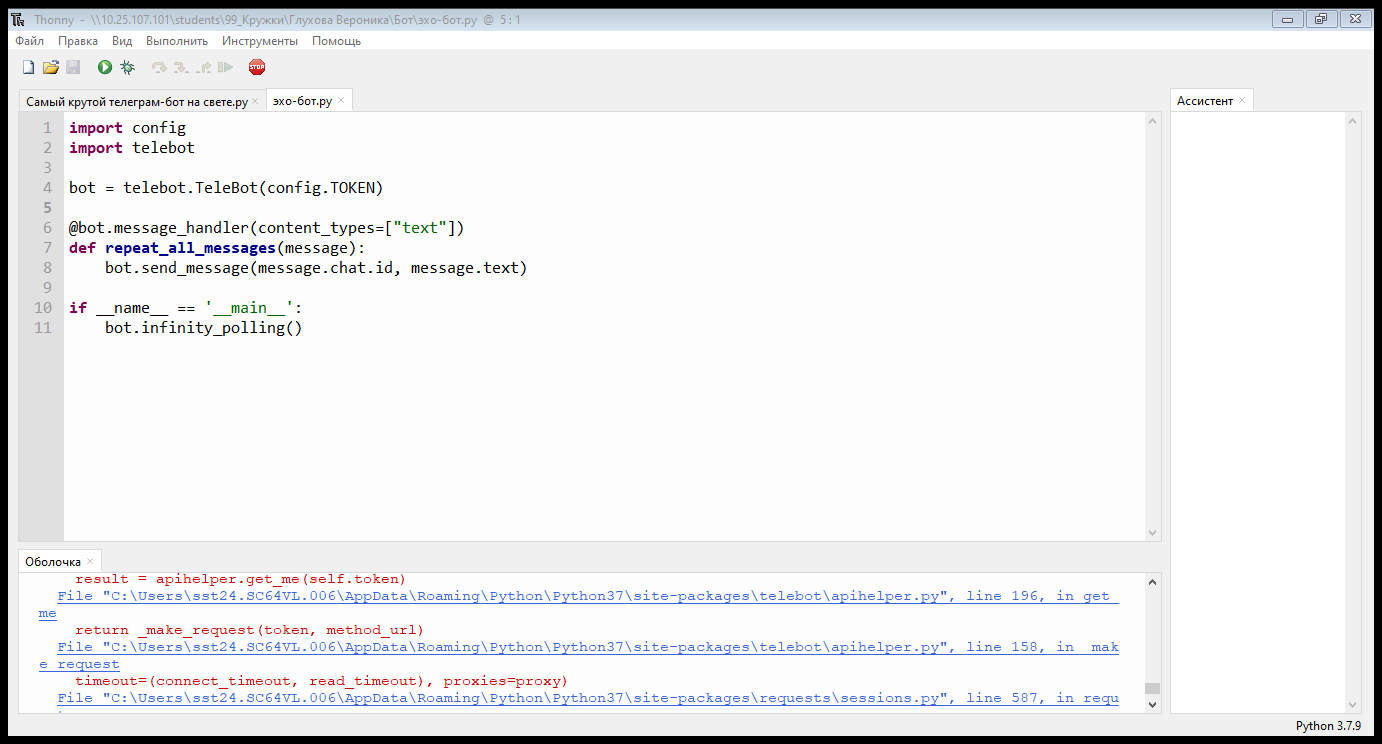


Рисунок 5. Готовая программа

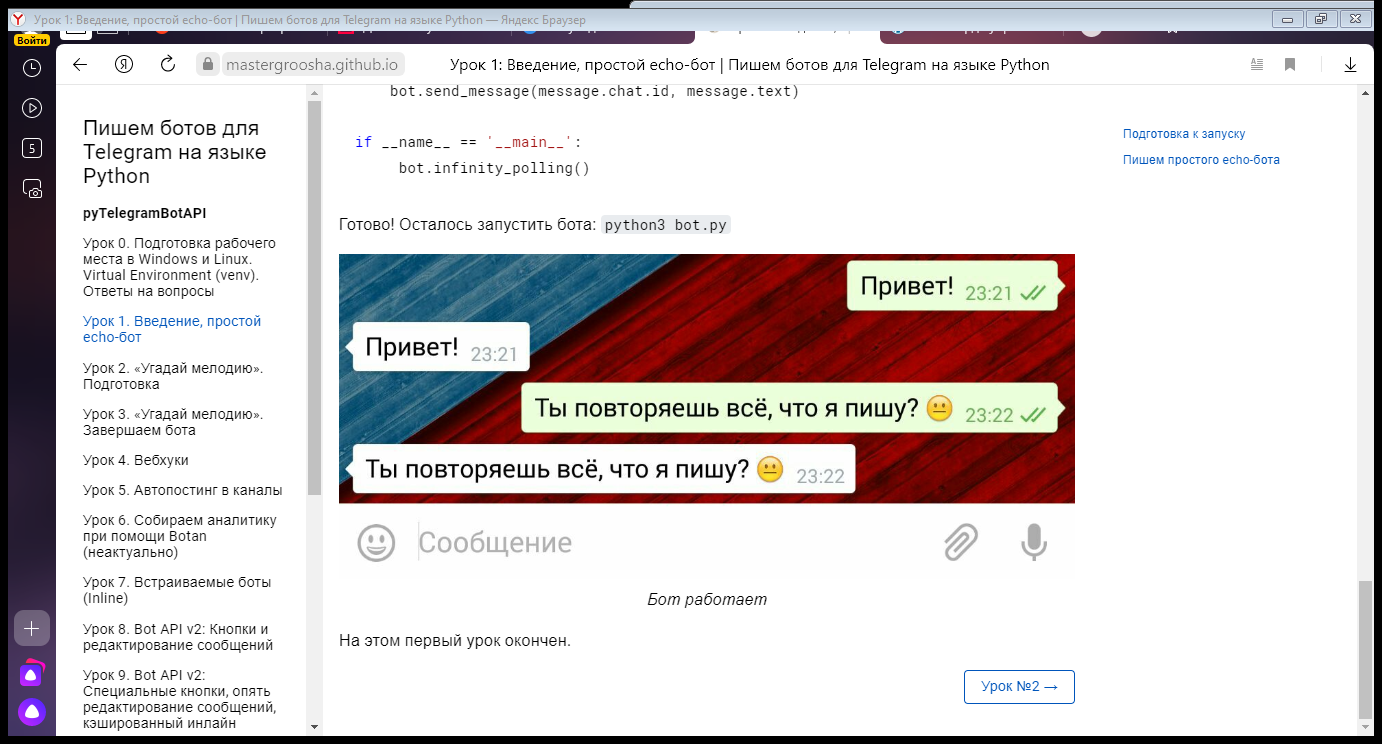


Рисунок 6. Пример работы программы

## Реализация основной логики чат-бота

Первый запуск бота сопровождается автоматической командой «start», и, если мы хотим, чтобы «Карманный дневник» соответствующе реагировал на нее, необходимо прописать новый обработчик. В нем мы так же, как и в прошлый раз указываем тип сообщения и заводим новую функцию, обрабатывающую новые сообщения. Однако в этой функции вместо сообщения пользователя мы отправляем человеку приветственный текст, где и рассказываем, что бот может и зачем он нужен.

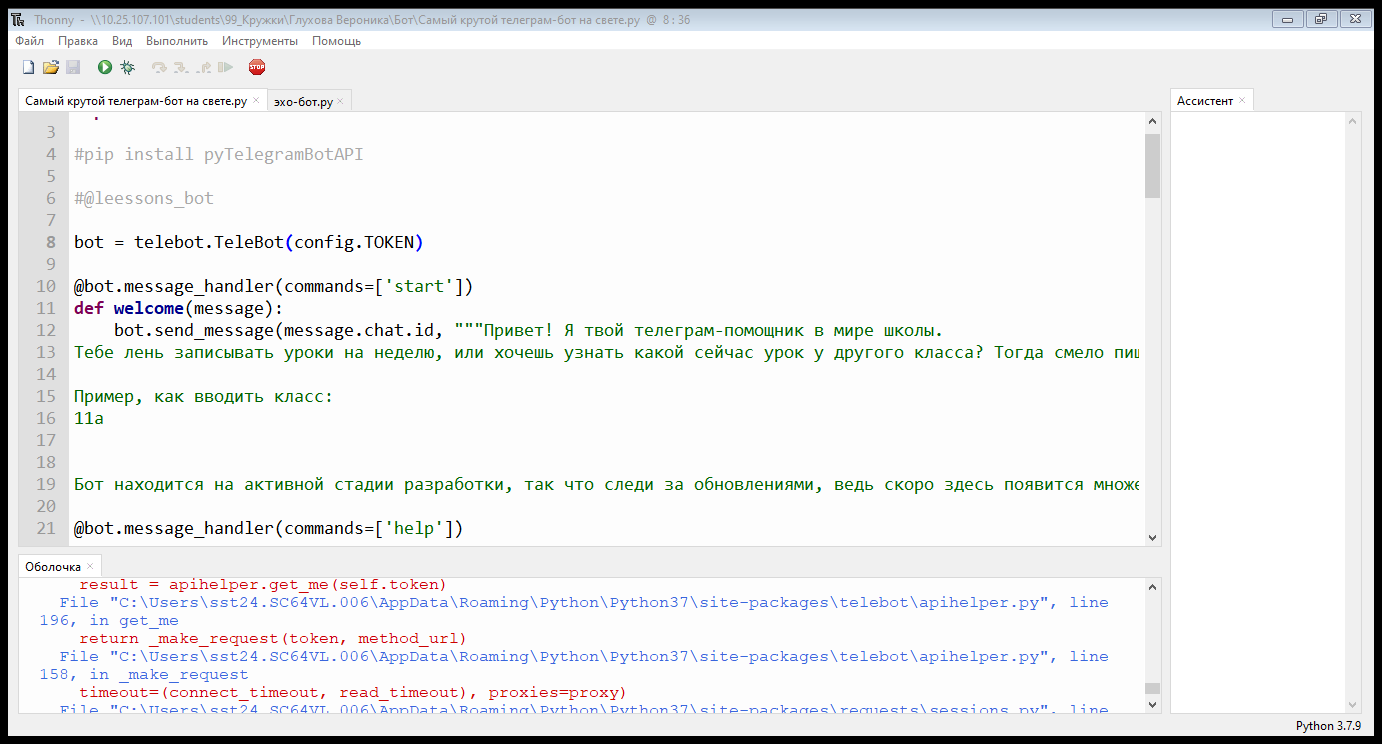


Рисунок 7. Рисунок обработчик команды «start»

Таким же способом оформляется функция «help».

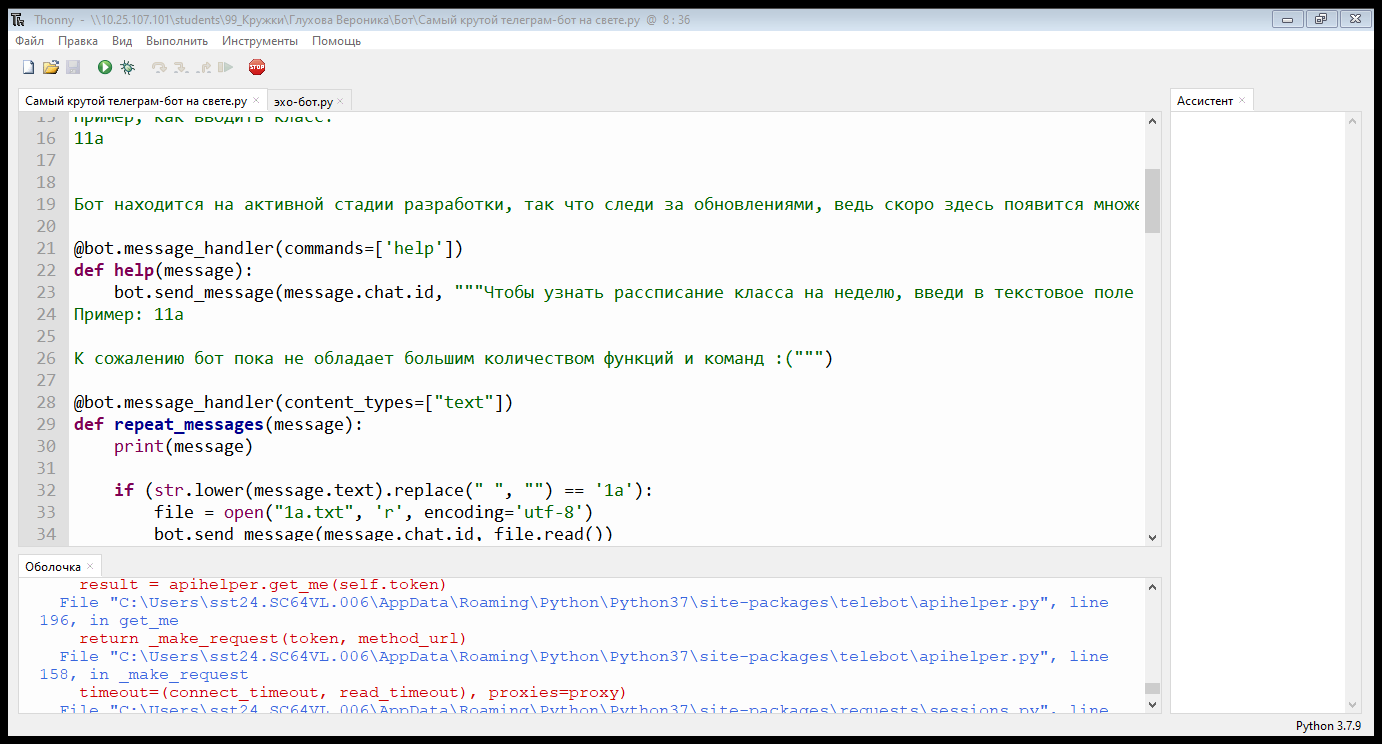


Рисунок 8. Обработчик команды «help»

Пользователь ввел свой класс, но что же дальше? Дальше программа снова должна обработать текст и вывести соответствующий файл с расписанием запрошенного класса.

Как это работает? Бот получает на вход текстовое сообщение и начинает проверять, есть ли совпадения введенного текста с нашим. Для этого бот удаляет лишние пробелы из полученного сообщения, если такие остались, и убирает их, после чего сравнивает, есть ли совпадения с нашими файлами. Если совпадения есть, то бот открывает файл в режиме чтения [12], в необходимой кодировке, и отправляет его пользователю, тем самым давая ему необходимое расписание занятий на неделю.

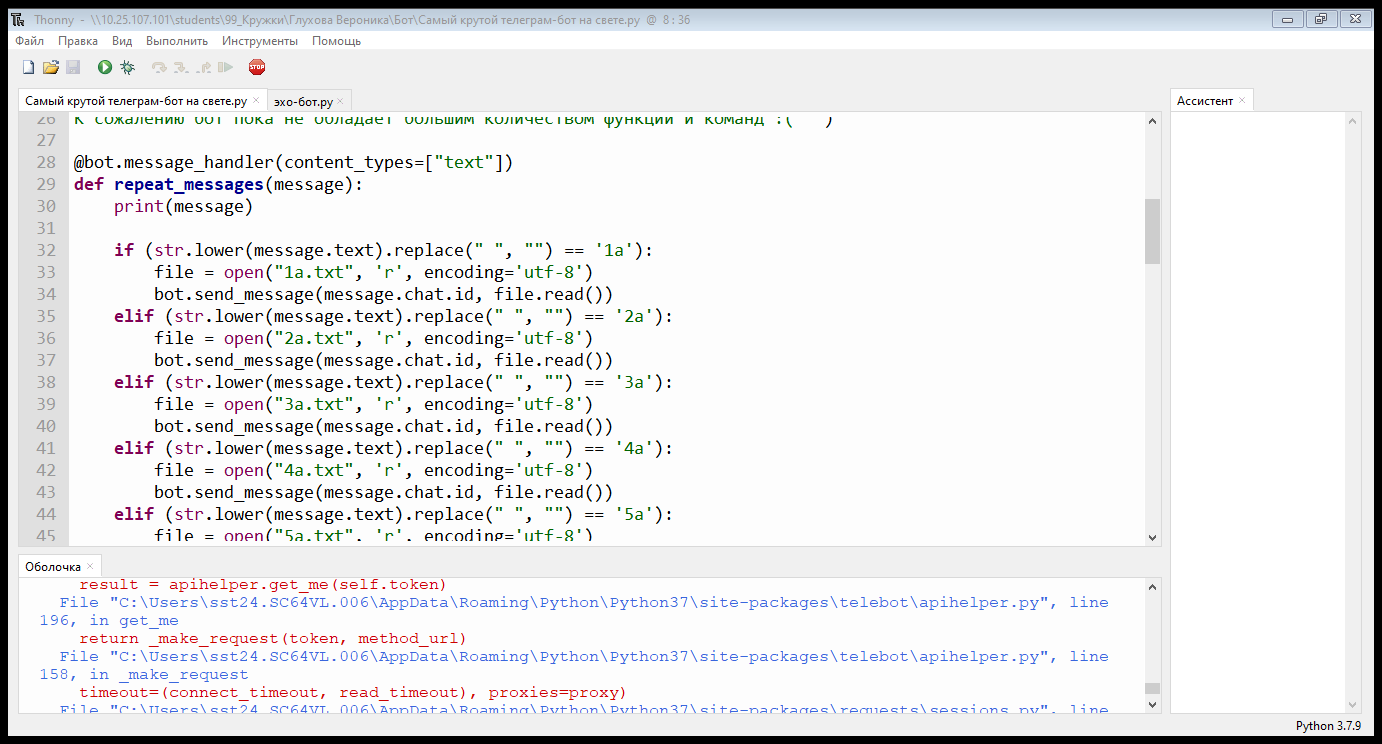


Рисунок 9. Обработчик текста

Так же помимо расписания бот может отправить и свои сообщения в ответ на определенные сообщения пользователя, например, «дневник» может поздороваться, попрощаться и повторить за пользователем слово «эхо». А если сообщение не похоже ни на один из вышеперечисленных вариантов, то бот отвечает пользователю соответствующим сообщением [13].

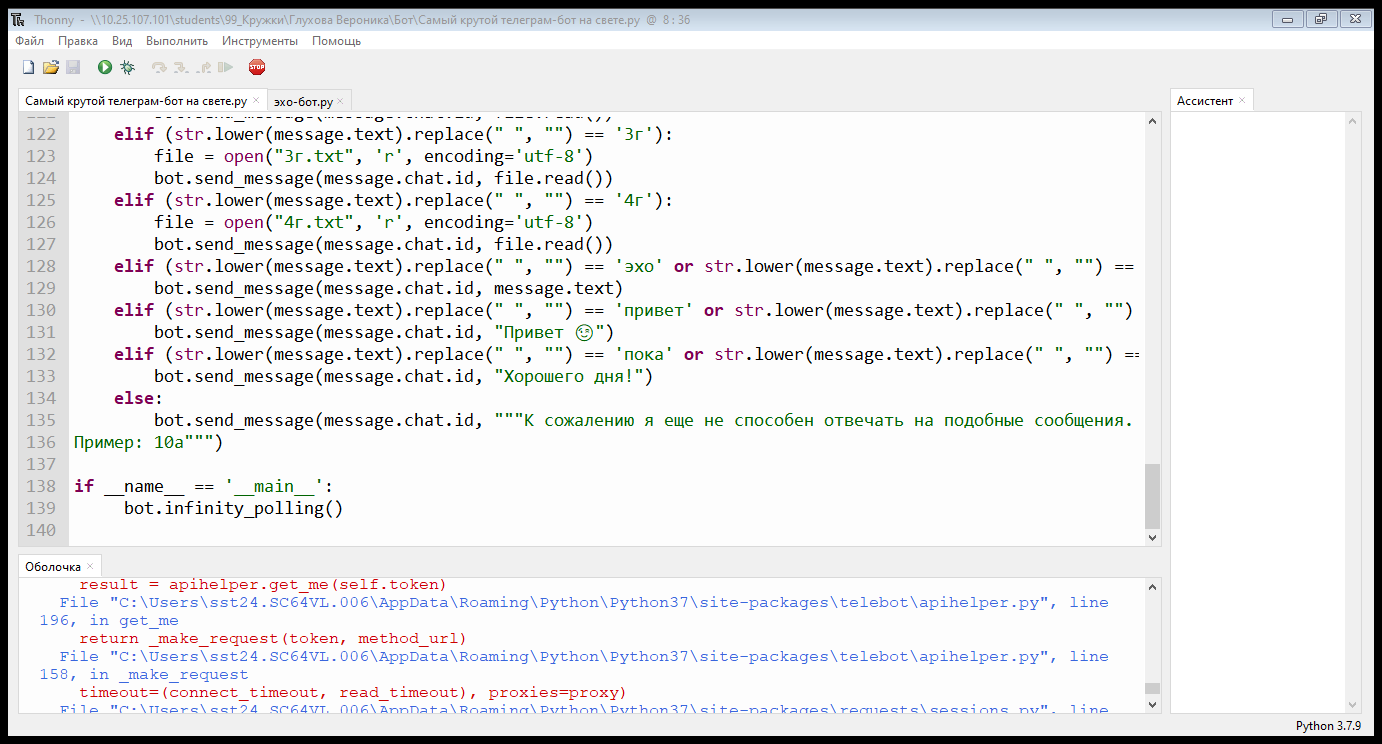


Рисунок 10. Некоторые дополнительные условия

Примеры работы программы:

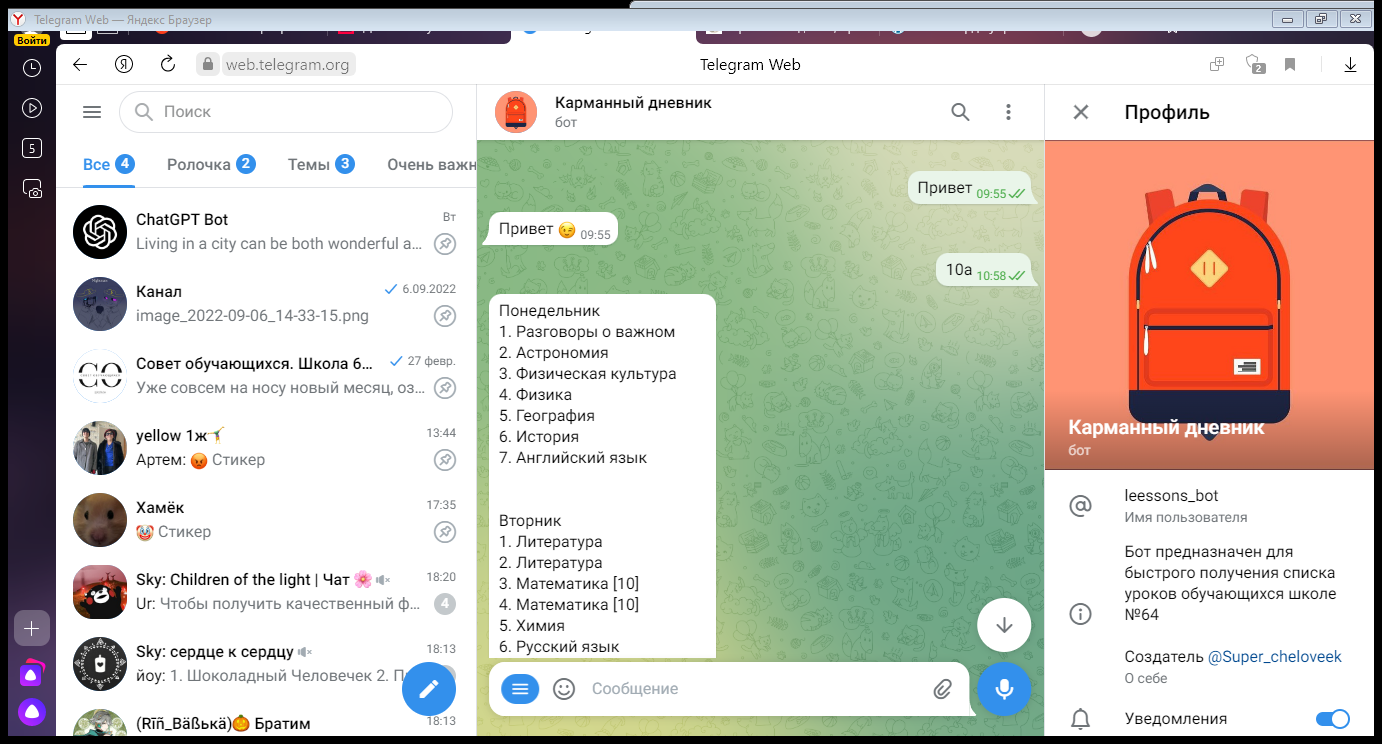


Рисунок 11. Пример работы готовой программы

## Настройка поставки информации о расписании уроков из информационной системы образовательной организации

На завершающем этапе необходимо было выполнить интеграцию чат-бота в информационную систему МБОУ «СОШ №64», расположенную по адресу <https://www.sc64vl.ru/>. В рамках организованной поставки информации о расписании уроков из ГИС «РО» (государственной информационной системы «Региональное образование») на сервера информационного портала, необходимо было реализовать функционал получения данных с MySQL-сервера базы данных в приложение, для этого мной был использован функционал библиотеки mysql-connector-python.

Используя функцию create\_connection, приведенную на Рисунке 12 и функцию execute\_read\_query, приведенную на Рисунке 13 реализуется функционал подключения к серверу базы данных и выполнения запроса типа «SELECT» с целью получения информации из базы данных.

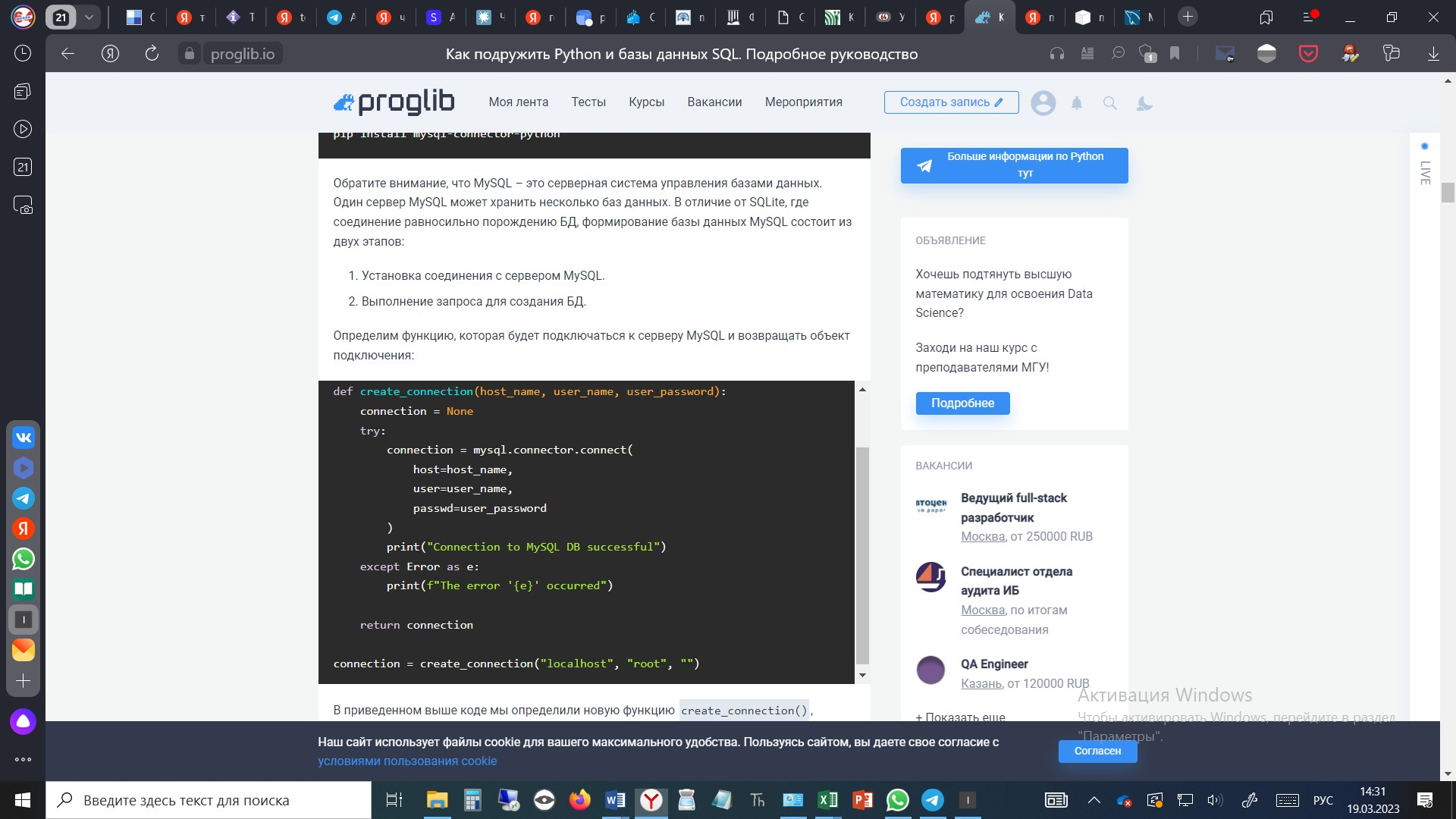


Рисунок 12



Рисунок 13

# Заключение

В ходе работы над данным проектом были изучен теоретические и практические аспекты проектирования Telegram-ботов, была создана модель, описывающая поведение чат-бота с применением языка программирования Python и Telegram API и организована поставка информации и расписании уроков из информационной системы образовательной организации.

Таким образом была достигнута цель создания инструмента доставки информации о расписании занятий.

# Источники

1. Боты: введение для разработчиков / Текст: электронный // Информационный портал telegram.org. – URL: <https://core.telegram.org/bots> (дата обращения: 9.03.2023). – Режим доступа: свободный.
2. Компьютерные языки / Python & ЕГЭ | ОГЭ Информатика – Текст: электронный // Информационный портал dzen.ru. – 13 февраля – URL: <https://dzen.ru/a/Y-M5goDexDLdx8T2> (дата обращения: 9.03.2023). – Режим доступа: свободный.
3. Что такое Python? / Текст: электронный // Информационный портал aws.amazon.com. – URL: <https://aws.amazon.com/ru/what-is/python/> (дата обращения: 9.03.2023). – Режим доступа: свободный.
4. Telegram вошёл в топ-5 самых загружаемых приложений в мире. Павел Дуров рассказал об уникальности мессенджера / Текст: электронный // Информационный портал iXBT.com. – 28 апреля 2022 – URL: https://www.ixbt.com/news/2022/04/28/telegram-voshel-v-top5-samyh-zagruzhaemyh-prilozhenij-v-mire-pavel-durov-rasskazal-ob-unikalnosti-messendzhera.html (дата обращения: 9.03.2023). – Режим доступа: свободный.
5. Telegram часто задаваемые вопросы / Текст: электронный // Информационный портал telegram.org. – URL: <https://telegram.org/faq> (дата обращения: 9.03.2023). – Режим доступа: свободный.
6. API для Telegram / Текст: электронный // Информационный портал core.telegram.org. – URL: <https://core.telegram.org/api> (дата обращения: 9.03.2023). – Режим доступа: свободный.
7. Для чего нужна библиотека программисту? / Текст: электронный // Информационный портал synergy.ru. – 19.07.2021 – URL: <https://synergy.ru/akademiya/programming/chto_takoe_biblioteka_v_programmirovanii> (дата обращения: 9.03.2023). – Режим доступа: свободный.
8. Введение, простой echo-бот / Текст: электронный // Информационный портал mastergroosha.github.io. – URL: <https://mastergroosha.github.io/telegram-tutorial/docs/lesson_01/> (дата обращения: 9.03.2023). – Режим доступа: свободный.
9. Добро пожаловать в документацию pyTelegramBotAPI! / Текст: электронный // Информационный портал pytba.readthedocs.io/. – URL: [https://pytba.readthedocs.io/ru/latest/index.html#](https://pytba.readthedocs.io/ru/latest/index.html) (дата обращения: 9.03.2023). – Режим доступа: свободный.
10. PythonDecorators / Текст: электронный // Информационный портал wiki.python.org. – 20 мая 2016 – URL: <https://wiki.python.org/moin/PythonDecorators> (дата обращения: 9.03.2023). – Режим доступа: свободный.
11. Функции и их аргументы / Текст: электронный // Информационный портал pythonworld.ru. – URL: <https://pythonworld.ru/tipy-dannyx-v-python/vse-o-funkciyax-i-ix-argumentax.html> (дата обращения: 9.03.2023). – Режим доступа: свободный.
12. Изучаем Python / Марк Лутц : Диалектика, 2019 г. – 73 с.
13. Программируем на Python / Майкл Доусон : Питер, 2020 г. – 43 с.