***Абросимов Ростислав Алексеевич***

**СОЗДАНИЕ ТЕЛЕГРАМ-БОТОВ  
 ДЛЯ ОРГАНИЗАЦИИ ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ ОБУЧАЮЩИХСЯ**

**Введение**

В настоящее время одним из приоритетных направлений в сфере образования является активное внедрение современных информационных технологий на всех этапах образовательного процесса. Исследования показывают, что в большинстве случаев цифровые ресурсы используются на занятиях только как источник нового содержания учебной деятельности и средство контроля, а в организации учебного процесса применяются для хранения (планы, списки, отчеты и др.) и передачи информации. Возможности применения новых технологий как способа проектирования учебного процесса и управления им использовались до последнего времени значительно реже. Однако сегодня наблюдается большой рост работ в данном направлении.

Возрастающая активность в привлечении информационных технологий к организации учебного процесса связана, в том числе, с изменениями отношений людей к компьютерам, смартфонам и прочим «гаджетам». Все больше людей пользуются смартфоном больше как средством общения через переписку, а не через разговор по мобильной связи. Причиной этому является широкое распространение мессенджеров среди населения. В последнее десятилетие тенденция по замене мобильной связи мессенджерами заметно изменилась в пользу вторых. проведенные исследования показывают [1], что за последние полтора года количество сообщений о мессенджерах в интернете выросло почти в полтора раза – с 11,5 млн в марте 2018 года до 16,4 млн в августе 2019 года. В отчете о состоянии цифровой сферы Digital 2020, который каждый год готовят We Are Social и Hootsuite [2], указано, что в январе 2020 года у 87% россиян на смартфонах были установлены мессенджеры, а приложения соцсетей – у 92% мобильных пользователей. Самое популярное мобильное приложение в России по количеству пользователей — WhatsApp. За ним следует Viber, и замыкает тройку лидеров приложение ВКонтакте [3]. Свое достойное место среди популярных мессенджеров занимает Telegram.

В связи с ростом популярности мессенджеров растет и популярность чат-ботов. Чат-бот – это специальная программа, которая работает в приложениях-мессенджерах и социальных сетях [4]. Боты помогают решать типовые задачи: задают пользователям вопросы и отвечают, ищут информацию, выполняют простейшие поручения и т.д. Отмечается [5], что процент пользователей, открывающих сообщения от чат-ботов существенно превышает процент тех, кто открывает сообщения в e-mail-рассылке (85% против 1-3%).

Telegram – это кроссплатформенный мессенджер, являющийся отечественной разработкой, и написанный на языке C++. Основными причинами популярности данного мессенджера являются:

* защита от несанкционированного чтения осуществляется благодаря применению протокола связи MTProto;
* наличие опции переписки в «секретном чате», обеспечивающей возможность обмена зашифрованными сообщениями;
* наличие таймера для автоуничтожения сообщений спустя заданное время;
* синхронизация между устройствами;
* создание каналов и чат-ботов.

MTProto – криптографический протокол, используемый в системе обмена сообщениями Telegram для шифрования переписки пользователей. В основе протокола лежит оригинальная комбинация симметричного алгоритма шифрования AES, протокол Диффи-Хеллмана для обмена 2048-битными RSA ключами между двумя устройствами и ряд хеш-функций [6].

В зависимости от подхода к разработке чат-бота, их можно разделить на две группы: простые и умные. Первая группа ботов распоряжается базой знаний, представляющая собой набор паттернов, на которые реагирует бот при получении сообщения. Одними из наиболее распространенных являются ключевые слова или совпадение фразы. Разработчик прописывает их заранее, используя регулярные выражения либо другие формы анализа строк.

Вторая же базируется на использовании нейронных сетей. В отличие от простых ботов, умные анализируют каждое сообщение от пользователя, и используют их для самообучения и предоставления более точной информации.

Из-за своей простоты первая группа ботов пользуется большей популярностью у разработчиков. С помощью ботов компании могут как улучшить обслуживание существующих клиентов, так и привлечь новых. Это достигается за счет быстрого отклика и автоматизации повторяющихся задач.

**1 Постановка задачи**

В работе рассматривается разработка бота, предназначение которого – организация записи обучающихся школы на занятия внеурочной деятельностью.

В соответствии с ФГОС внеурочная деятельность рассматривается как важная и неотъемлемая часть процесса образования детей школьного возраста [7]. Цели организации внеурочной деятельности определяются изложенными в ФГОС требованиями к результатам освоения основных образовательных программ общего образования (начального, основного, среднего), интересами и потребностями обучающихся, запросами их родителей, целевыми установками педагогического коллектива образовательного учреждения.

Внеурочная деятельность направлена на развитие личности по всем направлениям:

* общеинтеллектуальное развитие,
* общекультурное развитие,
* спортивно-оздоровительное развитие,
* духовно-нравственное развитие,
* социальное развитие.

Внеурочная деятельность в школе позволяет решить целый ряд очень важных задач:

* обеспечить благоприятную адаптацию ребенка в школе;
* оптимизировать учебную нагрузку обучающихся;
* улучшить условия для развития ребенка;
* учесть возрастные и индивидуальные особенности обучающихся.

При организации внеурочной деятельности школы используют различные формы, такие, как кружки, секции, круглые столы, конференции, экскурсии, диспуты, школьные научные общества, олимпиады, соревнования, поисковые и научные исследования, общественно-полезные практики и другие.

Школа предлагает ученикам большой список организуемых занятий, сориентироваться в котором часто бывает сложно и детям, и их родителям. Кроме того, учителя также сталкиваются с определенными трудностями: им необходимо проинформировать детей и родителей о занятиях (курсах, кружках, секциях), согласовать списки участников и расписание и т.д.

С учетом актуальности данной проблемы, авторами предлагается разработка инструментария для оказания информационной поддержки всем участникам процесса (ученикам, родителям, учителям). Основным инструментарием выбран Telegram-бот [8], который выступает информационным консультантом школьников по вопросам выбора занятий для внеурочной деятельности.

Данное решение, в отличие от полноценного приложения, позволит ученикам, вне зависимости от операционной системы смартфона, получить рекомендации по выбору занятий, посмотреть расписание и записаться на занятия.

Функциональные требования. Основными возможностями бота должны являться:

* получения от пользователя ключевой информации для поиска подходящих курсов;
* отправка пользователю информации о курсе (общее описание, тематика занятий, расписание);
* добавление курса в список пользователя, просмотр списка.

Аппаратный интерфейс. Должно быть установлено приложение Telegram на платформу iOS или Android. Версия Android не ниже 7-й, версия IOS не ниже 12-й. Для работы программного средства требуется подключение к сети Интернет.

Характеристики пользователя. Основными пользователями являются сотрудники образовательного учреждения, обучающиеся и их законные представители, владеющие смартфоном с установленным приложением Telegram.

В последние годы появляется все большее число работ, посвященных различным аспектам использования мессенджеров в образовательных учреждениях [9-11].

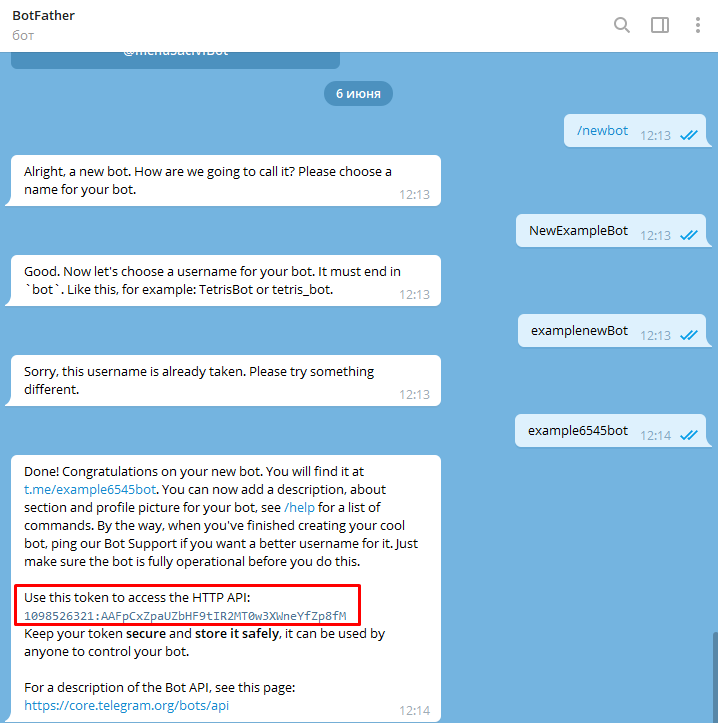
**2 Описание технологий разработки**

**2.1** **Фреймворк aiogram для работы с TelegramBotAPI**

TelegramBotAPI – это интерфейс для создания ботов в мессенджере Telegram. Работа данного API основывается на отправке и получении HTTPS запросов к серверу Telegram [12]. Для этого каждому боту присваивается уникальный токен, который имеет следующий вид:

123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11.

Чтобы его получить, нужно создать нового бота в мессенджере Telegram, написав боту @BotFather команду /newbot и ввести название и имя бота. Процесс создания бота изображен на рисунке 1.



***Рисунок 1. Создание бота и получение токена***

После этого запросы осуществляются по специальной ссылке вида https://api.telegram.org/bot<токен>/название\_метода, где название\_метода – метод, формирующий запрос к API.

Существует два способа получения обновлений от бота – это polling (long-polling) и webhook [13]. Polling – это «опрашивание» клиентом сервера на наличие в нем обновлений от бота. Это частые запросы, которые происходят раз в несколько секунд. Однако, данный подход может быть ресурсозатратным, поэтому в основном используется long-polling. Основное его отличие в том, что сервер дольше отвечает клиенту. У каждого запроса есть timeout – время, в течение которого нужно ответить и, в зависимости от того, есть ли обновления или нет, сервер либо отвечает на запрос, не дожидаясь конца timeout, либо не отвечает вообще.

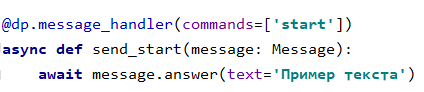
Webhook практически то же самое, за тем исключением, что сервер будет сам отвечать клиенту. Для этого создается специальный URL-адрес, на который сервер отправляет команды и сообщения, пришедшие от пользователя. Как только боту написал пользователь, сервер отправляет POST-запрос на webhook-адрес вместе с Update объектом, сериализованным в формате JSON. В данном объекте указывается вся информация о пришедшем событии.

У каждого из способов получения обновлений есть свои преимущества и недостатки. Из-за того, что polling периодически посылает запросы, в ответ часто приходят пустые сообщения, однако простота реализации является его основным плюсом. Webhook же, за счет использования URL-адреса для принятия и обработки запросов от сервера снижает нагрузку на сеть.

Сам API бота может быть реализован либо с помощью стандартной библиотеки, предоставляемой Telegram, либо с помощью сторонних библиотек. Авторами проанализированы следующие библиотеки: python-telegram-bot и aiogram, написанные на языке Python [14].

В результате анализа выбор сделан в пользу aiogram по нескольким причинам. Во-первых, это полноценный фреймворк, который предоставляет куда больше возможностей, по сравнению с python-telegram-bot, являющейся библиотекой. Во-вторых, он написан на основе асинхронных методов asyncio и aiohttp. Асинхронность позволяет боту выполнять больше запросов одновременно, и не нагружать сервер при большом количестве одновременных пользователей.

Одним из главных преимуществ использования библиотеки является читабельность кода. Например, для фильтрации сообщений от пользователя используются декораторы. Декоратор в языке Python является «оберткой» для некоторой функции, с помощью которого можно добиться избавления от повторяющегося кода в программе. На рисунке 2 приведен пример обработки команды start.



***Рисунок 2. Пример обработки команды start***

Декоратор message\_handler принимает параметры фильтрации, такие, как список команд, типов (текст, фотография, стикер) или регулярное выражение. Следующая за ним функция выполнится только в том случае, когда выполняется условие фильтрации, что происходит при получении команды start от пользователя.

**2.2 Формат хранения данных JSON**

JSON – сокращение от JavaScript Object Nonation, являющимся текстовым форматом для хранения и обмена информацией [15]. Данный формат хранит структурированную информацию и в основном используется для передачи данных между сервером и клиентом. Исходя из названия, JSON произошел из языка программирования JavaScript, но он также доступен для использования во многих других языках, в том числе Python.

Структура файла состоит из набора пар ключ-значения. В этой паре ключ отделяется от значения с помощью знака двоеточия, а одна пара от другой – с помощью запятой. Ключ должен быть неизменяемым типом данных, то есть строкой или числом, а значение – любым другим. При этом ключ в JSON, в отличие от объекта, обязательно заключается в двойные кавычки. На рисунке 3 показан пример структуры файла.



***Рисунок 3. Структура файла JSON***

Данный пример является простым представлением объекта, но некоторые ключи также могут хранить вложенные структуры. В языке Python структура, хранящая пары ключ-значение, называется словарем. Поэтому данный формат удобен для работы с большим количеством вложенных структур за счет простого доступа к ним.

Преимуществами выбора именно данного формата для хранения категорий «Курс», а также их описаний вместе с характеристиками являются:

компактность – занимает сравнительно небольшой объем;

удобство чтения – текстовое содержание может легко создаваться и интерпретироваться компьютером и человеком;

мобильность – можно преобразовать в структуру практически для всех видов формальных языков, использующихся для создания компьютерных программ;

большинство языков программирования, обладают функциями и специальными инструментами для чтения и редактирования файлов.

**2.3 Регулярные выражения**

Регулярные выражения – язык поиска подстроки или подстрок в тексте. Регулярные выражения применяются в любых задачах по поиску в множестве данных, для которых нужно получать выборку по определенным параметрам. С их помощью можно искать некоторые шаблоны, например, электронные адреса, телефонные номера, адреса веб-сайтов и так далее.

Язык регулярных выражений состоит из обычных символов или групп символов, позволяющих найти определенную последовательность в строке. Для упрощения поиска подстроки используются специальные операторы, которые приведены в таблице.

***Таблица.***

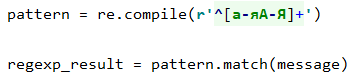
**Часто используемые операторы в регулярных выражениях**

|  |  |
| --- | --- |
| **Оператор** | **Находит** |
| . | Один любой символ, кроме новой строки \n |
| + | 1 и более вхождений предыдущего шаблона |
| \* | 0 и более вхождений предыдущего шаблона |
| ^ и $ | Начало или конец строки |
| \w | Любую цифру или букву |
| \d | Любую цифру в интервале [0-9] |
| \s | Любой пробельный символ |
| \b | Границы слова |

В Python механизм регулярных выражений подключается с помощью модуля re. Данный модуль предоставляет набор некоторых функций для работы с регулярными выражениями:

* re.match(pattern, text) – поиск совпадения по заданному шаблону в начале строки;
* re.search(pattern, text) – поиск совпадения во всей строке и возврат первого найденного;
* re.findall(pattern, text) – поиск всех совпадений и возврат списка всех найденных.

В качестве фильтра сообщений от пользователя в чат-боте может выступать регулярное выражение. Простой пример использования регулярного выражения представлен на рисунке 4.



***Рисунок 4. Проверка сообщения на соответствие***

Здесь используется метод compile, компилирующий выражение, которое в дальнейшем можно использовать как шаблон. Выражение в параметрах шаблона задает слово, содержащее в себе только буквы русского языка и расположенное в начале строки.

## **2.4 Реляционная СУБД Postgresql**

Для работы чат-бота необходимо информацию о учебных курсах (кружках), учениках, а также о сделанном учеником выборе. Информация хранится в базе данных. В качестве основных были рассмотрены два варианта, а именно SQLite [16] и PostgreSQL [17].

SQLite – это встраиваемая кроссплатформенная СУБД, поддерживающая достаточно полный набор команд SQL. Встраиваемая означает, что SQLite не использует парадигму клиент-сервер, то есть ее ядро не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы.

SQLite может претендовать на звание наиболее используемой базы данных в мире благодаря ее внедрению во многие популярные веб-браузеры, операционные системы и мобильные телефоны. Так как SQLite не является клиент-серверным механизмом, это создает ее главное преимущество: во встраиваемых или распределенных системах каждая машина несет полную реализацию базы данных. Такое решение может значительно повысить производительность баз данных, поскольку снижает потребность в межсистемных вызовах.

В отличие от SQLite PostgreSQL является объектно-реляционной СУБД, которая основана на POSTGRES – программе, разработанной на факультете компьютерных наук Калифорнийского университета в Беркли. PostgreSQL является одной из самых популярных баз данных. За более чем 20-летнюю историю развития на прочном фундаменте, заложенном академической разработкой, PostgreSQL выросла в полноценную СУБД уровня предприятия и составляет реальную альтернативу коммерческим базам данных.

Данная СУБД поддерживает крайне необходимые и интегральные функциональные возможности объектно-ориентированной или реляционной базы данных, такие как полная поддержка транзакций, в том числе концепта ACID (Atomicity, Consistency, Isolation, Durability).

В результате сравнения данных СУБД для реализации проекта была выбрана PostgreSQL в качестве основной. Поскольку чат-боты являются серверными приложениями, а не портативными, то для их работы требуется постоянное общение с сервером. SQLite позволяет быстро выполнять запросы к базе данных, если приложение является встроенным, так как прямой доступ к дисковой памяти является одним из плюсов данной РСУБД. Однако для выполнения большого количества операций записи она не подходит, так как SQLite может выполнять только одну операцию записи в любой момент времени, что существенно ограничит ее пропускную способность.

В то же время PostgreSQL обладает такими особенностями, как обширный список типов данных, поддержка сообществом и безопасности, позволяющие в дальнейшем расширить базу без каких-либо трудностей. Следовательно, данная СУБД идеально подходит для хранения информации о большом количестве учащихся школы и в будущем позволит расширить функционал приложения.

## 

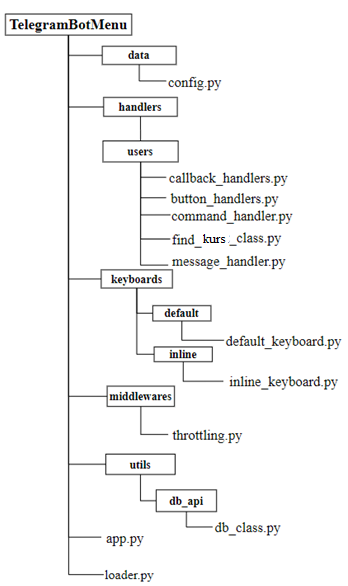
## **3 Разработка программного средства**

## **3.1 Основная логика и структура программного средства**

Язык программирования Python является интерпретируемым языком, то есть файл программы исполняется строка за строкой. Это позволяет разбить программу на отдельные составляющие, где каждый файл будет являться модулем, отвечающим за те или иные задачи. На рисунке 5 представлена подробная схема проекта.

Так как работа чат-бота разделена на отдельные составляющие, это позволяет точно настроить его для общения с пользователем. Такое общение достигается за счет диспетчера, занимающийся приемом всех обновлений, которые приходят от бота.

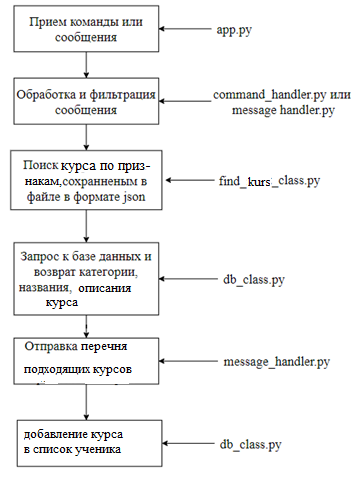
Диспетчер объявляется в файле loader.py, и, в дальнейшем импортируется из него в модули, называемые обработчиками. Обработчики получают переданный запрос от диспетчера об обновлении и определяют тип данного запроса. Тип запроса может варьироваться от того, какое действие совершил пользователь, например, отправка сообщения имеет тип message, содержащий в себе уникальный id чата и переданный текст. В структуре, показанной выше, все обработчики находятся в директории users.



***Рисунок 5. Схема структуры проекта***

После того как обработается запрос, измененный диспетчер возвращается в файл app.py, где происходит запуск самого приложения, а также long-polling, описанный ранее, то есть происходит опрашивание сервера на наличие новых обновлений.

Логика работы бота показана на схеме, изображенной на рисунке 6.



***Рисунок 6. Схема работы чат-бота***

На схеме видно, что каждому блоку соответствует определенный обработчик. Стоит заметить, что помимо обработчиков, в структуре присутствует два особо важных модуля, а именно find\_kurs\_class.py и db\_class.py.

Первый модуль реализует класс FindIng, содержащий основные методы для проверки сообщения и поиска признаков учебного курса. Данный класс содержит в себе следующие методы:

* open\_menu – открытие файла получения данных о предлагаемых курсах;
* parse\_message – проверка входящего сообщения от пользователя на соответствие шаблону;
* find\_dish – основной метод по нахождению принятого признака курса в общем списке.

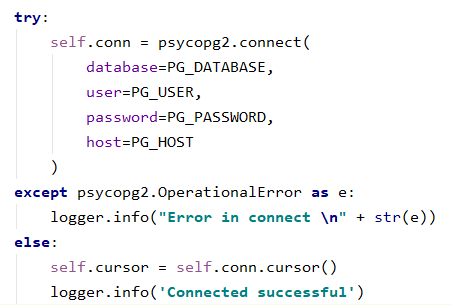
Второй – класс DbWork для подключения и работы с базой данных. Структура базы данных, а также описание методов данного класса приведены в следующем разделе.

## **3.2 Реализация базы данных чат-бота**

База данных состоит из девяти связанных между собой таблиц. Основные таблицы следующие:

* Сustomer – хранит информацию о пользователе;
* Kurs – хранит данные о курсах, кружках, секциях;
* Category – хранит названия категорий, к которым относятся курсы;
* Order\_customer и order\_kurs – таблицы, связывающие выбранные учеником курсы и информацию о курсах.

Работа с базой данных в языке Python осуществляется с использованием библиотеки psycopg2. Для подключения к серверу требуется передать основные его параметры (название базы данных, имя пользователя, пароль и адрес хоста) в метод connect класса psycopg2. Выполнение запросов и получение нужной информации о курсах и выборе ученика происходит с помощью отдельного класса cursor, получаемый от объекта conn (рисунок 7).



***Рисунок 7. Подключение к базе данных***

В результате создания основного кода класса базы данных были реализованы следующие методы:

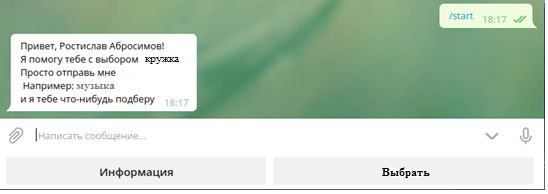
* insert – вставка новой информации в базу данных;
* select\_dish – запрос информации о курсах;
* select\_order – запрос информации о выборе ученика;
* delete\_order – удаление курса у пользователя.

## ***3.3 Программный интерфейс приложения***

Взаимодействие с чат ботом происходит посредством отправки ему команд, сообщений или нажатий на кнопки. Для общения доступны следующие команды:

* /start – начало диалога с ботом;
* /help – просмотр доступных команд;
* /info – информация о боте;
* /order – просмотр выбранных курсов.

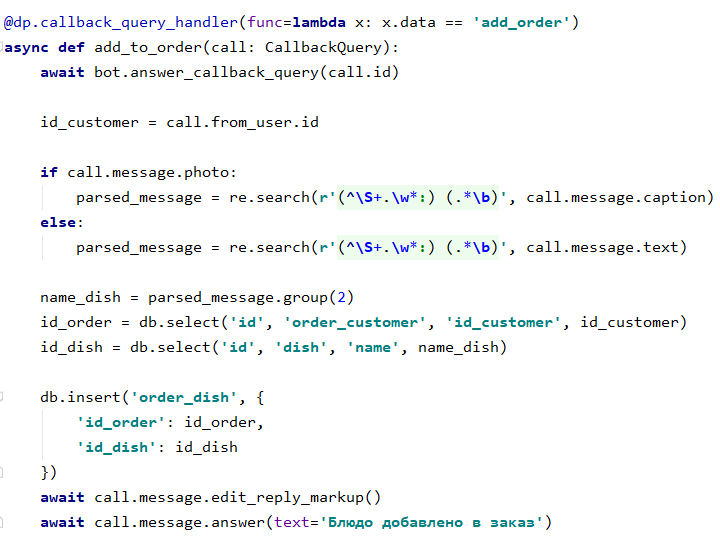
Каждый новый пользователь начинает диалог с ботом отправкой команды start (рисунок 8).



***Рисунок 8. Начало общения с пользователем***

Следующим шагом будет отправка боту какого-либо признака курса для поиска подходящих курсов.

Под каждым курсом есть кнопка «Выбрать». Данная кнопка является inline-кнопкой, то есть при ее нажатии, она посылает «обратную информацию» (callback data), которую принимает обработчик, находящийся в модуле callback\_handlers.py (см. рисунок 5). Фрагмент кода функции обработчика изображен на рисунке 9.



***Рисунок 9. Функция добавления курса***

Декоратор callback\_query\_handler принимает callback data с названием add\_order. Далее сработает функция add\_to\_order, которая с помощью регулярных выражений найдет информацию о курсе в ответном сообщении. В результате из базы данных будут выбраны нужные поля для использования их в запросе записи ученика на курс.

После выбора курса список выбранных курсов можно просмотреть, нажав на кнопку «Мои курсы» либо отправив команду order. От выбора можно отказаться, нажав на inline-кнопку «Удалить заказ».

## **3.4 Обработка ошибок пользователя**

Ошибки, возникающие в процессе использования бота, можно разделить на три типа, каждый из которых обрабатывается по-разному:

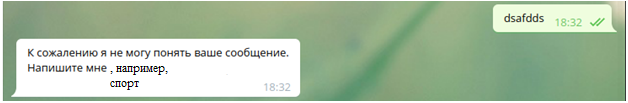
* первый тип – слишком много запросов одновременно;
* второй тип – неправильный ввод сообщения;
* третий тип – не найден признак курса.

Первый тип ошибок – это обращение к боту путем частой отправки различных обновлений, например, команды или сообщения. Чтобы не обрабатывать каждый из таких запросов, был добавлен так называемый middleware. Middleware – это связующая функция, которое стоит между запросом и его обработкой. В структуре проекта на рисунке 5 он реализован в файле throttling.py. Пример ответа бота на два запроса подряд можно увидеть на рисунке 10.



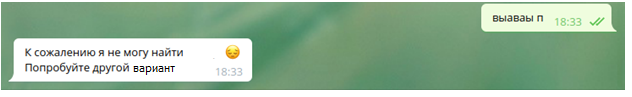
***Рисунок 10. Обработка частых запросов***

Второй тип ошибок возникает в случае несоответствия сообщения пользователя заданному шаблону. В качестве шаблона выступает регулярное выражение, описание которого находится в подразделе 1.3. Пример обработки данного типа ошибок изображен на рисунке 11.



***Рисунок 11. Ошибка несоответствия заданному шаблону***

И последний тип ошибок – введенного ингредиента попросту нет ни в одном блюде конкретного заведения (рисунок 12).



***Рисунок 12. Курс (кружок, секция) не найден***

**Заключение**

В работе описан процесс создания Telegram-бота для поддержки организации внеурочной деятельности в школе. Использованные при выполнении работы технологии могут применяться для создания чат-ботов различного назначения.

**Список литературы:**

1. Мессенджеры в России в 2019 году: новые лидеры и перспективные новички [Электронный ресурс] // URL: https://www.web-canape.ru/business/internet-2020-globalnaya-statistika-i-trendy/ (дата обращения: 20.06.2020).
2. Digital in 2020 [Электронный ресурс] // URL: https://wearesocial.com/digital-2020 (дата обращения: 20.06.2020).
3. Вся статистика интернета на 2020 год — цифры и тренды в мире и в России [Электронный ресурс] // URL: https://vc.ru/social/83628-messendzhery-v-rossii-v-2019-godu-novye-lidery-i-perspektivnye-novichki/ (дата обращения: 20.06.2020).
4. Как использовать чат-ботов: сценарии и примеры [Электронный ресурс] // URL: https://www.uplab.ru/blog/how-to-use-chat-bots/ (дата обращения: 20.06.2020).
5. Пять самых успешных чат-ботов в бизнесе [Электронный ресурс] // URL: https://www.vedomosti.ru/partner/characters/2018/03/01/752546-pyat-uspeshnih (дата обращения: 20.06.2020).
6. Протокол шифрования MTProto [Электронный ресурс] // URL: https://ru.wikipedia.org/wiki/MTProto (дата обращения 12.05.2020).
7. Федеральные государственные образовательные стандарты [Электронный ресурс] // URL: https://fgos.ru/ (дата обращения 12.05.2020).
8. What is a Chatbot and How to Use It for Your Business [Электронный ресурс] // URL: https://medium.com/swlh/what-is-a-chatbot-and-how-to-use-it-for-your-business-976ec2e0a99f (дата обращения 13.05.2020).
9. Аббакумов А. А., Сидоров Д. П., Егунова А. И. Использование мессенджеров для информирования слушателей учебных заведений // Образовательные технологии и общество. – 2018. – Т. 21. – № 3. –   
   С. 330-336.
10. Гатулин Р. Р., Колупаева Д. А. Использование мессенджера TELEGRAM для реализации технологии электронного обучения в вузе // Санкт-Петербургский образовательный вестник. – 2017. – № 11-12 (15-16). –   
    С. 31-33.
11. Маликова В. Н., Перевощикова Е. Н. Использование программы Telegram как средства мобильного обучения // Инновационные тенденции развития системы образования : материалы VI Междунар. науч.-практ. конф. (Чебоксары, 19 февр. 2017 г.) / редкол.: О. Н. Широков [и др.]. – Чебоксары: ЦНС «Интерактив плюс», 2017. – С. 75-78.
12. Telegram Bot API [Электронный ресурс] // URL: https://core.telegram.org/bots/api (дата обращения 13.05.2020).
13. Polling vs webhook [Электронный ресурс] // URL: https://blog.cloud-elements.com/webhooks-vs-polling-youre-better-than-this (дата обращения 16.05.2020).
14. Aiogram documentation [Электронный ресурс] // URL: https://aiogram.readthedocs.io/en/latest/index.html (дата обращения 15.05.2020).
15. Что такое формат JSON и для чего он используется [Электронный ресурс] // URL: https://semantica.in/blog/format-json-chto-eto-takoe-kak-sozdat-i-otkryt-fajl.html (дата обращения 17.05.2020).
16. Справочная информация по Sqlite [Электронный ресурс] – режим доступа: https://ru.wikipedia.org/wiki/SQLite (дата обращения 23.05.2020)
17. Моргунов Е. П. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов, Е. В. Рогова, П. В. Лузанова. – СПб.: БХВ-Петербург, 2018. – 336 с.