

Министерство образования Пензенской области

ГБОУ ПО «Академический лицей № 14»

Проект

для участия в VI открытый региональный конкурс
исследовательских и проектных работ школьников

«Высший пилотаж - Пенза» 2024

на тему:

«Олимпиадный хронограф»

Выполнили:

Пугачев Александр, Юдин Павел

Научный руководитель:

Трофимов Юрий Александрович

Пенза, 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ЧАСТЬ 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОЗДАНИЯ СЕРВИСА «ОЛИМПИАДНЫЙ ХРОНОГРАФ»	5
1.1. Определение и эволюция понятия виртуального помощника	5
1.2. Основные направления информационной поддержки участников олимпиад	6
1.3. Формирование требований к приложению.....	8
1.4. Выбор мессенджера	8
ЧАСТЬ 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СЕРВИСА «ОЛИМПИАДНЫЙ ХРОНОГРАФ»	10
2.1. Описание архитектуры приложения	10
2.2. Реализация парсера для получения информации с официальных источников.....	10
2.3. Внешний вид системы	11
ЗАКЛЮЧЕНИЕ	15
СПИСОК ЛИТЕРАТУРЫ.....	16
ПРИЛОЖЕНИЕ 1	17
ПРИЛОЖЕНИЕ 2	21

ВВЕДЕНИЕ

Олимпиады – это интеллектуальные соревнования для обучающихся разных возрастов. Только в России их проводится более 700. Форматов очень много: индивидуальные и командные, очные и заочные, квесты, регаты, карусели, бои и т.д. [1] Олимпиады способствуют выявлению и поддержке талантливых молодых людей, что в свою очередь стимулирует развитие индустрии технологий и научных исследований в этой области. Участие в подобных мероприятиях способствует повышению образовательного уровня и развитию навыков участников. Это позволяет им расширить свой кругозор, развить логическое мышление и навыки решения сложных задач, что является крайне важным в современном высокотехнологичном обществе.

Для участников всегда является актуальным вопрос о своевременном информировании обо всех событиях на олимпиаде. Отслеживание расписания, участников, последних результатов и актуального времени до начала или окончания олимпиады позволяет участникам, наставникам, зрителям и организаторам быть в курсе происходящего и делает опыт участия более полноценным и интересным. При этом очень сложно держать в голове и контролировать все события даже для одной олимпиады (многие олимпиады проходят в несколько туров, на каждый тур нужно вовремя зарегистрироваться, время проведения может меняться, какие-то туры очные, а какие-то проходят в дистанционном режиме и т.п.). Если необходимо участие в нескольких олимпиадах, то без специальных средств отследить все события становится просто невозможным.

Именно для решения этих проблем мы и разработали проект «Олимпиадный хронограф», который не только напоминает об очередном событии, но также упорядочивает информацию и делает участие в олимпиаде более эффективным и увлекательным для всех заинтересованных сторон.

Цель проекта – создать виртуального помощника (чат-бота), который будет отображать информацию о контрольных строках всех основных олимпиадных событий, и напоминать об этом пользователям. Программа будет предоставлять актуальное расписание событий, информацию об участниках, о результатах, и будет отображать актуальное время до начала или окончания олимпиады и регистрации на нее.

Исходя и поставленной цели, были определены следующие **задачи проекта**:

1. Изучить материалы по теме проекта.
2. Разработать графический интерфейс, который будет отображать информацию о начале или окончании олимпиады, включая дату, время, местоположение и другие детали.
3. Разработать таймер, который отсчитывал бы время до начала или до окончания олимпиады.
4. Интегрировать внешние источники данных, такие как официальные сайты олимпиад и календари событий, для автоматического обновления информации о времени начала и окончания.
5. Разработать функционал напоминаний для пользователей.
6. Протестировать работу приложения.
7. Разработать документацию для описывания функционала и использования хронографа.

Методы исследования: анализ литературы, моделирование, статистико-математический, экспериментальный, наблюдение.

Практическая значимость проекта: разработанное устройство сможет облегчить повседневный тяжёлый труд хирургов, избавив их от надобности постоянно проверять показатели пациента во время операций. Данные очки могут применяться не только в области хирургии, но и в любых других областях, где имеется необходимость контроля пульса.

Целевая аудитория: сервис «Олимпиадный хронограф» может использоваться в образовательной сфере как эффективный инструмент отслеживания всех событий, связанных с олимпиадами, конкурсами и пр.

В перспективе планируется расширить функционал сервиса за счет добавления отслеживания других видов активностей (международные олимпиады, конкурсы, спортивные соревнования и пр.).

ЧАСТЬ 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОЗДАНИЯ СЕРВИСА «ОЛИМПИАДНЫЙ ХРОНОГРАФ»

1.1. Определение и эволюция понятия виртуального помощника

Виртуальный помощник, также называется чат-ботом (англ. chatbot) – это программа, помогающая выяснить и удовлетворить все потребности того или иного пользователя. Общение с чат-ботами происходит автоматически, непосредственно, при помощи текста или голоса. Виртуальный собеседник обрабатывает лексические данные, предоставляя клиенту логически верные и полезные ответы. Каждый чат-бот ставит своей целью упростить онлайн-общение с пользователями, предоставить им актуальную и достоверную информацию в оперативном режиме. Чат-боты могут вести общение с клиентами от лица той или иной компании или бренда, или учреждения. Программа-собеседник позволяет заменять переписку с живым человеком, помогает быстрее получить нужную пользователю информацию, не прибегая к звонкам операторам.

Несмотря на тот факт, что в настоящее время чат-боты получают очень большую популярность и не перестают стремительно распространяться, их нельзя назвать современным изобретением. Первые виртуальные собеседники появились ещё очень давно, около полувека назад и развивались, дорабатывались и изменялись на протяжении долгого времени.

С течением времени также появилась потребность в системе, которая способна обработать, проанализировать и извлечь необходимую информацию из большого массива данных. Так, в 2006 года IBM принялась за разработку суперкомпьютера, оснащенного системой искусственного интеллекта, получившего название Watson. В итоге ими был реализован чат-бот, получивший способность обрабатывать вопросы, сформулированные на естественном языке. Таким образом он может мгновенно ответить на практически любой вопрос, например, как открыть счет в банке. Сегодня усовершенствованное решение от IBM управляет когнитивными центрами информационной безопасности.

Считается, что этот чат-бот является предтечей Siri от Apple.

В 2010 году компания Apple запустила виртуальный голосовой помощник Siri. Этот голосовой помощник, умеющий подстраиваться под определённого пользователя, стал первым, доступным во всём мире. Siri является помощником для решения многих задач бытового плана, например, она помогает составлять и следить за расписанием дня конкретного пользователя, предоставляет просмотр новостей и узнавать нужную информацию о транспортном графике и погодных условиях того или иного города. Siri также может помочь в идентификации песен, которые воспроизводятся на телефоне, и проложить маршрут в нужную точку города или даже страны.

Компания Google вслед за Apple выпустила Google Now в 2012 году. В 2014 году были представлены Microsoft Cortana и Amazon Alexa.

В 2017 году «Яндекс» представил русскоязычного голосового помощника «Алиса». С данным виртуальным собеседником можно общаться как посредством голоса, так и посредством ввода запросов через клавиатуру. Алиса отвечает пользователю разными способами: показывает ответы непосредственно в диалоговом интерфейсе, выводит поисковую выдачу по запросам в браузере или же открывает нужное приложение, которое установлено на устройство. Кроме вопросно-ответной формы, этот чат-бот также может

решать и другие задачи: включать музыку и фильмы, ставить будильник и таймер, заказывать такси и открывать определенные программы.

С середины 2018 года виртуальный собеседник Алиса интегрирован при помощи Yandex.io в пользовательскую электронику.

На данном этапе развития технология виртуальных собеседников проникла в множество сфер жизни обычных людей. Чат-боты разного уровня интеллектуального развития используются повседневно, помогают отвечать на часто задаваемые вопросы клиентов компаний, являются справочными системы некоторых других учреждений. В основном пользователи ценят быстроту и доступность получаемой информации, также отмечается удобство получения нужного набора данных в любое время суток.

Первыми в России технологию чат-ботов для бизнеса начали использовать крупные компании банковского сектора, затем большинство предприятий IT-сектора, и телеком-компаний.

Таким образом, благодаря стремительному развитию технологий и активной компьютеризации, такие технологии как виртуальные собеседники получают широкое распространение. Большое количество людей в повседневной жизни всё чаще прибегает к их помощи во много благодаря тому, что данная технология является по-настоящему удобной в использовании. История развития, признание пользователями и различные подходы к реализации делают технологию широко применимой, в том числе, и к предметной области, краткое описание которой приведено в введении.

1.2. Основные направления информационной поддержки участников олимпиад

Развитие школьного олимпиадного движения обусловлено не только престижем участия в олимпиаде, но и возможностью получения учащимися бесценного познавательного опыта. Победа в олимпиаде позволяет получить существенные преимущества при поступлении в вуз в виде льгот или автоматического зачисления без вступительных экзаменов. Победителей крупных Всероссийских олимпиад ждут еще более заманчивые перспективы: участие в научно-исследовательской работе, в том числе и зарубежном, а в дальнейшем и устройство на интересную, высокооплачиваемую работу.

Для государства, в свою очередь, олимпиадное движение – это прекрасная возможность отбора и сопровождения одаренных детей в образовательном процессе.

Одним из приоритетных направлений развития Всероссийского олимпиадного движения является создание равных возможностей участия в олимпиаде для школьников всех регионов Российской Федерации. Эту задачу способно решить своевременное информирование о проведении олимпиад учащихся труднодоступных регионов [2].

Таким образом, основной целью олимпиадного движения является выявление талантливых детей за счет обеспечения доступности информации о проведении олимпиады, формах участия и преимуществах участия в олимпиаде.

В задачи информационной кампании по освещению олимпиадного движения входят:

- организация обратной связи с целевой аудиторией посредством спецпроектов, конкурсов и разъяснительной работы в социальных сетях;

- обеспечение взаимодействия со средствами массовой информации по информационному сопровождению олимпиадного движения;
- создание условий для информационной доступности сведений о ходе олимпиадного движения на сайте <http://rosolymp.ru> («РосОлимп»).

Для выполнения поставленных задач были проведены следующие масштабные мероприятия:

- публикации в разделах интернет-портала Всероссийской олимпиады школьников <http://rosolymp.ru> («РосОлимп»);
- публикации информационных материалов в печатных и электронных специализированных средствах массовой коммуникации;
- публичные мероприятия с участием прессы;
- проведение радиоэфиров с руководителями и участниками сборных команд, а также с победителями прошлых лет;
- официальные аккаунты олимпиадного движения в социальных сетях;
- онлайн-трансляции в социальных сетях;
- печатные материалы, содержащие значимую информацию о Всероссийском олимпиадном движении школьников.

Участие в олимпиадах требует от каждого человека развития навыка анализа большого количества информации. Одна из основных проблем состоит в динамично изменяющихся правилах и сроках проведения. В связи с этим каждый год участникам приходится изучать доступные информационные ресурсы, чтобы не пропустить очередное событие в конкретной олимпиаде.

Поскольку объём данных слишком велик для их ручной обработки и анализа, на помощь всем участникам олимпиадного движения могут прийти системы информационной поддержки, основными задачами которых являются: аккумулирование наиболее важной информации и её предоставление в удобной для восприятия форме. Также такие системы могут обладать некоторыми средствами интеллектуального анализа, работающими на основе данных о текущей ситуации, связанной с конкретной олимпиадой.

Сейчас существуют ряд проектов, предоставляющие информацию о большинстве олимпиад России, например порталы «Олимпиада.ру» и «РусОлимп». К сожалению, данные информационные порталы имеют ряд существенных недостатков, преодолеть которые невозможно в силу масштабности решаемой задачи. Так, одной из нерешаемых задач является своевременное информирование всех участников олимпиадного движения (учеников, наставников, руководителей образовательных организаций) обо всех изменениях в сроках проведения.

Таким образом, можно сделать вывод, что справочную информацию проще всего получить с помощью виртуального собеседника (чат-бот), потому что он может быстро и чётко в интерактивном режиме дать ответ на интересующий человека вопрос.

1.3. Формирование требований к приложению

На основе анализа основных направлений информационной поддержки участников олимпиад становится возможно сформулировать основной перечень требований, предъявляемых к приложению.

Необходимо спроектировать и реализовать базу данных, способную хранить важную для пользователей информацию. Хранимые данные должны включать в себя сведения об олимпиадах (id олимпиады, название, предмет, класс, уровень, рейтинг и т.п.), также сведения о выбранных для участия олимпиадах (id пользователя, id олимпиады, даты основных событий, напоминания и т.п.).

Основное приложение является скриптовым ботом, реализованным на основе одной из существующих библиотек для взаимодействия с API мессенджера. Архитектура разработанного приложения должна быть организована таким образом, чтобы можно было легко добавлять функционал (меню, подменю, разделы справки). При этом должен быть удобный и понятный диалог с пользователем, содержащий всю необходимую информацию.

В результате, должно быть создано гибкое, легко расширяемое приложение, которое можно эффективно поддерживать, направленное ключевым образом на информационную поддержку участников олимпиад и предоставляющее все необходимые сведения.

1.4. Выбор мессенджера

В настоящее время существует достаточно большое количество мессенджеров, которые пользуются популярностью в обществе. Принимая во внимание данный факт, становится очевидным, что идея создания чат-бота неразрывно связана с мессенджерами, так как данные приложения, ориентированные на ежедневное использование весомым количеством людей, как нельзя лучше подходят для реализации виртуальных собеседников.

Среди имеющихся на данный момент мессенджеров можно выделить Telegram. Это приложение можно охарактеризовать как самое быстро растущие по числу активных пользователей. Так, аудитория мессенджера в апреле 2020 года перешагнула отметку в 400 миллионов пользователей, январе 2021 года данная цифра уже преодолела отметку в 500 миллионов, а в 2023 году Telegram посещают 800 миллионов человек в месяц. По количеству аудитории Telegram входит в пятерку самых популярных мессенджеров в мире. При такой тенденции к росту невозможно игнорировать возрастающую с каждым днем популярность Telegram.

Ещё одним немаловажным фактором в пользу использования Telegram в качестве платформы для реализации чат-бота является то, что основной частью аудитории данного мессенджера являются молодые люди. Данный факт позволяет говорить об ориентированности данного мессенджера на целевую аудиторию разрабатываемого приложения.

К тому же Telegram обладает достаточно развитым средством создания чат-ботов – Telegram Bot API, позволяющим удобную и быструю разработку ботов. Библиотеки

взаимодействия с Telegram Bot API реализованы для большинства популярных языков программирования.

Следующим значительным преимуществом Telegram является простая регистрация ботов в системе и доступность созданных ботов для всех пользователей сразу, то есть отсутствие необходимости верификации. Этот подход позволит быстрое развертывание системы информационной поддержки участников олимпиад.

ЧАСТЬ 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СЕРВИСА «ОЛИМПИАДНЫЙ ХРОНОГРАФ»

2.1. Описание архитектуры приложения

В соответствии с сформированными ранее требованиями необходимо было разработать скриптовый чат-бот для информационной поддержки участников олимпиад. Получившийся в результате виртуальный собеседник имеет достаточно гибкую архитектуру, благодаря которой возможно удобно сопровождать приложение, а также добавлять новые ветки диалогов.

В качестве основного средства взаимодействия с API Telegram будет выступать библиотека aiogram [5].

Основная идея реализации состоит в том, что чат-бот основан на таком часто используемом инструменте мессенджера Telegram, как обработчик сценария диалогов (ConversationHandler). Сценарий представляет собой серию вопросов или реплик, в которой бот хранит: состояние диалога, а также предыдущие ответы пользователя. Таким образом, можно посылать ряд заранее определенных последовательных сообщений.

Ещё одним не менее важным инструментом, реализованным в Telegram, является обработчик клавиатур (наборов интерактивных кнопок), привязанных к определенному сообщению (InlineKeyboardMarkup). Благодаря этому становится возможным организовать что-то наподобие всем понятного и привычного меню, интегрированного в диалоговое окно. На нажатие пользователем кнопок на таких клавиатурах реагирует программа и обновляет уже посланное или посылает новое сообщение. В данном случае в архитектуре приложения предусмотрено начинать встроенные в сценарии диалогов верхнего уровня вложенные сценарии, что упрощает процесс добавления нового сценария диалога в общую структуру за счет её иерархического строения.

Отдельные сценарии удобно подключать или изменять, так как всё, что касается одной ветки диалога, собрано в отдельный класс (класс-обработчик). Каждый из таких классов-обработчиков предоставляет для главного класса, запускающего чат-бота, стандартный интерфейс своего создания и дальнейшего использования.

Другой важной особенностью созданного приложения является наличие конфигурационного файла, который хранит токен чат-бота и, возможно, путь до файла с БД. Это делает возможным запуск приложения от имени любого бота, зарегистрированного в базе ботов Telegram. Также конфигурационный файл может хранить в формате json вид сохраняемой и выводимой на экран истории взаимодействия с ботом (так называемых логов).

С полным текстом программы, реализующей чат-бота, можно ознакомиться в Приложении 1.

2.2. Реализация парсера для получения информации с официальных источников

Веб-скрапинг – это процесс автоматического сбора информации из онлайн-источников. Для выбора нужных сведений из массива «сырых» данных, полученных в ходе скрапинга, нужна дальнейшая обработка – парсинг. В процессе парсинга выполняются синтаксический анализ, разбор и очистка данных. Результат парсинга –

очищенные, упорядоченные, структурированные данные, представленные в формате, понятном конечному пользователю.

Для реализации веб-скрапинга и парсинга воспользуемся возможностями библиотеки BeautifulSoup. Она позволяет получать нужную информацию с нестандартных страниц с произвольным синтаксисом [4].

В качестве основного ресурса для получения информации об олимпиадах будем использовать портал <https://olimpiada.ru/> – наиболее полный и структурированный сайт с перечнем основных внутренних олимпиад [3]. Основной веб-скрапинг происходит на странице <https://olimpiada.ru/activities>.

```
url = 'https://olimpiada.ru/activities'  
response = requests.get(url)  
soup = BeautifulSoup(response.content, 'html.parser')
```

Извлечение нужных данных из HTML разметки происходит следующим образом:

```
items = soup.find_all("div", class_="o-block")
```

Затем идет парсинг данных, соответствующих каждой олимпиаде, и их запись в базу данных.

```
for item in items:  
    subject = item.find(class_='subject_tag').get_text()  
    form = item.find(class_='classes_dop').get_text()  
    rating = item.find(class_='pl_rating').get_text()  
    title = item.find(class_='headline').get_text()  
    time = item.find(class_='headline red')
```

...

С полным текстом парсера можно ознакомиться в Приложении 2.

2.3. Внешний вид системы

На рисунке 1 изображено начало диалога с ботом. Запуск сеанса диалога происходит при вводе команды /start, её Telegram предлагает ввести при первом открытии чата с ботом. В ответ на данную команду бот отправляет приветственное сообщение, а также сообщение с привязанными к нему кнопками, которые представляют собой главное меню.



Рисунок 1. Начало диалога с ботом

В результате нажатия на кнопку «Олимпиады» происходит переход к вложенному меню (рисунок 2). Плюсом такого построения диалога является освобождение пользователя от необходимости набирать текст, что будет являться несомненным удобством для тех пользователей, которые в основном пользуются приложением через мобильные устройства.

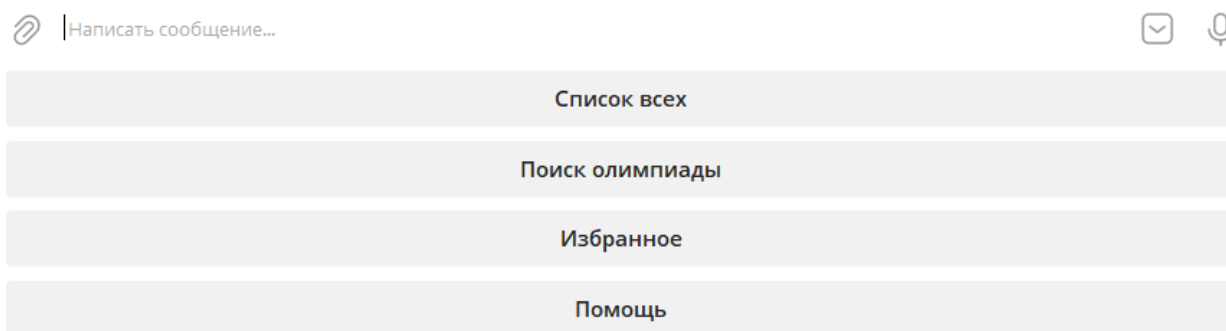


Рисунок 2. Вложенное меню

Рисунок 3 демонстрирует сообщение с выводом перечня олимпиад. Каждая олимпиада выводится в виде инлайн-кнопки, нажав на которую пользователь может увидеть подробную информацию – предмет, класс, контрольные даты, ссылку на сайт и пр. Так как список олимпиад достаточно большой, для удобства он делится на группы по 10 олимпиад с системой перехода между группами.

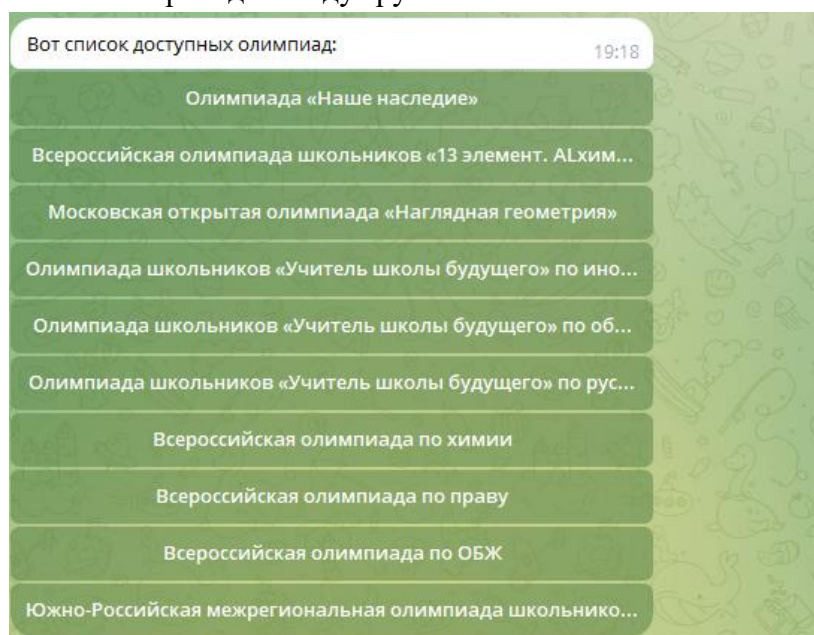


Рисунок 3. Вывод списка олимпиад

Рисунок 4 демонстрирует сообщение с выводом основной информации о выбранной олимпиаде. Вся важная информация, которая обычно интересует пользователей, оказывается собранной в одно сообщение, что, вне всякого сомнения, является достаточно практичным. Так же в нем присутствует ссылка на сайт олимпиады, если участник захочет ознакомиться с ней более подробно, и возможность добавить олимпиаду в избранное.

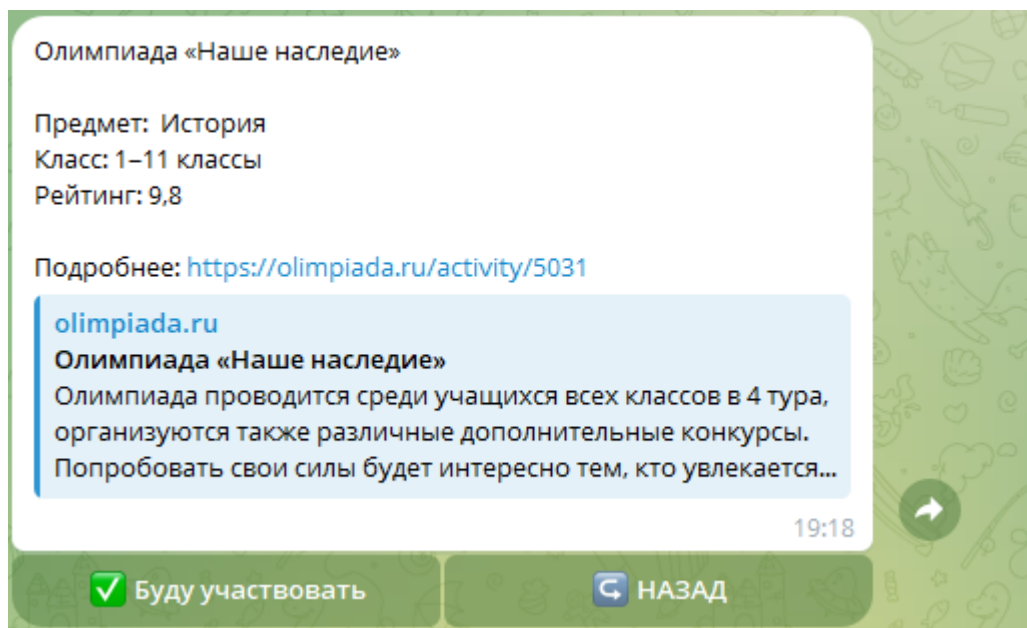


Рисунок 4. Экран описания олимпиады

Основной «фишкой» данного бота является меню «Избранное», где отображаются только те олимпиады, в которых пользователь собирается принять участие, и настраивается система оповещений (напоминаний) по каждой олимпиаде индивидуально. При нажатии на кнопку «Избранное», пользователь видит список «своих» олимпиад (рис. 5).

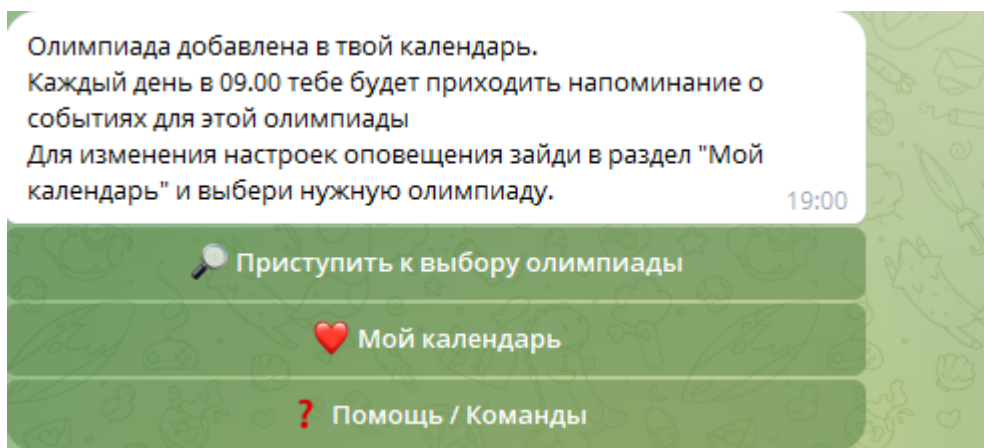


Рисунок 5. Отображение избранных олимпиад

При выборе нужной олимпиады пользователь может отказаться от участия или настроить систему оповещений (рис. 6).

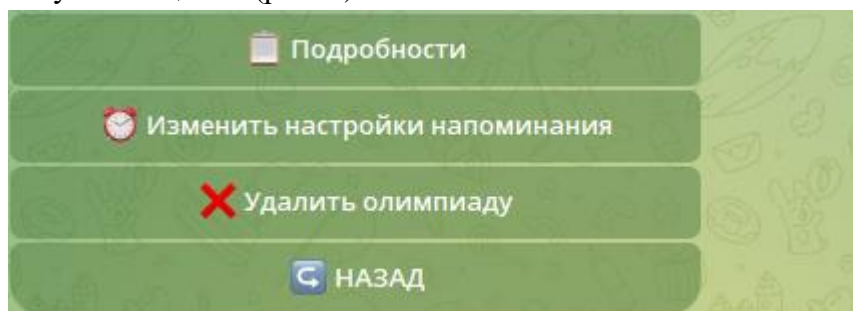


Рисунок 6. Настройка олимпиад в избранном

По умолчанию, при занесении олимпиады в избранное, чат-бот начинает отправлять напоминания о контрольных сроках (регистрация, первый тур, второй тур и т.п.) каждый день в 09.00 индивидуально каждому пользователю.

Для получения справочной информации о функционале чат-бота пользователь может воспользоваться командой /help или нажать соответствующую кнопку в главном меню (рис. 7).

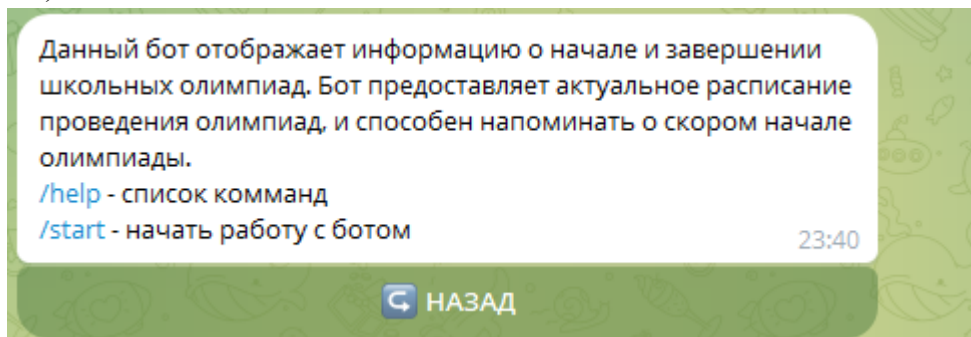


Рисунок 7. Экран справки и помощи

Таким образом, все функции разработанного чат-бота работают корректно. Реализованная архитектура, построенная на интерактивных меню, является гибкой и легко расширяемой. Добавив альтернативный парсер и кнопки меню, можно наполнить чат-бота новым функционалом, например, отслеживание спортивных соревнований, конкурсов творческой самодеятельности и т.п. Это означает достижение поставленной в работе цели.

ЗАКЛЮЧЕНИЕ

Проект «Олимпиадный хронограф» остается актуальным и значимым в современной цифровой эпохе. В силу растущей популярности информационных технологий и цифровой образовательной среды, заинтересованность в участии в олимпиадах продолжает расти. Программное обеспечение «Олимпиадный хронограф» предоставит необходимую информацию о событии, упорядочит расписание и результаты, и поможет участникам и зрителям быть в курсе последних новостей.

Таким образом, проект «Олимпиадный хронограф» остается актуальным, поскольку поддерживает важность и значимость олимпиад в обществе, а также способствует развитию информационной грамотности и цифровых навыков.

СПИСОК ЛИТЕРАТУРЫ

1. Что такое олимпиады [Электронный ресурс]. URL: <https://olimpiada.ru/article/661> (Дата обращения: 04.12.2023).
2. Освещение олимпиадного движения в России: актуальность, инструменты и итоги информационной кампании [Электронный ресурс]. URL: https://vercont.ru/stati_i_obzory/osveshchenie_olimpiadnogo_dvizheniya_v_rossii_aktualnost_inst.html (Дата обращения: 05.12.2023).
3. Олимпиада.ру – официальный портал [Электронный ресурс]. URL: <https://olimpiada.ru/> (дата обращения: 15.11.2023) (Дата обращения: 05.12.2023).
4. Официальная документация по Beautiful Soup [Электронный ресурс]. URL: <https://beautiful-soup-4.readthedocs.io/en/latest/> (Дата обращения: 05.12.2023).
5. Пишем Telegram-ботов с aiogram 3.x [Электронный ресурс]. URL: <https://mastergroosha.github.io/aiogram-3-guide/> (Дата обращения: 05.12.2023).

Листинг программного кода чат-бота

```

import asyncio
import logging
import sys
import random
from os import getenv
import os
import aiocron
from apscheduler.schedulers.asyncio import AsyncIOScheduler
from datetime import datetime
from aiogram.client import bot
from dotenv import load_dotenv
from aiogram import Bot, Dispatcher, Router, types, F
from aiogram.enums import ParseMode
from aiogram import Bot, Dispatcher, Router, types, F
from aiogram.enums import ParseMode
from aiogram.filters import CommandStart, Command
from aiogram.types import Message, KeyboardButton, ReplyKeyboardMarkup,
ReplyKeyboardRemove, InlineKeyboardButton, InlineKeyboardMarkup
from aiogram.utils.keyboard import InlineKeyboardBuilder
from aiogram.utils.markdown import hbold
import olymp_parser

data = None
favor_data = {}
user_id = None

# load_dotenv()
TOKEN = open('config.txt').readline().strip
dp = Dispatcher()

@dp.callback_query(F.data.startswith("1_https://"))
async def olymp_like(callback: types.CallbackQuery):
    global data, favor_data, user_id
    info = callback.data.split('_')[1]
    for line in data:
        if line['link'] == info:
            if user_id in favor_data:
                favor_data[user_id].append(info)
            else:
                favor_data[user_id] = [info]
            break
    print(favor_data)
    await callback.message.answer('Олимпиада добавлена в твой
календарь.\nКаждый день в 09.00 тебе будет приходить напоминание о событиях
для этой олимпиады\nДля изменения настроек оповещения зайди в раздел "Мой
календарь" и выбери нужную олимпиаду.', reply_markup=builder1.as_markup())

@dp.callback_query(F.data.startswith("https://"))
async def olymp_about(callback: types.CallbackQuery):
    global data

```

```

info = callback.data
for line in data:
    if line['link'] == info:
        break
    rep = line['title'] + '\n\n' + 'Предмет: ' + line['subject'] + '\n' +
'Класс: ' + line['form'] + '\n' + 'Рейтинг: ' + line['raiting'] + '\n\n' +
'Подробнее: ' + line['link']
    builder = InlineKeyboardBuilder()
    builder.row(types.InlineKeyboardButton(
        text='👉 Буду участвовать',
        callback_data='1_' + line['link']), types.InlineKeyboardButton(
        text='👈 НАЗАД👈',
        callback_data='start2')
    )
    await callback.message.answer(rep, reply_markup=builder.as_markup())

@dp.callback_query(F.data == "start2")
async def filter(callback: types.CallbackQuery):
    global data
    if not data:
        data = olymp_parser.get_data()
    builder = InlineKeyboardBuilder()
    for line in data[:10]:
        builder.row(types.InlineKeyboardButton(
            text=line['title'],
            callback_data=line['link'])
        )
    await callback.message.answer('Вот список доступных олимпиад:',
        reply_markup=builder.as_markup())

@dp.callback_query(F.data == "start3")
async def favorite(callback: types.CallbackQuery):
    await callback.message.answer('""Здесь переменная с названием
олимпиады""',
        reply_markup=podr.as_markup())

@dp.callback_query(F.data == "start4")
async def send_random_value(callback: types.CallbackQuery):
    await callback.message.answer('""Данный бот отображает информацию о
начале и завершении школьных олимпиад. Бот предоставляет актуальное
расписание проведения олимпиад, и способен напоминать о скором начале
олимпиады.
/help - список команд
/start - начать работу с ботом""',
        reply_markup=back1.as_markup())

@dp.callback_query(F.data == "start001")
async def start01(callback: types.CallbackQuery):
    await callback.message.answer("👉 Выберите, что вам нужно:",
        reply_markup=builder1.as_markup())

back1 = InlineKeyboardBuilder()
back1.row(types.InlineKeyboardButton(text='👈👈 НАЗАД👈👈',
callback_data='start001'))

```

```

kclass = InlineKeyboardBuilder()
kclass.row(types.InlineKeyboardButton(text='1',
callback_data='start21'), (types.InlineKeyboardButton(text='2',
callback_data='start21')), (types.InlineKeyboardButton(text='3',
callback_data='start21')))
kclass.row(types.InlineKeyboardButton(text='4',
callback_data='start21'), (types.InlineKeyboardButton(text='5',
callback_data='start21')), (types.InlineKeyboardButton(text='6',
callback_data='start21')))
kclass.row(types.InlineKeyboardButton(text='7',
callback_data='start21'), (types.InlineKeyboardButton(text='8',
callback_data='start21')), (types.InlineKeyboardButton(text='9',
callback_data='start21')))
kclass.row(types.InlineKeyboardButton(text='10',
callback_data='start21'), (types.InlineKeyboardButton(text='11',
callback_data='start21')))

podr = InlineKeyboardBuilder()
podr.row(types.InlineKeyboardButton(text='⏪                Подробности',
callback_data='start31'))
podr.row(types.InlineKeyboardButton(text='⏪ Изменить настройки напоминания',
callback_data='start32'))
podr.row(types.InlineKeyboardButton(text='⏪                Удалить                олимпиаду',
callback_data='start33'))
podr.row(types.InlineKeyboardButton(text='⏪                НАЗАД',
callback_data='start001'))

subject = InlineKeyboardBuilder()
subject.row(types.InlineKeyboardButton(text="Русский                язык",
callback_data="start22"), (types.InlineKeyboardButton(text='Математика',
callback_data='start22')))
subject.row(types.InlineKeyboardButton(text="Информатика",
callback_data="start22"), (types.InlineKeyboardButton(text='Английский язык',
callback_data='start22')))
subject.row(types.InlineKeyboardButton(text="Обществознание",
callback_data="start22"), (types.InlineKeyboardButton(text='История',
callback_data='start22')))
subject.row(types.InlineKeyboardButton(text="Биология",
callback_data="start22"), (types.InlineKeyboardButton(text='Химия',
callback_data='start22')))
subject.row(types.InlineKeyboardButton(text="Физика",
callback_data="start22"), (types.InlineKeyboardButton(text='География',
callback_data='start22')))
subject.row(types.InlineKeyboardButton(text='⏪                НАЗАД',
callback_data='start001'))

builder1 = InlineKeyboardBuilder()
builder1.row(types.InlineKeyboardButton(text="⏪                Приступить к выбору
олимпиады", callback_data="start2"))
builder1.row(types.InlineKeyboardButton(text='❤️                Мой календарь',
callback_data="start3"))
builder1.row(types.InlineKeyboardButton(text='⏪                Помощь / Команды',
callback_data="start4"))

```

```

@dp.message(Command('help'))
async def command_start(message: Message):
    await message.answer('''Данный бот отображает информацию о начале
изавершении школьных олимпиад. Бот предоставляет актуальное расписание
проведения олимпиад, и способен напоминать о скором начале олимпиады.
/help-список команд
/start-начать работу с ботом''',
                        reply_markup=back1.as_markup())

@dp.message(Command('start'))
async def command_start(message: Message):
    # await message.answer_sticker('CAACAgIAAxkBAAECRVxlbOeBuYcU7hzZnw7VXQAB-
Svyx3sAAmcBAAIWQmsKc4m-zqWwACIzBA')
    await message.answer(f"□ {hbold(message.from_user.first_name)}, добро
пожаловать в наш Бот",
                        reply_markup=builder1.as_markup())

@dp.message()
async def command_understand(message: Message):
    await message.reply(f'''К сожалению я Вас не понимаю,
воспользуйтесь командой --> /help''')

async def send_message_cron(bot: Bot):
    if favor_data:
        for data in favor_data:
            user_id = data
            olymp = favor_data[data]
            # print(data, favor_data[data])
            await bot.send_message(user_id, f'''Напоминание!\n\nВы записаны
на следующие олимпиады''')

async def main():
    bot = Bot(TOKEN, parse_mode=ParseMode.HTML)
    scheduler = AsyncIOScheduler(timezone="Europe/Moscow")
    scheduler.add_job(send_message_cron, trigger='cron', hour=19, minute=20,
start_date=datetime.now(), kwargs={'bot': bot})
    scheduler.start()
    await dp.start_polling(bot)

if __name__ == "__main__":
    logging.basicConfig(level=logging.INFO, stream=sys.stdout)
    asyncio.run(main())
    aiocron.asyncio.get_event_loop().run_forever()

```

Листинг программного кода парсера

```

import requests
from bs4 import BeautifulSoup

def get_data():
    url = 'https://olimpiada.ru/activities'
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')
    items = soup.find_all("div", class_="o-block")
    data = []

    for item in items:
        subject = item.find(class_='subject_tag').get_text()
        form = item.find(class_='classes_dop').get_text()
        raiting = item.find(class_='pl_rating').get_text()
        title = item.find(class_='headline').get_text()
        time = item.find(class_='headline red')
        time = time.get_text() if time != None else ''
        desc = item.find(class_='none_a black olimp_desc')
        desc = desc.get_text() if desc != None else ''
        link = str(item.find(class_='none_a
black')).split()[3].split('=')[1][1:-1]

        #title = item.find("h2", class_="title").get_text()
        #date = item.find("time").get_text().split()[0]
        #link = item.a["href"]
        data.append({
            'subject': subject,
            'form': form,
            'raiting': raiting,
            'title': title,
            'time': time,
            'desc': desc,
            'link': 'https://olimpiada.ru' + link
        })

    return data

```

**Рецензия научного руководителя
на исследовательскую работу
«Олимпиадный хронограф»
(авторы: Пугачев Александр Дмитриевич, Юдин Павел Дмитриевич)**

**VI открытого регионального конкурса исследовательских и
проектных работ школьников «Высший пилотаж - Пенза» 2024**

Проект «Олимпиадный хронограф» представляет собой интересное и полезное решение для учащихся, готовящихся к школьным олимпиадам. Он помогает не забыть о предстоящих событиях и мероприятиях, напоминая о них в нужное время. Проект представляет собой чат-бота в мессенджере Телеграмм, что является удобным и практичным решением для всех участников.

Одной из особенностей проекта «Олимпиадный хронограф» является то, что напоминания могут быть настроены индивидуально для каждого ученика. Это позволяет учесть особенности каждого ребенка и сделать процесс подготовки к олимпиадам более эффективным. Кроме того, проект позволяет отслеживать результаты участия в олимпиадах, что помогает ученикам видеть свои успехи и неудачи. Это может мотивировать их на дальнейшее развитие и улучшение своих результатов.

Помимо непосредственно участников олимпиад, данный проект может быть интересен наставникам учащихся и кураторам, отвечающим за олимпиадную подготовку в образовательных организациях. Кроме того, проект «Олимпиадный хронограф» может быть легко расширен и на другие сферы научной, спортивной и общественной деятельности в образовательных организациях любого уровня и профиля.

На данный момент проект «Олимпиадный хронограф» не имеет аналогов, а его отличительными особенностями являются доступность (сервис абсолютно бесплатный), универсальность применения, удобство использования (минимум действий), автоматический режим работы, индивидуальная настройка.

«Олимпиадный хронограф» является полезным инструментом для подготовки к школьным олимпиадам и может существенно помочь ученикам в достижении успеха, что делает этот проект весьма актуальным и перспективным.

Научный руководитель



Ю.А. Трофимов