

GEEKBRAINS

Разработчик. Программист. Цифровые профессии

Чат-Бот для фанатов Formula 1.

Воропаев Олег Владимирович

Москва

2024

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	2
1. Теоретический анализ и постановка задач	3
1.1 Аудитория «Формулы 1»	3
1.2. Зачем нужен чат-бот для фанатов «Формулы-1»	4
1.3. Концептуальная задача чат-бота	4
2. Глава 1. Выбор языка и месенджера	6
2.1 Что такое чат-бот, их классификация и особенности	6
2.2 Особенности современных меседжеров, платформ	8
2.3. Сравнение языков программирования, как наиболее подходящих для написания бота	11
2.3.1 Что такое язык программирования и какими они бывают	11
2.3.2 Рейтинг языков программирования	12
2.3.3 Анализ и сравнение популярных языков программирования	14
2.3.4 Python	15
2.4.5 Java	15
2.4.6 C#	16
2.4.7 JavaScript	17
3. Глава 2. Практическая реализация чат-бота	18
3.1. Средства разработки чат-бота	18
3.1.1 Необходимые модули для разработки сервиса	18
3.1.2 Среда разработки	19
3.1.3 Система контроля версий	20
3.1.4 Telegram Bot API	22
3.2 Разработка телеграмм-бота	23
3.2.1 Регистрация чат-бота	23
3.2.2. Описание структуры бота	24
3.2.3 Некоторые примеры работы бота	26
3.2.4 Публикация (хостинг) бота	28
Заключение	30
Список литературы	32

1. Теоретический анализ и постановка задач

1.1 Аудитория «Формулы 1»

В 2021 г. трансляции 22 гонок «Формулы-1» посмотрели в общей сложности 1,55 млрд человек, что на 4% больше показателя 2020 г. Россия вошла в число стран, где максимально увеличилось количество зрителей, посмотревших хотя бы одну гонку в сезоне. В Испании это число выросло на 272%, в России — на 129%, в США — на 53%.¹

Количество фолловеров в соцсетях за сезон выросло на 40% и достигло 49,1 млн, число просмотров видеороликов поднялось на 50% до 7 млрд. Общая вовлеченность в Facebook, Twitter, Instagram, YouTube, Tiktok, Snapchat, Twitch и китайских социальных сетях Weibo, WeChat, Toutiao и Douyin выросла на 99% до 810 млн. (рис.1)

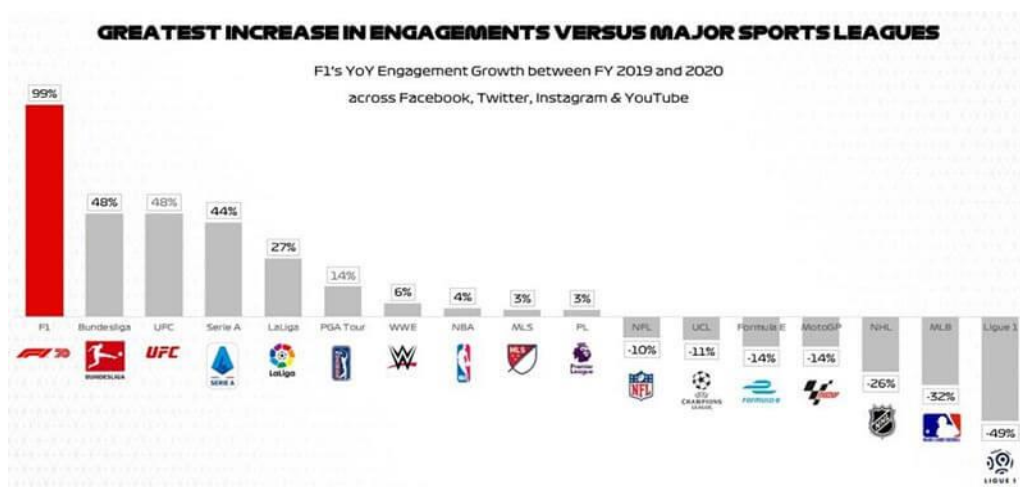


Рис.1 Рост вовлеченности аудитории Формула 1 в социальных сетях.

События 2022-2023 годов привели отказу руководства Формулы 1 в показе гонок и проведению Гран-При на территории России.² Вместе с тем, аудитория

¹ <https://www.formula1.com/en/latest/article.formula-1-announces-tv-race-attendance-and-digital-audience-figures-for-2021.1YDpVJI0HGNuok907sWcKW.html>

² https://www.championat.com/tags/20056formula1/?utm_source=Rambler&utm_medium=sport&utm_campaign=transition

данного вида спорта в России продолжает расти и имеется потребность в получении информации о предстоящих и прошедших событиях.

Зрители находят другие способы просмотра и не пропускают любимое соревнование. Вот лишь некоторые из вариантов:

Зарубежные телеканалы. Прямые трансляции гонок проводят многие каналы, включая Sky Sports F1, FOX Sports, ESPN, Setanta Sports. Большинство легально доступны в разных стриминговых сервисах, но для просмотра, скорее всего, понадобится VPN.

F1 TV — удобный ресурс с большим количеством дополнительного контента: онборды каждого пилота, телеметрия, архивные записи. Однако доступ к нему получить довольно сложно: помимо регионального ограничения сервис нельзя оплатить с российской карты — пригодится счет в иностранном банке или помощь друзей из-за рубежа.

Трансляции в интернете и соцсетях. При желании можно найти любительские эфиры, в том числе с комментариями на русском языке.

Таким образом, решение разработать чат-бота было продиктовано желанием вовлечь в «мир» формулы 1 большее количество людей из России, в настоящее время вынужденных затрачивать большое количество времени на получение информации, содержащейся во, все еще, открытых источниках.

1.2. Зачем нужен чат-бот для фанатов «Формулы-1»

Чат-бот о «Формуле 1» - идеальное решение для всех фанатов этого захватывающего спорта. Бот позволит быть в курсе последних новостей, расписания гонок и получить информацию о пилотах. Кроме информационно-развлекательной составляющей, чат-бот планируется использовать для предложения забронировать столик в кафе и ресторанах (осуществляющий трансляцию Гран-При), что, в конечном счете, вызовет его экономическую целесообразность.

1.3. Концептуальная задача чат-бота

Чат-бот должен предоставлять следующую информацию:

1. Последние новости:

- Бот обновляется в режиме реального времени, поэтому фанат всегда будет первым, кто узнает о последних событиях и разработках в мире «Формулы 1».

- Свежие новости о командах, гонках, изменениях в составе и других моментах.

2. Расписание гонок:

- Когда и где пройдет следующая гонка «Формулы 1».

- Бот должен предлагать полное и обновляемое расписание, позволяя быть готовым к каждой гонке.

3. Информация о пилотах:

- Информация о любимом пилоте или достижения новичков.

- Бот должен предоставлять полезную информацию о пилотах, их командах, статистике и других интересных фактах.

- Полная картина о каждом пилоте «Формулы 1».

4. Удобство и простота:

- Предоставление информации в удобном и доступном формате.

- Нет необходимости искать и открывать различные сайты или приложения - все, что нужно, можно получить в одном месте, просто введя соответствующую команду.

5. Персонализация:

- Бот должен позволять настроить уведомления и получать информацию только о тех новостях или гонках, которые интересуют.

В итоге, бот про «Формулу 1» предоставит фанатам этого захватывающего спорта удобный и непрерывный доступ к самой свежей информации. Даст возможность быть в курсе всех событий, получать полезную информацию и наслаждаться каждой гонкой «Формулы 1» дома или в кафе. Данный чат-бот должен предлагать посмотреть ближайшую гонку «Формулы 1» в кафе или ресторане, а также заказать там столик.

2. Глава 1. Выбор языка и месенджера

2.1 Что такое чат-бот, их классификация и особенности

Чат-бот — это программа, которая имитирует человеческое общение. Алгоритм бота ориентирован как на ведение неструктурированного диалога, так и на решение поставленных клиентом задач. Большое распространение боты получили в бизнесе, как в высококонкурентной среде, где особенно важна клиентская поддержка. Аналитики KPMG³ поставили чат-ботов на вторую строчку среди самых популярных инструментов цифровизации у российских компаний (на первом месте — большие данные и предиктивная аналитика).

Подходы и технологии, используемые при создании бота, зависят от типа и назначения системы. Классификация диалоговых систем опирается на 2 критерия:

- Ориентированность на задачу (goal-oriented) VS общее назначение
- Закрытая VS открытая предметная область (домен)

Goal-oriented боты разрабатываются для решения конкретных задач — выдачи информации по запросу, выполнения операций (проверка баланса, подключение тарифа). Диалоговые системы общего назначения (так называемые боты- «болталки») направлены исключительно на поддержание диалога и развлечение пользователя.

Системы с закрытым доменом выполняют задачи или ведут диалог, ограниченный узкой предметной областью, например банковской или медицинской сферой. Системы с открытым доменом предполагают универсальные возможности использования - к ним относятся ассистенты типа Алексы от Amazon или Алисы от Яндекса.

Сложность разработки чат-бота возрастает по мере расширения его предметной области и генерализации назначения (или увеличения количества

³ <https://ict.moscow/research/cifrovye-tehnologii-v-rossiyskih-kompaniyah/>

задач, которые он способен выполнять). На рисунке представлен пример классификации некоторых чат-ботов по данным критериям (рис. 2).

Чат-ботов можно разделить на некоторые категории, в зависимости от форматов взаимодействия с ним: Кнопочный чат-бот. Такой бот взаимодействует с пользователем посредством кнопок (уже готовых запросов), другими словами, связь весьма ограничена группой созданных предварительно фраз, нет возможности текстового ввода.

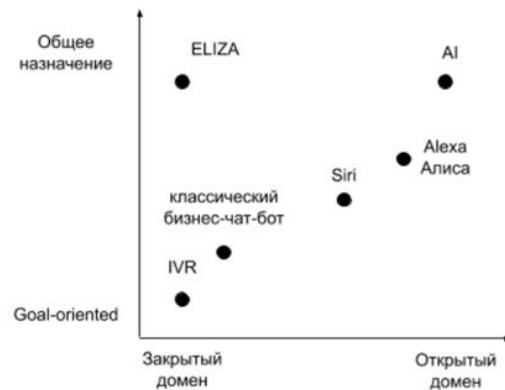


Рис.2 Классификация некоторых чат-ботов

Текстовый чат-бот. Данный вид уже более прогрессивный и функциональный: возникает возможность текстового ввода, происходит анализ запроса и выдается на его базу ответ.

Чат-бот является прикладной программой, которая логически преобразует информацию, полученную от пользователя, в нужный ему ответ. Предоставляет человеку возможность без участия живого человека по ту сторону, получить информацию, которая ему нужна. Сейчас уже создано огромное кол-во ботов для различных идей.

Ниже описаны несколько интересных представителей:

- чат-боты с различными играми;
- новостные чат-боты, которые оповещают о последних событиях;
- чат-боты для доставки продуктов, одежды или только что приготовленной пиццы;
- чат-боты для бронирования отелей или билетов на самолет;
- чат-боты юристы, такие могут помочь в консультации по банковским вопросам и т.п.

2.2 Особенности современных меседжеров, платформ

Своё начало общение в сети интернет берет с чатов, потом приоритет отдали мессенджерам, потом на первое место встали социальные сети, но не так давно мессенджеры снова возглавили этот список, став самым перспективным сервисом для общения. Это было подтверждено исследованиями компании «Yota» в 2022 году: количество пользователей, использующих мессенджеры в Российской Федерации, увеличилось на более чем 15%. А также, по средствам опроса пользователей, они определили, как формируется новый цифровой этикет. Не малая часть людей предпочитает голосовые сообщения обычным текстовым сообщениям⁴.

Мессенджеры пользуются популярностью в основном у людей среднего возраста, так на долю лиц младше 18 лет будет приходиться всего 8% от всего количества аудитории, пользователи в возрасте до 25 лет составляют 10%, от 25 до 35 лет 32%, от 35 до 45 лет – 26%. При этом 13% людей от 45 до 55 лет, а 8% - от 55 до 64 лет. И всего лишь 3% пользователей находится в возрасте старше 64-х лет (рис 3).



Ещё одной весомой причиной резкой популярности мессенджеров является изменение тарифов мобильного интернета. Из этих данных можно сделать вывод -

⁴ Резкий рост популярности российских мессенджеров: анализ интернет-трафика пользователей Yota 2022 год https://www.cnews.ru/news/line/2022-0317_rezkij_rost_populyarnosti

большая часть платежеспособной мобильной аудитории сейчас начинает активно использовать более современные средства связи.

По рис. 5 видно, что в России самыми популярными мессенджерами являются: WhatsApp, ВКонтакте, Telegram, последние места занимают Instagram* и Facebook*.

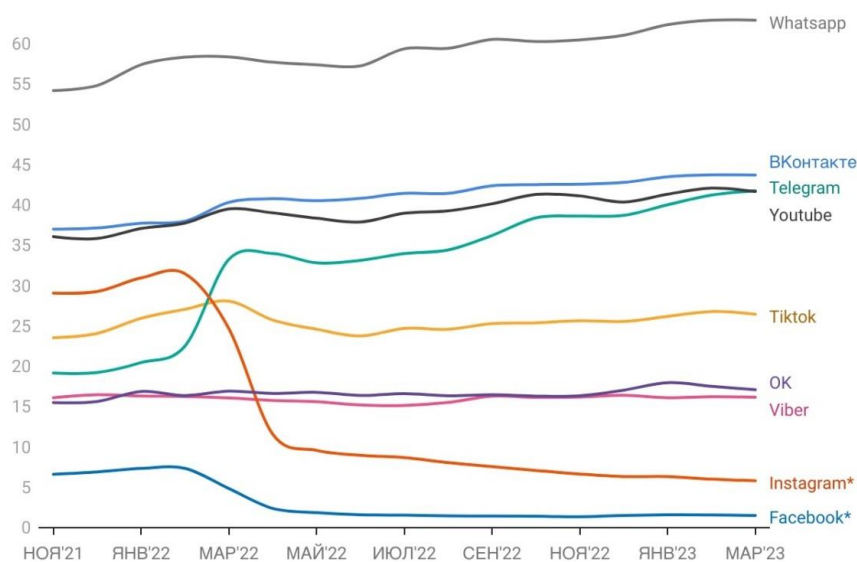


Рис.4 Среднесуточный охват менеджеров в России (% населения)

ВКонтакте - есть API, который позволяет ботам общаться со своими пользователями от имени сообщества. Вы можете отправлять текст, фотографии, файлы, местоположение и наклейки для клиентов. На других платформах чат ботов нет элементов взаимодействия: кнопок, элементов меню, расширений, шаблонов сообщений.

Telegram - одним из первых открыл возможность создания ботов в мессенджере. Всего, по данным Forbes, Telegram создал около 100 тысяч роботов. Примеры в России: чат-бот Сбербанка, OneTwoTrip_Bot, бот Почты России, публикации Медузы и так далее. Не так давно чат-роботы в Telegram научились принимать платежи от пользователей, в России Яндекс.Касса принимает платежи.

Для среднего и крупного бизнеса сделали WhatsApp Business API —

* Принадлежит компании Meta Platforms, признанной в России экстремистской организацией и запрещенной.

прямой доступ к API мессенджера, позволяющим реализовывать службы поддержки в мессенджере, ботов и т.п. Чат-бот в Telegram:

Основным преимуществом этого приложения для обмена сообщениями по сравнению с WhatsApp и Facebook Messenger является его API (интерфейс прикладного программирования), открытый для всех, который предлагает больше возможностей и позволяет реализовать множество креативных идей.

Telegram - отличная безопасная платформа для самых креативных и сложных роботов, например, для интеграции криптовалютных кошельков, обработки банковской информации и финансовых транзакций. Команда Telegram также подчеркивает простоту интеграции онлайн-запросов, игровых роботов HTML5 и пользовательских клавиатур. Примеры иностранных телеграммроботов: MeansBot, BBVA, XirkleBot. Чат-бот во «ВКонтакте»:

ВКонтакте обладает очень простой функциональностью. Вы можете создать примитивного чат-бота для аудитории, даже самостоятельно. Для этого есть специальные сервисы: Robochat, BotVK, Chatgun. Преимущества простой функциональности — это возможность создать полноценного чат-бота.

Чат-бот в WhatsApp:

API для этого мессенджера платный, а возможность платить за него не гарантирует доступ. Кроме того, чат-боты в этом приложении основаны на принципе отправки сообщений (нет никаких команд в виде кнопок). Фактически, это не чат-бот в классическом смысле этого слова, а платформа для автоматических сообщений и мгновенных приветствий. Все это делает WhatsApp наименее популярной платформой для запуска чат-бота. Мессенджер является относительно новичком в области чат ботов, решения внедряются медленно, поэтому невозможно предсказать, когда появятся новые функции и произойдет ли это вообще

Кросс-платформенность, надежность доставки сообщений и скорость - отличительные качества Telegram. Факт того, что приложение работает на всех платформах, сейчас не удивляет ни кого, но вопрос в том, насколько связаны разные приложения взаимодействуют друг с другом.

Исходя из перечисленного и рассмотренного выше можно сделать вывод, что популярность чат-ботов набирает обороты, а наиболее подходящим мессенджером для бота является Telegram.

2.3. Сравнение языков программирования, как наиболее подходящих для написания бота

2.3.1 Что такое язык программирования и какими они бывают

Язык программирования — это набор правил, которые определяют, как выглядит написанная компьютерная программа и что компьютер может делать под ее контролем. Программа — это код, написанный в соответствии с правилами данного языка программирования. Код, из которого состоит программа, называется «исходным кодом». Языки программирования — это формальные искусственные языки. Как и естественные языки, они имеют алфавит, словарный запас, грамматику и синтаксис, а также семантику⁵.

Алфавит — разрешенный к использованию набор символов, с помощью которого могут быть образованы слова и величины данного языка.

Синтаксис — система правил, определяющих допустимые конструкции языка программирования из букв алфавита.

Семантика — система правил однозначного толкования каждой языковой конструкции, позволяющих производить процесс обработки данных.

Все языки программирования делятся на два вида — языки низкого и высокого уровня⁶:

- Языки низкого уровня — это способ написания компьютерных инструкций на аппаратном языке, то есть в машинных кодах (в виде последовательности нулей и единиц). Языки низкого уровня жестко ориентированы на конкретный тип оборудования (система

⁵ <https://ibrain.kz/informatika/ponyatie-yazyk-programmirovaniya>

⁶ <http://csaa.ru/vidy-jazykov-programmirovaniya/>

управления процессором, каждый тип процессора имеет свой машинный код).

- Языки высокого уровня — это языки программирования, которые позволяют записывать программы в удобной для человека форме. Эти языки ориентированы не на систему инструкций того или иного процессора, а на систему операторов (инструкций), характерную для написания определенного класса алгоритмов.

Языки высокого уровня проще в использовании, поскольку их задача - обслуживать потребности программиста, а не определять возможности компьютера. Программы, написанные на этих языках, должны быть перекодированы - переведены на машинный язык, чтобы перед запуском программ компьютер мог их понять. Поэтому системы программирования на Java включают в себя либо интерпретатор языка, либо компилятор.

Языки низкого уровня, близкие к машинному языку, позволяют создавать программы, которые работают быстрее и позволяют более эффективно использовать ресурсы компьютера.

2.3.2 Рейтинг языков программирования

Согласно отчету RedMonk⁷ за январь 2020 года, Python поднялся на вторую позицию в рейтинге популярности языков программирования, опередив Java. Ранее Java занимала ту же позицию в рейтинге уже длительное время, с момента формирования этого списка в 2012 году.

Популярность JavaScript и Java и их места в рейтинге можно объяснить. JavaScript является основным языком для веб-разработки уже на протяжении многих лет, тогда как Java широко используется для разработки под Android и в корпоративных средах. (Рис. 5)

⁷ <https://redmonk.com/sograzy/2020/02/28/language-rankings-1-20/>

Однако в рейтинге не были учтены такие научные языки, как Mathematica, поскольку их применение и сообщества находятся за пределами ресурсов, на основе которых составлялся этот рейтинг.

Специалисты считают, что основной причиной роста популярности Python является его универсальность. Как и Java, Python применяется во многих областях и способен решать различные задачи. Такая гибкость в применении делает Python привлекательным для многих разработчиков по всему миру. Еще одним фактором, способствующим росту популярности Python, является его низкий порог вхождения. Таким образом, постоянно возрастает количество людей, присоединяющихся к «питонистскому» сообществу, аналогично с PHP или JS.

Весь рейтинг составлялся на базе данных GitHub Archive с перекрестной проверкой по базам Stackoverflow. Эти два ресурса использовались для сбора статистики, так как являются наиболее крупными сообществами разработчиков во всем мире.

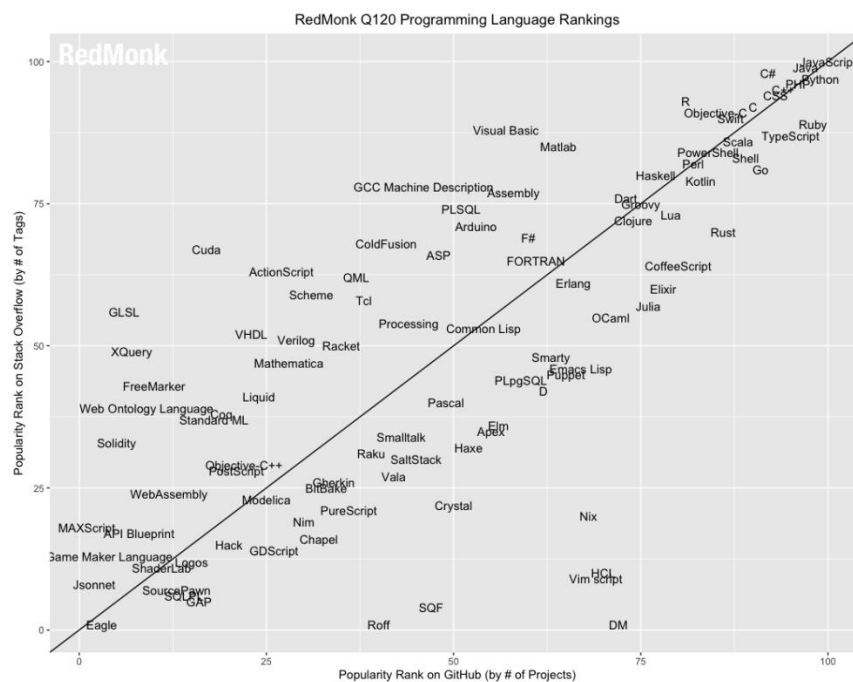


Рис.5 Рейтинг языков программирования от RedMonk

Вместе с тем, Американский журнал IEEE Spectrum опубликовал ежегодный рейтинг языков программирования.⁸ Для оценки относительной

⁸ <https://spectrum.ieee.org/top-programming-languages-2022>

популярности разных языков рейтинг учитывает несколько показателей из разных источников, таких как GitHub, Google, Stack Overflow, Twitter и IEEE Xplore. Всего было составлено три рейтинга с упором на различные ключевые факторы. Самыми популярными языками в этих рейтингах стали Python, SQL, C и Java. (Рис. 6)

Первый рейтинг Spectrum ориентирован на предпочтения и профессиональное мнение инженеров IEEE (Института инженеров электротехники и электроники). Здесь в пятёрку лучших входят Python, C, C++, C# и Java. При этом журнал отмечает, что совокупная популярность C и больших C-подобных языков намного превосходит Python. В пятёрку наименее популярных языков в этом рейтинге попали: Elm, Raku, WebAssembly, CoffeeScript, Eiffel.

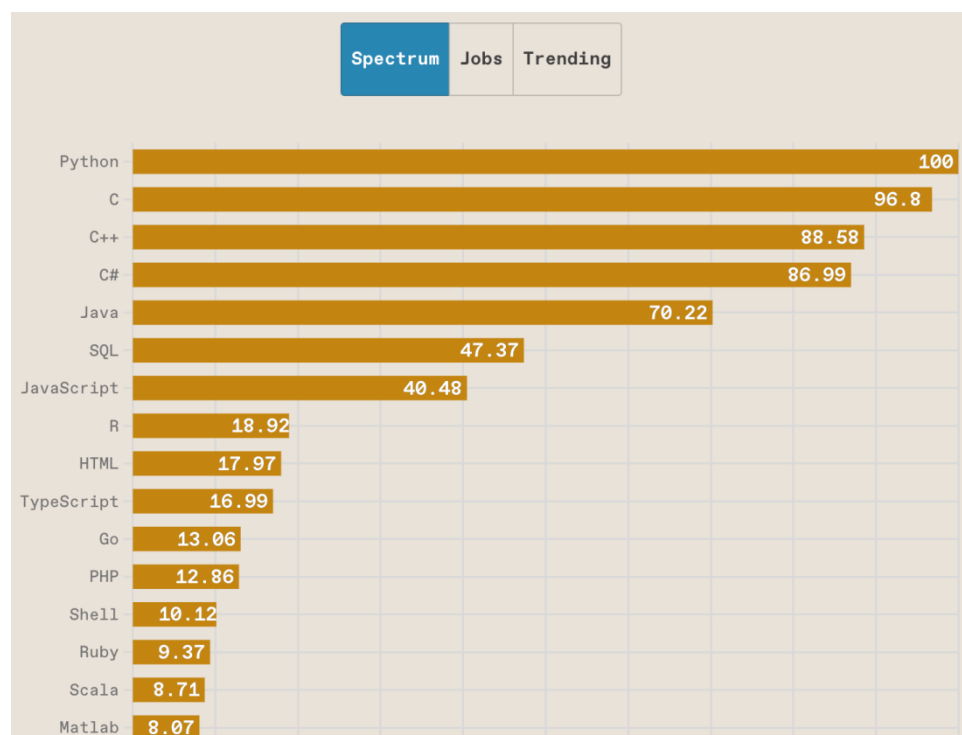


Рис.6 Рейтинг языков программирования от IEEE Spectrum

2.3.3 Анализ и сравнение популярных языков программирования

По данным, представленным выше самыми популярными языками на данный момент, являются Java, JavaScript, Python и C#, поэтому для анализа и сравнения выберем именно эти языки.

Язык, который широко используется в различных областях разработки программного обеспечения. Он был создан в конце 1980-х годов Гвидо ван Россумом и быстро стал одним из самых популярных языков программирования в мире.

Преимущества языка:

Простота использования

Python имеет интуитивный и простой синтаксис, что делает его легким для изучения и использования.

Гибкость в применении

Данный язык обладает стандартными и дополнительными библиотеками, что делает его гибким для решения различных задач.

Адаптированность для науки и анализа данных

Язык имеет множество библиотек (например, NumPy, SciPy и Pandas), что делают его оптимальным для применения в науке, инженерии и анализе данных.

Применимость для машинного обучения и искусственного интеллекта

Python также имеет библиотеки TensorFlow и PyTorch, которые делают его отличным выбором для работы в области машинного обучения и искусственного интеллекта.

Какие продукты были созданы: Sims 4, YouTube.

2.4.5 Java

Объектно-ориентированный язык, который позволяет создавать программы с использованием классов и объектов, что облегчает процесс разработки и дает возможность более эффективного управления кодом. Был создан в 1995 году компанией Sun Microsystems.

Преимущества языка:

Высокая производительность

Java использует компиляцию Just-in-time (JIT), что позволяет усовершенствовать производительность программ.

Простота в изучении

Язык становится доступным для обучения начинающим программистам благодаря его простому и читаемому синтаксису.

Масштабирование в программах

Он прекрасно может быть использован для разработки как небольших приложений, так и больших сложных систем.

Mobile-разработка

Java является одним из наиболее популярных языков программирования для разработки мобильных приложений в версии Android.

Какие продукты были созданы: Uber, Twitter, Netflix.

2.4.6 C#

Объектно-ориентированный язык программирования, используется в основном для создания приложений на платформе .NET. Разработан в 1998–2001 годах группой инженеров Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота.

Преимущества языка:

Автоматизация

C# поддерживает автоматическую управляемую память, что облегчает разработку, так как не нужно выделять и освобождать память вручную

Производительность

Он является компилируемым языком. Это значит, что программа транслируется в машинный код, а это повышает производительность.

Средства разработки

Данный язык поддерживает современные инструменты, такие как Visual Studio, которые упрощают процесс разработки.

Распространение

C# используется для разработки многих программных приложений и игр, что делает его одним из популярных языков программирования.

Какие продукты были созданы: движок Unity 3D.

2.4.7 JavaScript

Динамический язык программирования, используемый для разработки веб-сайтов и приложений. JavaScript был изначально создан для клиентской стороны веб-страниц, чтобы придать ей дополнительную функциональность и интерактивность. Он используется для создания динамических пользовательских интерфейсов, анимаций, игр, мобильных приложений и многого другого. JavaScript был создан в 1995 Бренданом Айком, инженером компании Netscape.

Преимущества языка:

Поддержка открытых стандартов

JavaScript — стандартный язык программирования, который используется большинством браузеров. В связи с этим он поддерживает многие стандарты и расширения, такие как DOM, AJAX и JSON.

Количество библиотек и фреймворков

Данный язык имеет множество библиотек и фреймворков, что позволяет разработчикам быстрее и проще создавать сложные веб-приложения.

Интерактивность

Он дает возможность создавать интерактивные веб-сайты с привлекательным пользовательским интерфейсом, возможностью анимации и других эффектов.

Масштабируемость

JavaScript позволяет создавать масштабируемые веб-приложения, что обеспечивает высокую производительность вне зависимости от количества пользователей и трафика.

Какие продукты были созданы: Javascript — язык программирования, на котором можно написать фронтенд сайта, поэтому он используется на подавляющем большинстве веб-сайтов.

3. Глава 2. Практическая реализация чат-бота

3.1. Средства разработки чат-бота

3.1.1 Необходимые модули для разработки сервиса

Из первой главы мы выяснили, что язык программирования Python хорошо подойдет для написания чат-бота для мессенджера Telegram. Рассмотрим библиотеки данного языка необходимые для написания чат-бота.

Модуль `datetime` предоставляет классы для обработки времени и даты разными способами. Поддерживается и стандартный способ представления времени, однако больший упор сделан на простоту манипулирования датой, временем и их частями⁹.

`Requests` — это модуль Python, который можно использовать для отправки всех видов HTTP-запросов. Это простая в использовании библиотека с множеством функций, от передачи параметров в URL до отправки пользовательских заголовков и проверки SSL¹⁰.

`Ergast Developer API` — это экспериментальный веб-сервис, который предоставляет исторические записи данных об автогонках для некоммерческих целей.

⁹ <https://pythonworld.ru/moduli/modul-datetime.html>

¹⁰ <https://code.tutsplus.com/ru/tutorials/using-the-requests-module-in-python--cms-28204>

API Wikipedia - поддерживается API MediaWiki и обеспечивает доступ к Wikipedia и другим данным MediaWiki без взаимодействия с пользовательским интерфейсом.

pyTelegramBotAPI (telebot) - это библиотека Python, которая предоставляет интерфейс для взаимодействия с Telegram Bot API. Она позволяет отправлять сообщения, создавать встроенные клавиатуры, обрабатывать обратные вызовы и выполнять множество других действий с легкостью. Библиотека проста в использовании и может быть установлена с помощью pip.

Модуль sys обеспечивает доступ к некоторым переменным и функциям, взаимодействующим с интерпретатором Python¹¹.

3.1.2 Среда разработки

В качестве среды разработки будем использовать Visual Studio Code (VS Code). VS Code — это редактор исходного кода. Его разработал Microsoft для всех популярных операционных систем: Windows, Linux и macOS. Визуальный редактор кода позволяет:

- Работать с IntelliSense — автоматическим дописыванием функций при вводе первых букв;
 - Выполнять отладку — искать и устранять ошибки в написанном коде;
 - Удобно писать код — автоматически заполнять нужную информацию, подсвечивать элементы синтаксиса в зависимости от выбранного языка, расставлять нужные отступы;
 - Контролировать версии кода, в том числе с помощью системы управления версиями Git;
 - Рефакторить код для улучшения его работы и читабельности.
- Все эти функции значительно облегчают труд разработчика: позволяют избежать опечаток, реже ошибаться и работать быстрее.

¹¹ <https://pythonworld.ru/moduli/modul-sys.html>

В 2021 году Visual Studio Code стал самым популярным редактором исходного кода по версии пользователей сайта Stack Overflow. За него проголосовали 71% респондентов¹².

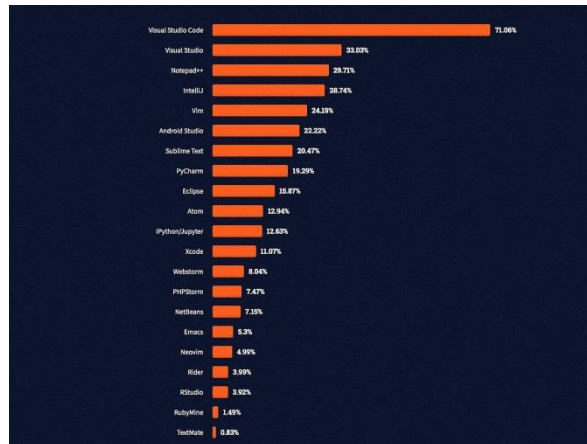


Рис.7 Рейтинг IDE по данным отчёта Stack Overflow Developer Survey 2021

3.1.3 Система контроля версий

Система контроля версий (англ. Version Control System) - программное обеспечение хранящееся все версии возможного файла, и дающее возможность получить к ним доступ. Существуют разные системы: git, mercurial, subversion(svn), Team Foundation server (TFS), однако наибольшую популярность завоевала система Git из-за простоты использования и внедрения в другие системы.

Git — пожалуй, самая известная и популярная подобная система в области программирования. Она была разработана Линусом Торвальдсом для управления разработкой ядра Linux. Сегодня это один из самых значимых репозиторий, который используют Android, Swift, Drupal, jQuery и многие другие.

Распределенная система контроля версий подразумевает под собой сразу три сервиса:

1. Хранение, обработка и передача данных. Сама возможность работы с «КОНТЕНТОМ».

¹² <https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies>

2. Контроль за всеми вариациями проекта. То, о чем мы говорили выше: в любой момент времени вы можете восстановить предыдущие версии, буквально вернуться в прошлое до «поломки» кода.
3. Параллельная разработка. Удаленное файловое хранилище делает доступной одновременную редакцию одних и тех же объектов разными людьми и при этом гарантирует отсутствие конфликтов в коде. Это достигается благодаря тому, что каждый участник проекта хранит свои правки в локальной репозитории на своем устройстве, и они не вступят в силу, пока он не выгрузит их на сервер и не сольет с рабочей версией.

Прежде чем использовать Git, необходимо установить его на своём компьютере. Официальная сборка для Window доступна для скачивания на официальном сайте Git. Существует несколько способов установки Git на Mac. Самый простой — установить Xcode Command Line Tools. В версии Mavericks (10.9) и выше вы можете добиться этого просто первый раз выполнив 'git' в терминале.

```
$ git --version
```

Если Git не установлен, вам будет предложено его установить. Чтобы посмотреть все установленные настройки и узнать где именно они заданы, используйте команду:

```
$ git config --list --show-origin
```

VSCode автоматически определит местоположение Git, и затем появится возможность синхронизации (рис. 8).

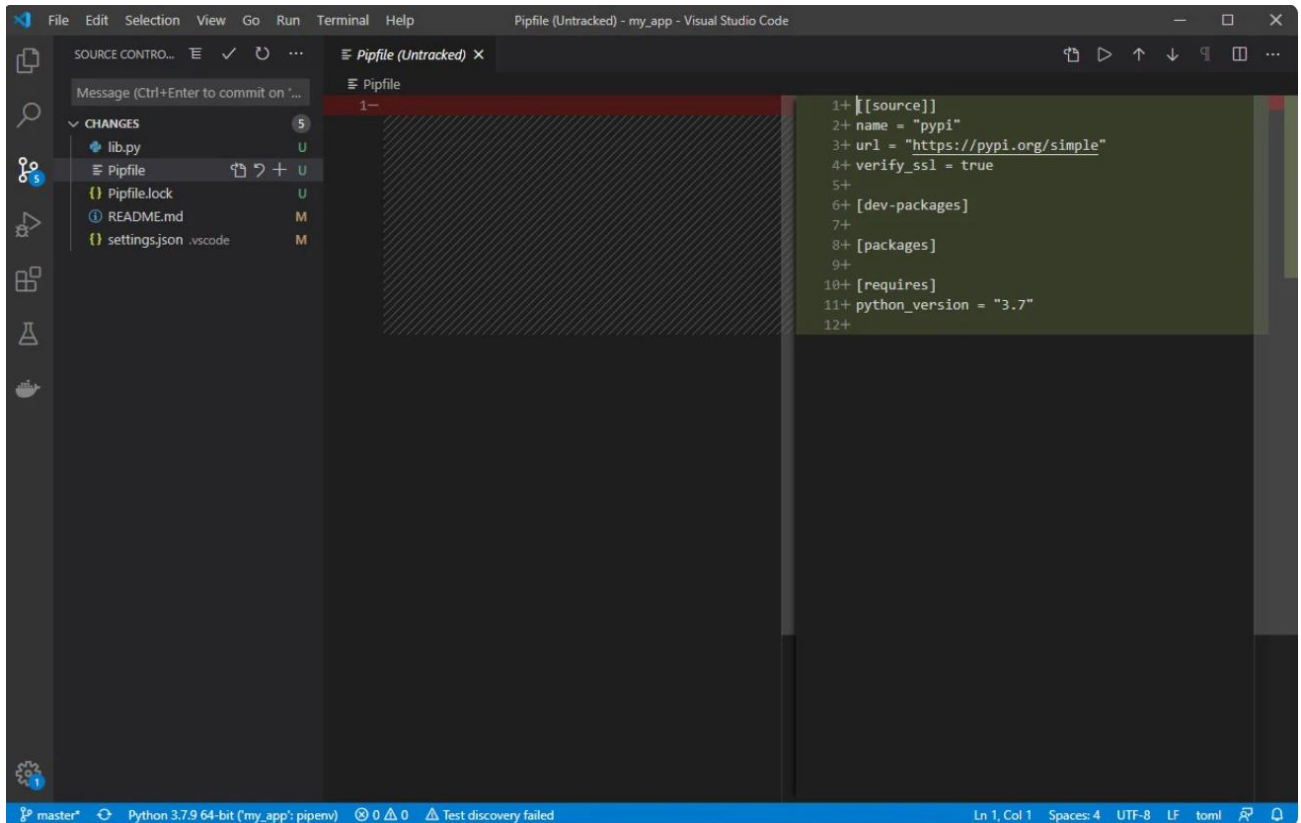


Рис.7 Для работы с git, зайдите в меню слева "Source Control"

3.1.4 Telegram Bot API

Это обычный HTTP-interface для взаимодействия с чат-ботами в мессенджере Telegram. Каждый созданный бот должен уметь автоматически отправлять сообщения и принимать входящие. Для этого нам и нужен сервер, принимающий http запросы и выдающий ответы¹³.

Есть всего 2 способа получать обновления с сервера:

- long pulling – телеграмм сам будет раз в какое-то время, например в 1 секунду, спрашивать сервер о новых уведомлениях;
- webhook – или сервера сами будут присылать оповещение, сразу как появится новое уведомление¹⁴.

¹³ Telegram Bot API <https://core.telegram.org/bots/api>

¹⁴ Бот Telegram и HTTP сервис в 1C <https://infostart.ru/1c/articles/1368626/>

Все оповещения будут сохранены на сервере без обработки, но не долго, максимум до 24 часов. В ответ будет отправлен объект «Апдейт», заранее сериализованный в JSON. Все запросы к Telegram Bot API должны осуществляться через HTTPS в следующем виде:
https://api.telegram.org/bot<token>/НАЗВАНИЕ_МЕТОДА.

Принцип работы взаимодействия чат-бота и пользователя изображен на рисунке.

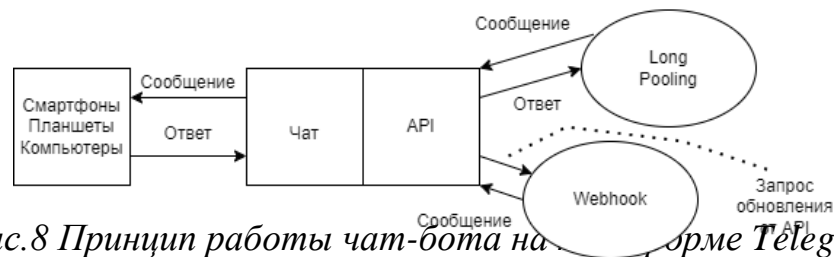


Рис.8 Принцип работы чат-бота на платформе Telegram

Чтобы получить для нашего бота ключ доступа, он же токен, необходимо написать botfather боту. Он создан для того, чтобы создавать ботов. Ниже приведены некоторые примеры методов:

- getUpdates – этот метод используется для получения обновлений по технологии long polling;
- sendMessage – метод осуществляет обычную отправку сообщения пользователю.

3.2 Разработка телеграмм-бота

3.2.1 Регистрация чат-бота

Для начала нам потребуется зарегистрировать нашего бота в телеграм, то есть нам нужно получить токен бота, он же ID уникальный идентификатор. В этом может помочь специальный бот @BotFather.

Как это сделать:

1. Пишем в бота команду /start.
2. Далее пишем команду /newbot (рис. 9)

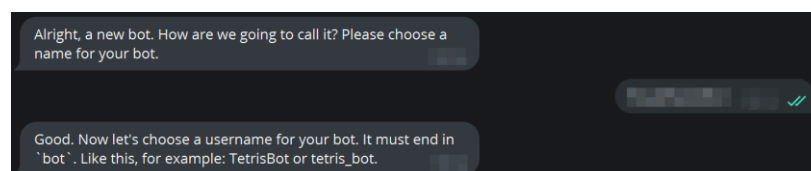
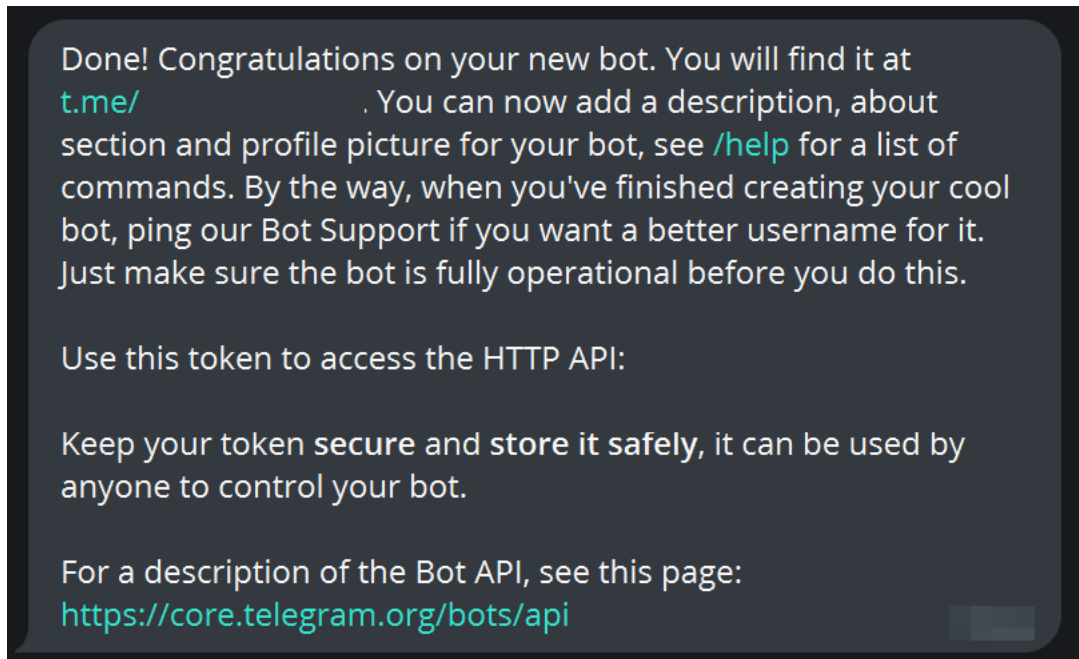


Рис.9 Регистрация бота

3. Придумываем имя для нашего бота. Отмечу что имя бота должно оканчиваться на «bot», иначе бота создать не удастся.

4. Помощник пришлет нам поздравления и выдаст токен, с помощью которого мы сможем взаимодействовать с «внутренностями» нашего бота.

*Рис.9 Получение Токена*

3.2.2. Описание структуры бота

Функционал бота подразумевает:

- Приветственное сообщение с предложением перейти в «меню» бота;
- Меню бота, состоящее из кнопок и сообщения;
- Модули (под словом модули далее понимаются функции бота, которые пользователь может использовать):

- Пилоты. Содержит список пилотов «Формулы 1» текущего сезона, упорядоченный по количеству очков в чемпионате на текущий момент (очки отображаются в списке). Данный модуль также предусматривает возможность получить более полную информацию о каждом конкретном пилоте, в том числе в Wikipedia;

- Конструкторы. Содержит список конструкторов «Формулы 1» текущего сезона, упорядоченный по количеству очков в чемпионате на текущий

момент. Данный модуль также предусматривает возможность получить более полную информацию о каждой конкретной команде, в том числе в Wikipedia. Этот модуль реализован визуально и функционально иначе, чем модуль «Пилоты»;

- Последняя гонка. Модуль содержит подробную информацию о последнем гран-при (название, дата, место, время). Так же модуль содержит список пилотов, сформированный по результатам финиша гонки. Имеется возможность ознакомиться с новостями по прошедшей гонке и почитать подробную информацию о трассе на Wikipedia;

- Последняя квалификация. Модуль содержит список пилотов сформированный по результатам квалификации к последней гонке, содержащий фамилию и имя пилота, а так же время, которое показал пилот, в каждом сегменте квалификации;

- Предстоящая гонка. В модуле присутствует подробная информация о предстоящей гонке, такая как: название, место, дата, время (локальное для пользователя), погода на трассе в текущий момент, а так же предоставляется ссылка на просмотр трансляции гонки, при ее наличии у поставщика;

- Трассы. Модуль содержит список актуальных для текущего сезона трасс формулы 1, упорядоченный по алфавиту. Т.е. в данном модуле можно посмотреть не только подробную информацию о трассе, но и узнать о дате гран-при на ней;

- Новости. Функция ищет новости в интернете и выдает пользователю их заголовок, краткое содержание и ссылку на полную новость. Так же дает возможность перейти в меню;

- Чемпионы прошлого. Модуль показывает чемпиона мира и победителя кубка конструкторов соответствующего года;

- Гифка. Функция находит гифки о формуле 1 (используя API Tenor) и показывает пользователю;

- Вспомогательные модули (функции) - функции, для определения текущей даты и времени, погоды, новостей и т.п. Данные функции необходимы

для корректной работы основных функций бота, но не участвуют в непосредственном взаимодействии с пользователем. Данные функции используют сторонние API, такие как Wikipedia, Tenor, News, OpenWeather.

3.2.3 Некоторые примеры работы бота

Приветственный экран и основное меню бота представляют из себя текстовое поле и ряд кнопок (InlineKeyboard) с соответствующими обработчиками.

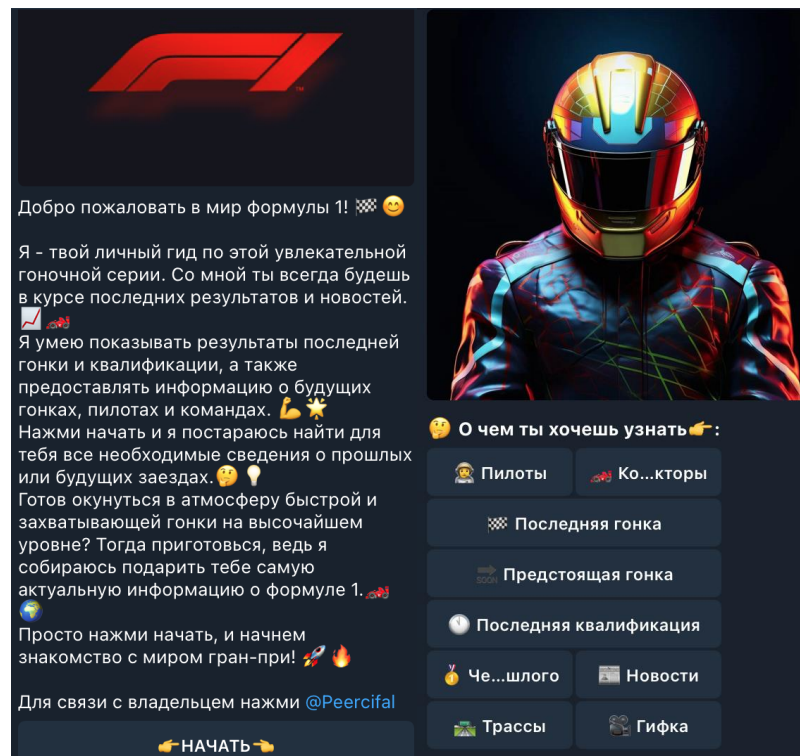


Рис.10 Вступительный экран бота

На данном этапе бот выводит функции Start и Menu, запускает обработку нажатий на клавиатуру и ждет ответа от пользователя. При выборе любой из функций бота на клавиатуре происходит запуск функции кода, которая обрабатывает нажатие и запускает соответствующую функцию. Например функция новости не только показывает новости о формуле 1 используя NewsApi, но и формирует файл, в котором содержится информация о уже показанных новостях с целью избежания их повторения.

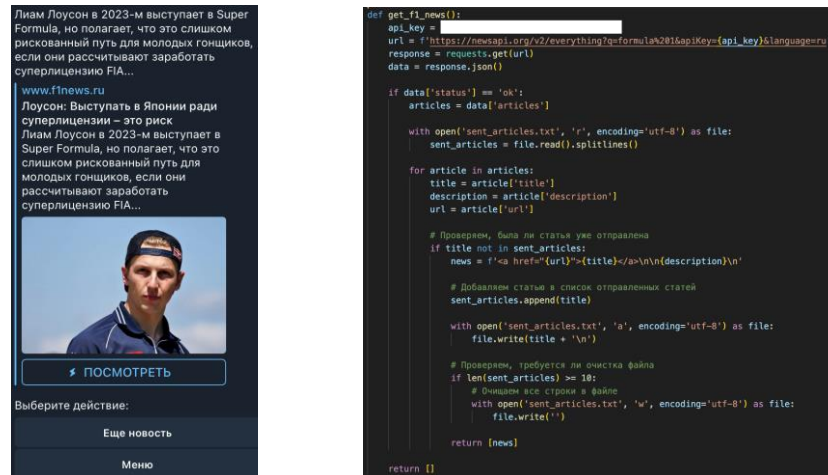


Рис.11 Пример функции «Новости»

Функция «Следующая гонка» используя API Ergast, API OpenWeather показывает информацию о следующей гонке. Так же функция check_live_stream трансляцию гонки в случае ее наличия на канале. Возможность использования данной функции оговорена с владельцем канала, с него получено согласие. Функция get_flag_emoji_country всегда выводит флаг соответствующей гонки.

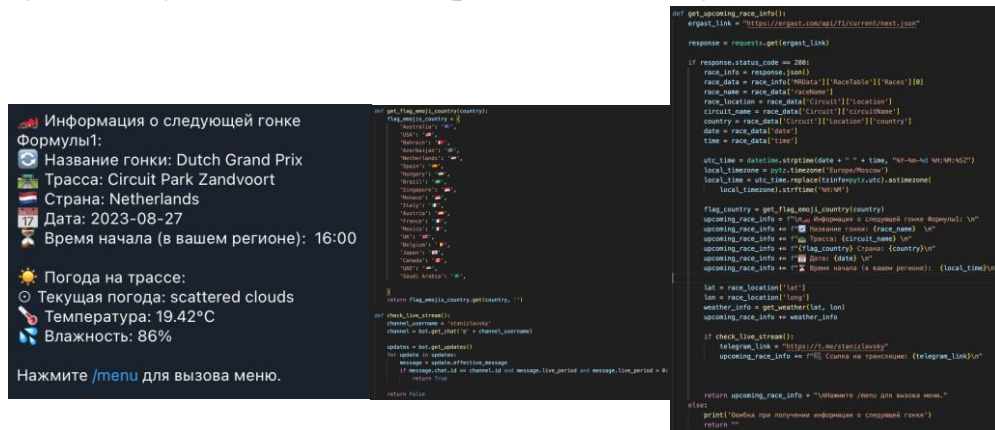


Рис.12 Пример функции «Следующая гонка»

Функция «последняя гонка»

показывает информацию о финишном зачете последнего гран-при, количество очков и время. Рандомно предлагает новости о последней гонке и подробную информацию, которая имеется в Wikipedia. Функция использует API Ergast, Wikipedia API.

```
def get_last_race_info():
    race_data = None
    try:
        response = requests.get(
            'http://ergast.com/api/f1/current/last/results.json')

        if response.status_code == 200:
            race_data = response.json()['MRData']['Races'][0]
            last_race_name = race_data['raceName']
            race_info = f'Последняя гонка:\n'
            race_info += f'Название: {race_data["raceName"]}\n'
            race_info += f'Дата: {race_data["date"]}\n'
            race_info += f'Кругов: {race_data.get("laps", "Информация недоступна")}\n'
            race_info += f'Дистанция: {race_data.get("distance", "Информация недоступна")}\n'

            # Добавление ссылки на Википедию
            wiki_link = race_data.get('url', '')
            if wiki_link:
                race_info += f'Ссылка на Википедию: {wiki_link}\n'

            # Добавление новостей о последней гонке Формулы 1
            race_info += get_last_race_news(last_race_name)

            # Добавление результатов гонки по пилотам
            race_info += get_race_results()

            return race_info + "\nНажмите /menu для вызова меню."
        else:
            return "Информация недоступна"
    except requests.exceptions.RequestException as e:
        print("Ошибка при запросе данных:", str(e))

    return race_data

def get_race_results():
    race_results_data = None
    try:
        response = requests.get(
            'http://ergast.com/api/f1/current/last/results.json')

        if response.status_code == 200:
            race_results_data = response.json()
            race_results_info = f'Результаты гонки (позиция, имя, время, очки):\n'
            winner_info = race_results_data['Results'][0]
            winner_name = winner_info['Driver']['givenName'] + \
                ' ' + winner_info['Driver']['familyName']
            winner_time = winner_info.get('Time', {}).get('time', 'N/A')
            nationality = winner_info['Driver']['nationality']
            flag = get_flag_emoji(nationality)
            points = winner_info['points']
            race_results_info += f'{flag} {winner_name} - {winner_time} ((points))\n'
            # race_results_info += f'Очки: (points)\n'

            for i, result in enumerate(race_results_data['Results'][1:], start=2):
                driver_name = result['Driver']['givenName'] + \
                    ' ' + result['Driver']['familyName']
                time_difference = result.get('Time', {}).get('time', 'N/A')
                nationality = result['Driver']['nationality']
                flag = get_flag_emoji(nationality)
                points = result['points']
                race_results_info += f'{i}. {flag} {driver_name} - {time_difference} ((points))\n'

            return race_results_info
        else:
            return "Информация недоступна"
    except requests.exceptions.RequestException as e:
        print("Ошибка при запросе данных:", str(e))

    return race_results_data

def get_last_race_news(last_race_name, num_articles=3):
    api_key = '33ce6ff8cd2240b68c329582a81ba240'
    last_race_name = last_race_name.replace(' ', '%20')
    newsapi_link = f'https://newsapi.org/v2/everything?q={last_race_name}&apiKey={api_key}&language=ru'

    response = requests.get(newsapi_link)

    if response.status_code == 200:
        news = response.json()
        articles = news['articles']
        news_info = f'Новости о последней гонке Формулы 1 📰\n'

        for i, article in enumerate(articles[:num_articles]):
            title = article['title']
            url = article['url']
            news_info += f'Заголовок: {title}\n'
            news_info += f'Ссылка: {url}\n'

        return news_info
    else:
        print("Ошибка при получении новостей")
    return ""
```

Последняя гонка:
 Название: Dutch Grand Prix
 Дата: 2023-08-27
 Кругов: Информация недоступна
 Дистанция: Информация недоступна
 Ссылка на Википедию:
https://en.wikipedia.org/wiki/2023_Dutch_Grand_Prix

Новости о последней гонке Формулы 1 📰
 Заголовок: Гран При Нидерландов: Трасса и статистика
 Ссылка:
<https://www.f1news.ru/Championship/2023/netherlands/index.shtml>

Заголовок: Главные герои и неудачники Гран При Нидерландов
 Ссылка: <https://autosport.com.ru/features/86056-glavnye-geroi-i-neudachniki-gran-pri-niderlandov#article>

Результаты гонки (позиция, имя, время, очки):
 1. 🇳🇱 Max Verstappen - 2:24:04.411 (25)
 2. 🇪🇸 Fernando Alonso - +3.744 (19)
 3. 🇫🇷 Pierre Gasly - +7.058 (15)
 4. 🇪🇸 Sergio Pérez - +10.068 (12)
 5. 🇪🇸 Carlos Sainz - +12.541 (10)
 6. 🇬🇧 Lewis Hamilton - +13.209 (8)
 7. 🇬🇧 Lando Norris - +13.232 (6)
 8. 🇩🇪 Alexander Albon - +15.155 (4)
 9. 🇮🇹 Oscar Piastri - +16.580 (2)
 10. 🇲🇽 Esteban Ocon - +18.346 (1)
 11. 🇨🇦 Lance Stroll - +20.087 (0)
 12. 🇩🇪 Nico Hülkenberg - +20.840 (0)
 13. 🇮🇪 Liam Lawson - +26.147 (0)
 14. 🇫🇮 Valtteri Bottas - +27.388 (0)
 15. 🇯🇵 Yuki Tsunoda - +29.893 (0)
 16. 🇳🇱 Kevin Magnussen - +31.410 (0)
 17. 🇬🇧 George Russell - +55.754 (0)
 18. 🇨🇳 Guanyu Zhou - N/A (0)
 19. 🇫🇷 Charles Leclerc - N/A (0)
 20. 🇺🇸 Logan Sargeant - N/A (0)

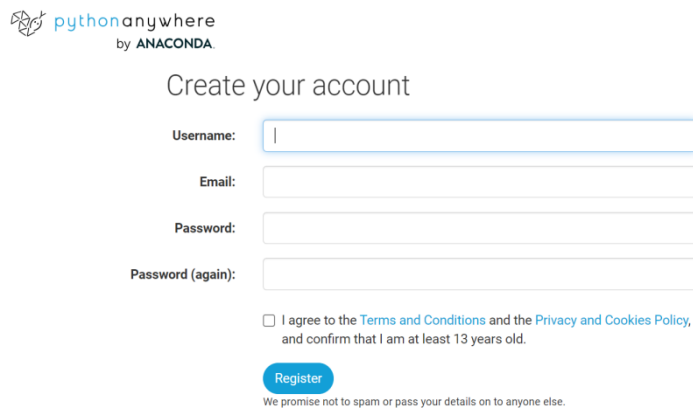
Нажмите /menu для вызова меню.

Wikipedia
 2023 Dutch Grand Prix
 The 2023 Dutch Grand Prix (officially known as the Formula 1 Heineken Dutch Grand Prix 2023) was a Formula One motor race that was...

Рис.13 Пример функции «Последняя гонка»

3.2.4 Публикация (хостинг) бота.

С целью хостинга бота будем использовать бесплатную версию pythonyanywhere. Заходим на сайт pythonyanywhere.com и регистрируемся. Можно использовать любую почту, включая временную. Не обязательно подтверждать ее, но полезные уведомления будут приходить на указанный адрес.



pythonanywhere
by ANACONDA

Create your account

Username:

Email:

Password:

Password (again):

☐ I agree to the [Terms and Conditions](#) and the [Privacy and Cookies Policy](#), and confirm that I am at least 13 years old.

[Register](#)

We promise not to spam or pass your details on to anyone else.

Рис.14 Регистрация на pythonanywhere

После успешной регистрации, открываем боковое меню и переходим в раздел "Файлы". Удаляем все файлы, которые находятся там.

Далее, начинаем загружать файлы бота по одному. Выбираем нужный файл, загружаем его и повторяем эту операцию с остальными файлами.

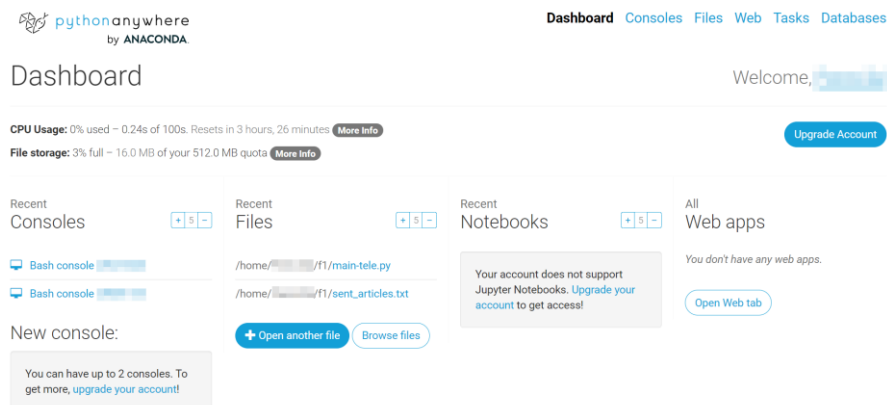


Рис.15 Интерфейс pythonanywhere

После загрузки всех файлов переходим к запуску консоли. Как только в консоли появится символ "\$", можем начинать установку необходимых библиотек для бота, например `pip3 install --user telebot` и т.д. Python уже установлен изначально.

Наконец, запускаем файл, необходимый для работы бота.

Заключение

Разработанный бот – это информационно-развлекательный бот, который посвящен миру Формулы 1. Бот предоставляет множество полезной и интересной информации об этом захватывающем спорте, способствуя распространению знаний и популяризации Формулы 1 среди любителей и профессионалов.

В современном быстром темпе жизни, электронные устройства и мессенджеры стали неотъемлемой частью нашей повседневности. Наш бот предлагает интерактивное и удобное инструментарий для общения с пользователем, позволяющий удовлетворить его потребности в получении информации о мире Формулы 1 и обеспечить интерактивное развлечение в этой тематике.

Основная цель разработки этого бота – дать возможность пользователям получать свежую информацию о гонках, командах, пилотах и аналитике в режиме реального времени. Это особенно важно для людей, которые не всегда могут следить за трансляциями гонок или иными событиями в Формуле 1. Благодаря нашему боту, они смогут быть в курсе последних новостей и событий, получать уведомления о предстоящих гонках, результаты квалификаций и гонок, а также статистические данные о каждом этапе чемпионата.

Бот Формулы 1 использует различные API, чтобы предоставить пользователям полезную информацию из надежных источников.

Один из ключевых API, с которыми работает бот, - это Ergast API. Оно предоставляет доступ к обширной базе данных о Формуле 1, включая исторические данные о гонках, результаты квалификаций и гонок, информацию о командах и пилотах, а также временные промежутки и маршруты каждого этапа чемпионата. Благодаря Ergast API бот может обновлять информацию мгновенно и давать актуальные результаты в режиме реального времени.

Кроме Ergast API, бот также использует API от Wikipedia. Это позволяет пользователям получить более обширную информацию о командах, пилотах и исторических фактах Формулы 1. Благодаря интеграции с этим API, бот может

предоставить пользователю дополнительные детали и интересные факты, которые могут быть полезными для расширения знаний пользователя о Формуле 1.

Бот также обращается к Openweather API, чтобы предоставить пользователям актуальную информацию о погоде в местах проведения гонок. Благодаря этому, пользователи смогут узнать, какие погодные условия могут повлиять на ход гонок и какие этапы могут быть более предвзятыми к определенным командам или пилотам.

Использование различных API позволяет нашему информационно-развлекательному боту Формулы 1 создавать уникальный и многофункциональный опыт для своих пользователей.

Разработанный бот это не просто справочник или новостной источник. Это целый электронный помощник, который помогает своим пользователям расширить знания о Формуле 1, быть в курсе всех событий и участвовать в интересных развлекательных активностях, связанных с этим захватывающим мировым состязанием.

Информационно-развлекательный бот Формулы 1 станет незаменимым и любимым помощником и развлечением для всех поклонников этого шоу.

Список литературы

1. <https://www.formula1.com/en/latest/article.formula-1-announces-tv-race-attendance-and-digital-audience-figures-for-2021.1YDpVJIOHGnuok907sWcKW.html>
2. https://www.championat.com/tags/20056formula1/?utm_source=Rambler&utm_medium=sport&utm_campaign=transition
3. <https://ict.moscow/research/cifrovye-tehnologii-v-rossiyskih-kompaniyah/>
4. Резкий рост популярности российских мессенджеров: анализ интернет-трафика пользователей Yota 2022 год
https://www.cnews.ru/news/line/2022-0317_rezkij_rost_populyarnosti
5. <https://ibrain.kz/informatika/ponyatie-yazyk-programmirovaniya>
6. <http://csaa.ru/vidy-jazykov-programmirovaniya/>
7. <https://redmonk.com/sograde/2020/02/28/language-rankings-1-20/>
8. <https://spectrum.ieee.org/top-programming-languages-2022>
9. <https://pythonworld.ru/moduli/modul-datetime.html>
10. <https://code.tutsplus.com/ru/tutorials/using-the-requests-module-in-python--cms-28204>
11. <https://pythonworld.ru/moduli/modul-sys.html>
12. <https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies>
13. Telegram Bot API <https://core.telegram.org/bots/api>
14. Бот Telegram и HTTP сервис в 1С
<https://infostart.ru/1c/articles/1368626/>