**Общество с ограниченной ответственностью**

**«ГикБреинс»**

Программист Python Цифровые профессии

Дипломная работа по дисциплине:

«Программист Python Цифровые профессии»

на тему:

**«****Telegram бот для мониторинга изменения цен товаров, выбранных пользователем на примере торговой площадки Wildberries»**

Выполнил:

Быков Иван Михайлович

Москва, 2024г.

**Содержание**

[**Введение** 3](#_Toc158566203)

[**1.** **ДЕТАЛЬНЫЙ АНАЛИЗ И ОБЗОР СРЕДСТВ ДЛЯ РАЗРАБОТКИ СЕРВИСА** 7](#_Toc158566204)

[**1.1** **Общие понятия. Что такое чат-бот. Его основные функции. Классификация ботов.** 7](#_Toc158566205)

[**1.2** **Плюсы, варианты применения и статистика использования ботов для бизнеса.** 11](#_Toc158566206)

[**1.3** **Выбор языка программирования и приведение статистики популярности языков на основе коммерческих рабочих проектов.** 16](#_Toc158566207)

[**1.4** **Выбор платформы для разработки чат-бота.** 19](#_Toc158566208)

[**1.5** **Выводы по главе 1.** 21](#_Toc158566209)

[**2.** **ОПИСАНИЕ МЕТОДОВ РАЗРАБОТКИ ПРИЛОЖЕНИЯ** 22](#_Toc158566210)

[**2.1.** **Язык программирования Python как идеальное средство для разработки** 22](#_Toc158566211)

[**2.1.1.** **Разбор необходимых модулей для разработки сервиса** 22](#_Toc158566212)

[**2.1.2.** **Что называется функцией в Python?** 25](#_Toc158566213)

[**2.1.3.** **Что называется условными операторами и циклами?** 25](#_Toc158566214)

[**2.1.4.** **Что такое исключения?** 27](#_Toc158566215)

[**2.1.5.** **Основной метод, используемый для проекта.** 28](#_Toc158566216)

[**2.2.** **Описание компании Wildberries** 29](#_Toc158566217)

[**2.3.** **Выводы по главе 2.** 31](#_Toc158566218)

[**3.** **Описание разработки и Практическая часть реализации сервиса** 32](#_Toc158566219)

[**3.1.** **Разработка IT проекта** 32](#_Toc158566220)

[**3.1.1.** **Устав проекта** 32](#_Toc158566221)

[**3.1.2.** **Структура проекта** 32](#_Toc158566222)

[**3.1.3.** **Риски проекта** 34](#_Toc158566223)

[**3.2.** **Практическая часть реализации бота** 36](#_Toc158566224)

[**3.2.1.** **Генерация токена для интеграции чат-бота в мессенджер Telegram.** 36](#_Toc158566225)

[**3.2.2.** **Парсинг сайта Wildberries.ru** 38](#_Toc158566226)

[**3.2.3.** **Создание базы данных и необходимых для чат-бота таблиц** 39](#_Toc158566227)

[**3.2.4.** **Сервис для автоматического уведомления о изменении цен.** 42](#_Toc158566228)

[**3.2.5.** **Описание работы бота** 47](#_Toc158566229)

[**3.3.** **Вывод по главе 3** 51](#_Toc158566230)

[**ЗАКЛЮЧЕНИЕ** 52](#_Toc158566231)

[**Список использованных Источников** 53](#_Toc158566232)

**Введение**

Динамика современной жизни, требует от людей быстрых, своевременных и взвешенных решений. Торговля, как и покупки, является неотъемлемой и вездесущей частью жизни каждого современного человека. Для закрытия этой базовой, в современном мире потребности, существует множество решений: быстрых, мобильных, качественно улучающих или ухудшающих ежедневное существование. Каждый из нас задумывался на тему того, как не утонуть в этом многообразии предложений рынка, а главное, как фильтровать, всю ту информацию, предоставляемую этими продуктами и получать лишь ту информацию, которая нам действительно необходима и желательно получать ее своевременно и в кратчайший срок. Да, решения безусловно есть, IT это самая динамично развивающаяся на сейчас сфера, каждый день, час, минуту люди по всему миру придумывают новые решения и способствуют улучшению мира, экологии, безопасности, изобретают все новые решения для улучшения различных процессов основываясь на собственном опыте и опыте различных людей, изменяя подход к решению ежедневных, для многих уже ставших рутинными, задач. Но вот вопрос, хороши ли эти решения, могут ли ими воспользоваться все, кому они необходимы, можно ли закрыть ими все вышеизложенные вопросы, думаю ответ очевиден – нет. Каждый продукт по-своему уникален и определенно найдет свою нишу и своего пользователя. При выборе инструмента для подготовки этого проекта, я рассматривал несколько платформ, механизмов и даже языков для решения задачи:

- Flutter (Dart - Google lang. fraimwork) и разработка на базе него мобильного приложения (т.к. обучение проходило на курсе Python – очевидно брать этот вариант было бы нелепо и глупо, но можно было бы взять Kivy fraimwork от Python – и после «танцев с бубном» таки скомпилировать мобильное приложение, но это «боль и страдания», а т.к. Python завещал помнить о том, что: «Простое лучше сложного» пропускаем этот вариант и идем дальше)

- Flask (Python micro framework) и разработка на базе него web приложения.

- PyQT6 и разработка на его базе Desktop приложения

- Asyncio (Python asinc framework) и разработка на базе него Telegram бота.

Разумеется, я выбрал последнее, ведь асинхронная составляющая фреймворка позволит мне без проблем поддерживать проект с большим кол-вом пользователей, это быстрое решение полностью покрывающее все нужды проекта и практически полностью закрывающее потребности по фронт и частично бэк разработке. Так же не мало важным фактором стала гибкость Telegram как мессенджера, его распространенность и кроссплатформенность. От пользователя потребуется наличие мессенджера и пара секунд на добавление бота – вуаля и он уже может начать пользоваться. А так же не кто не мешает, в случае успеха проекта поделиться бэкэнд частью проекта через API на базе Flask, подключить мобильное, web или desktop приложение с более широким и глубоким функционалом.

Что такое Telegram?

Как следует из википедии: «Telegram - кроссплатформенная система обмена сообщениями мгновенного действия (далее по тексту мессенджер) с функциями обмена сообщениями, содержащими текст, голос и(-или) видео сообщениями, а еще есть возможность обмена стикерами, фото и прочими файлами различных форматов. Также дает возможность совершать аудио- и видеозвонки, устраивать прямые эфиры в каналах и группах, организовывать конференции, многопользовательские группы и каналы.»

Что такое Telegram бот?

Из определения, приведенного в journal.tinkoff следует следующее: «Telegram -бот – это не большая программа в мессенджере, управляющая текстовыми командами в чате по принципу вопрос-ответ».

Технология используемая в Telegram не нова и использовалась ранее в различных продуктах, наиболее популярным из которых являются боты в мессенджере ICQ (на сленге: «Аська»), которые по запросу могли отправить запрашивающему гороскопы, анекдоты, различные статьи из «Википедии» и(или) могли помочь с переводом текста.

Звучит не плохо, а самое главное то, что Telegram входит в топ 3 самых популярных мессенджера в мире, а значит, вероятность того, что пользователю, с необходимостью закрыть потребность в каком-либо вопросе, даже не придется прибегать к установке дополнительного софта на свои устройства.

Так же к плюсам выбора можно отнести кроссплатформенность мессенджера, ведь пользователь сможет использовать написанного для этой платформы бота внутри Telegram на Android, iOS, Windows Phone, Windows, MacOS и даже Linux, а быстрая разработка такого проекта позволит быстрее выпустить его в продакшн (от англ. production) – термин, использующийся для обозначения некого процесса создания проекта и(или) творческого продукта, а так же предоставить пользователю удобное, доступное решение, для выполнения его задач.

Практическая значимость данного проекта заключается в создании чат-бота, автоматизирующего процесс отслеживания цены товара с площадки Wildberries и автоматического оповещения пользователя в чат о изменении базовой цены, а также в том, что данный чат-бот можно применить к любой торговой площадке внеся в него не значительные изменения, в том числе использовать для внутренних потребностей компаний, в которых есть похожие бизнес-процессы.

1. **ДЕТАЛЬНЫЙ АНАЛИЗ И ОБЗОР СРЕДСТВ ДЛЯ РАЗРАБОТКИ СЕРВИСА**
   1. **Общие понятия. Что такое чат-бот. Его основные функции. Классификация ботов.**

Как я описывал ранее, Telegram боты в частности или чат-боты в своем общем понимании — это микропрограммы, работающие как специальные аккаунты, работающие по принципу «вопрос-ответ», т.е. за ними не стоит человек, а полученные сообщения обрабатываются системой. Для обычного пользователя общение с ботом, а точнее отправка в него сообщений, выглядит как обычная переписка с вполне реальным человеком. Но на самом деле чат-боты, при должном исполнении, это умная программа «живущая» в мессенджере (-ах) и выполняющая запрограммированные в нее функции. Стоит отметить тот факт, что бот не может «переварить» абсолютно любую передаваемую в него информацию, лишь ту, что за скриптована его создателем.

Основные функции чат-бота:

К основным функциям любого бота, можно отнести такие функции, как:

- Поддержка пользователей. Чат-бот может в полной мере заменить раздел справка любого сервиса, ответить на типовые или как их чаще называют часто задаваемые вопросы пользователя, при чем может это делать 24 часа в сутки 7 дней в неделю, что, например может значительно снизить нагрузку на call центр организации или компании.

- Пользовательский сервис. С помощью чат-ботов можно осуществлять покупки, запрашивать услуги или делать заказы. В сфере розничной торговли, чат бот может стать незаменимым помощником бизнеса. Он может, например сделать выборку, по интересующему пользователя запросу, из базы данных и вернуть ссылку или список ссылок с товарами или услугами, удовлетворяющими запросу пользователя. При чем для бота это будет небольшой анализ, в то время как, если этим же вопросом займется реальный сотрудник – время выполнения такого запроса может быть кратно увеличено.

- Маркетинг. Реклама. Чат-боты – это мощное оружие маркетологов, т.к. корректно настроенный бот, может самостоятельно размещать нужную рекламу, ссылки на продукцию и т.п. в различных ресурсах от групп в том мессенджере, где он размещен, до сайтов, форумов, порталов, соц. сетей, проще сказать, он сможет разместить рекламу везде. Так же любой чат-бот можно с легкостью использовать как средство обратной связи, для сбора фитбека о продуктах или оказанных услугах.

- Работа внутри организации. Чат-бот прекрасно может справиться с работой внутри компании и обработкой служебной информации. При должной настройке, чат бот может обрабатывать запросы для Active Directory, управлять DHCP, обрабатывать консольные запросы к серверу, работать с внутренними сервисами компании. Может заниматься рассылкой и уведомлением сотрудников о изменениях в работе или графике, сообщать важные новости при чем делать он все это может без вашего участия по заранее запланированному разработчиком графику.

- Рекрутинг. HR функции. Чат-бот может стать для бизнеса неким HR фильтром, обрабатывая запросы от соискателей, парсить сайты с вакансиями и искать подходящих кандидатов, приглашая их на собеседование или напротив используя ряд заранее заготовленных вопрос-ответов отсекать тех, кто заведомо не годиться для работы в компании. В целом, используя шаблон, бот вполне сможет провести полноценное собеседование с кандидатом, собрать его личные данные и составить первичную характеристику. Благодаря этому бизнес сможет сократить затраты на отдел HR, уменьшив количество персонала, переложив часть функций на бот.

Существует множество классификаций чат-ботов, но немного разобравшись мы можем четко определиться, с тем, что на самом деле, говоря об основных классификациях, можно выделить только две, первая классификация — это бизнес классификация, вторая боты технического типа.

Боты относящиеся к бизнес классификации можно представить в виде такой схемы (представлена на рис. 1):



Рисунок 1. Бизнес классификация ботов.

Подробное описание типов бизнес классификации:

- Разговорные боты – они создаются для «общения», которое с ботами разговорного типа часто похоже на переписку с реальным человеком. Проще говоря – бот «болталка», в наше время такой бот может быть связан с генеративной сетью, вроде ChatGPT и в целом диалог с таким ботом будет мало отличим от общения с человеком.

- Боты ассистенты – чаще всего требуются для сбора данных, основанных на ответах пользователя. Пишутся с целью помощи бизнесу в получении пользовательских данных.

- Q & A (боты основанные на принципе «вопрос-ответ») – такие боты ожидают от пользователя запрограммированный вопрос и дают ожидаемый ответ, структура 1 вопрос = 1 ответ.

Боты, относящиеся к технической классификации, можно представить в виде такой схемы (представлена на рис. 2):



Рисунок 2. Техническая классификация ботов.

Подробное описание типов бизнес классификации:

- Основа: Бизнес-правила – в этом типе беседа пользователя и бота заранее продумана программистом и имеет древовидную структуру. Обычно содержит меню или инлайн кнопки (элементы меню доступные для взаимодействия с ботом отображаемые прямо внутри чата, в связке с сообщением от бота). Благодаря наличию меню, пользователь может прийти к заранее определенному решению своего вопроса. Получение ответов от бота в свободной форме (или, иными словами, в форме «вопрос-ответ») не предусмотрено.

- Основа: Искусственный интеллект – этот тип совершенно не похож на тип, где основу представляет бизнес-правила, такие боты не имеют заранее запрограммированной структуры. В них путь диалога определен неявно и за основу взяты тестируемые данные – данные полученные для обучения модели машинного обучения. Они должны иметь внушающий объем, заранее подготовленных данных, для обеспечения качественной работы.

- Боты гибридного типа – этот тип объединяет два описанных выше типа и является своего рода комбайном для выполнения заранее описанных программистом действий, но имеют возможность подключить к решению задач искусственный интеллект, чтобы предопределить желания и(или) намерения пользователя, а также извлекает и использует данные, полученные в ходе переписки для своего дальнейшего обучения и развития.

* 1. **Плюсы, варианты применения и статистика использования ботов для бизнеса.**

Чат боты кардинально изменили взаимодействие пользователей и бизнеса. Все больше компаний поднимают своих ботов с целью привнести свой продукт или услугу в новое и более увлекательной измерение, а также преследуя цель ускорения различных операций.

Статистика, актуальная на 2024 год:

* чат-боты являются самым быстрым средством коммуникации с бизнесом, их объем рынка за последние годы вырос на 92 процента, если взять 2020г. – то около четверти пользователей пользовались ботами для взаимодействия с бизнесом, а это на секунду на 13 процентов больше, чем год назад.
* Только за прошлый год, около 68 процентов пользователей со всего земного шара так или иначе использовали чат-ботов.
* Обработка около 70 процентов переговоров, происходит средствами чат-ботов.
* Чат-боты сейчас могут отвечать на более чем 80 процентов всех простых и наиболее распространенных запросов. Вероятно, поэтому по статистике на 2024г. Чат-боты активно используют более 1.4 миллиарда человек.
* Результаты последних опросов говорят о том, что около 30 процентов опрошенных не могут точно сказать, с кем именно они общались при обращении в тех. поддержку – с ботом или человеком.
* Маркетинговые инструментарий, представленный в чат-ботах, используется более чем 40 процентами пользователей ежедневно для осуществления покупок.

Статистика преимуществ ботов, актуальная на сегодня:

* Время отклика любого (даже самого медленного бота) в разы выше отклика человека.
* По результатам недавних опросов около 70% респондентов сочли ботов более удобными, просто потому что они дают более быстрые и корректные ответы.
* Так же к плюсам ботов, огромное количество людей относят их 24/7 график работы – да, боты не отдыхают … если только не настигнет баг :)
* Боты значительно снижают затраты бизнеса на тех. поддержку, клиенты, как показывает статистика быстрее получают ответы на свои запросы.

Преимущества использования ботов:

- Конфиденциальность. Это преимущество неоспоримый плюс и приоритет в выборе между соц. сетью и ботом для обмена конфиденциальными данными. Вообще все соц. сети по своей сущности являются открытыми и слабо защищенными платформами, через которые бизнесу легко взаимодействовать с пользователями и привлекать новых людей. Но соц. сети дорого платят за свою неспособность обеспечить конфиденциальность данных. Ярким примером для многих является Facebook, который в 2019г. выплатил огромный штраф в пять миллиардов долларов за утечку данных.

Мессенджеры же, в отличие от соц. сетей – это, по сути, канал связи, по которому данные могут быть переданы напрямую получателю. Так же многие мессенджеры оснащены механизмами безопасности, обеспечивающими сквозное шифрование передаваемых в них данных, что позволяет даже такому закрытому сектору как банковский рассматривать такую платформу как один из вариантов взаимодействия с клиентами.

- Репутация и связанные риски. Репутация для многих является основополагающей характеристикой успешной жизни. Именно поэтому использование мессенджеров для общения является более валидным – данные не станут общественным достоянием и предметом обсуждений. В личной переписке пользователь может сообщить куда больше и получить на много более точные данные, нежели общение в соц. сети.

- Просто, это удобно! В основе многих ботов лежит механизм асинхронности, а значит быстрота и все ваши переписки в одном месте – удобно!

Учитывая все вышеперечисленные доводы, а также данные официальной статистики, можно с уверенностью утверждать, что мессенджеры сейчас – это основное средство коммуникации.

Исследования предрекают увеличение продаж в сфере, связанной с искусственным интеллектом до 130 миллиардов уже к 2025 году, а ведь если оглянуться назад в 2018 – это были жалкие 10 миллиардов. Удивительно! Искусственный интеллект все глубже интегрируется в нашу жизнь, неся с собой и светлое будущее для мессенджеров в целом и ботов в частности.

На сейчас, многие топовые компании уже используют ботов, в связке и искусственным интеллектом, например:

* Мегафон и их бот с GPT и Миджорни на борту

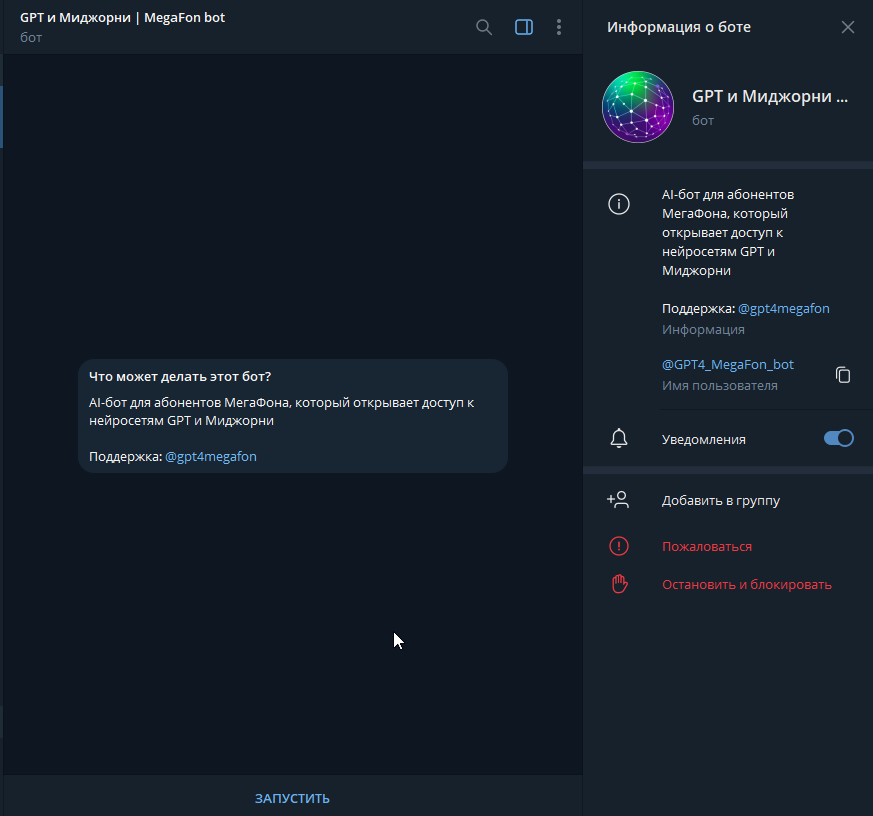


Рисунок 3. Скриншот чат бота компании Мегафон.

* Яндекс и их бот для поиска картинок

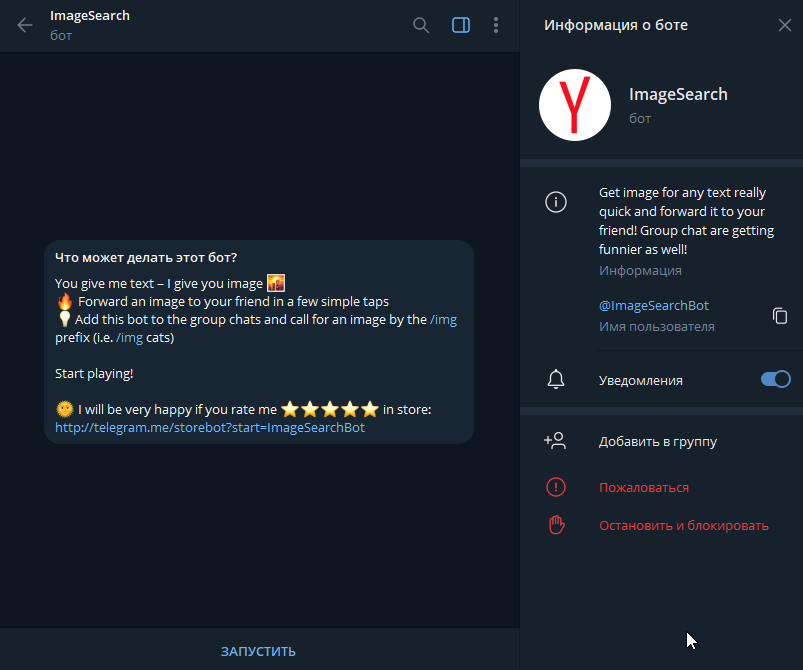


Рисунок 4. Скриншот чат бота компании Яндекс.

- GitHub и его бот для работы с сервисом:

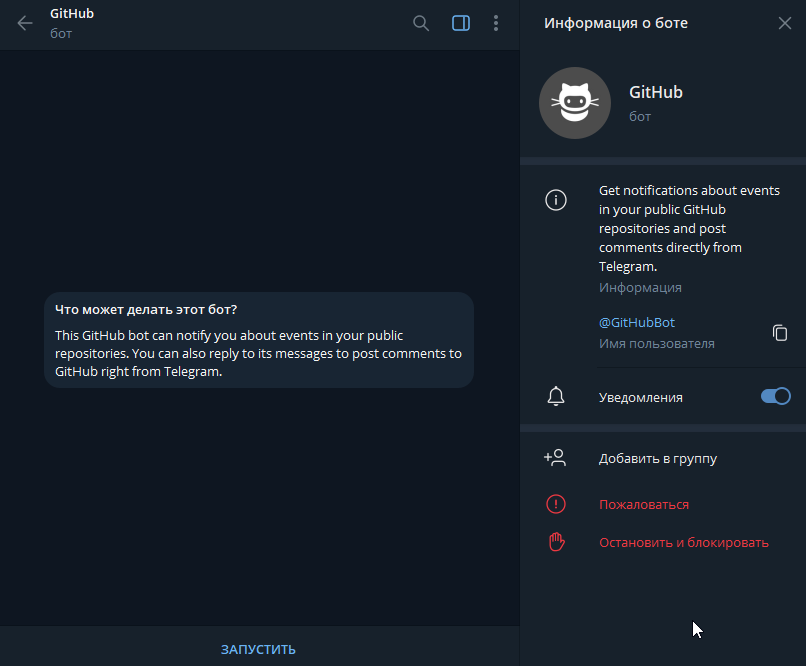


Рисунок 5. Скриншот чат бота сервиса GitHub.

Так же боты есть у таких компаний как Aviasales (представляет собой метапоисковик авиабилетов и отелей), Avito (получение уведомлений о новинках на платформе), Coinbase (бот для работы с биткоиновым кошельком) и т.д.

Чат-боты увеличивают продажи, позволяют в кратчайшие сроки привлечь пользователей, при правильном использовании они могут принести пользу при реализации продукции или предоставить, необходимую пользователю, информацию, более того, если обратиться к статистике более 40% запросов в ботах происходит в нерабочее время. Пользователь получат быстрый, своевременный ответ – бизнес, снижает затраты времени использования консультантов и специалистов call-центров. Любого работающего бота, легко быстро и без особых сложностей адаптировать к работе на любой платформе благодаря тому, что практически все мессенджеры кроссплатформенны. Поэтому покупая бота, можно за одну цену купить готовое решение, для таких мессенджеров как telegram, whatsapp, viber, прибегнув к незначительной настройке готового продукта при размещении его на необходимой бизнесу площадке.

* 1. **Выбор языка программирования и приведение статистики популярности языков на основе коммерческих рабочих проектов.**

В наше время существует огромное кол-во языков программирования, каждый по-своему хорош и отлично подходит для решения тех или иных задач, каждый программист при выборе подходящего именно для его проекта выбирает, отталкиваясь от множества параметров. Так же для написания чат-бота отлично подойдут многие, но мы обратимся к статистике популярности языков опираясь именно на те, что хорошо подойдут под написание чат-бота. Показанный на рис. 6 график основан на статистике, приведенной на портале vc.ru, являющимся на сегодня одной из самых крупных площадок в ру интернет-пространстве для предпринимателей и специалистов различных уровней квалификации из различных сфер деятельности и разномасштабных компаний.

Рисунок 6. Статистика используемых языков программирования для написания чат-ботов.

Как видно по диаграмме бесспорным лидером среди языков программирования для написания чат-ботов является Python, что, не странно учитывая то, что этот язык не без оснований считается самым универсальным, простым и легко интегрируемым в любые сферы использования.

Разберем более подробно все представленные на диаграмме языки программирования и постараемся разобраться в том, что каждый из них может предложить в вопросе разработки чат-бота.

Описание языков программирования:

- Python. Как упоминалось ранее, Python является, самым универсальным, простым и легко интегрируемым в любые сферы использования языком. Чат-бота на Python можно, при необходимости, развернуть на сайте или портале, в мессенджерах, а также в социальных сетях. А также еще одним огромным плюсом будет то, что специалистов способных написать бота на Python на рынке очень много, поэтому стоимость итогового продукта будет дешевле, чем аналогичный продукт на «специальном» языке. Кроме этого, данный язык поддерживает такие парадигмы программирования, как структурная, императивная, объектно-ориентированная, аспектно-ориентированная, а также функциональную. Python так же имеет автоматическое управление памятью, имеет полное самонаблюдение, множество механизмов исключений, поддерживает многопоточные вычисления и динамическую типизацию. Открытая разработка, простота изучений, особенно в самом начале, синтаксис позволяющий писать удобочитаемый код, огромное сообщество, адекватно относящееся к новичкам и способное помочь, огромное количество готовых решений и библиотек, отлично продуманные и легкие в использовании механизмы модульности – всё это станет бесспорным преимуществом языка в выборе его для проектирования и программирования на нем чат-бота.

- Java. Язык близкий к C++ своим синтаксисом. Так же язык довольно распространенный, широко и давно используемый программистами. На сегодня язык Java насчитывает огромное множество самых разнообразных библиотек, способных предоставить решения для самых разнообразных задач и требований, проще говоря, позволяет писать код «не изобретая велосипед». К тому же, этот язык из всех представленных будет самым удачным выбором, если бот будет написан на основе искусственного интеллекта.

- JavaScript. Это высокоуровневый язык, используемый обычно в связке с HTML, а также CSS. На нем чаще всего пишут фронтэнд составляющую сайтов и приложений, но тем ни менее можно встретить разработчиков, использующих JavaScript для написания чат-ботов и пусть таких разработчиков будет в разы меньше, чем тех же «питонистов», но несмотря на это их не мало на рынке. К преимуществам языка так же можно отнести его полезные функциональные настройки, то, что этот язык все время совершенствуется, написанное на JavaScript приложение с легкостью может осуществлять взаимодействие, даже через такие программы как Microsoft Office.

- PHP. Из всех представленных выше, этот язык используется реже всего для написания чат-ботов, ведь изначально этот язык использовался для web сегмента разработки. Но как бы это не было парадоксально именно web возможности языка и позволяют писать на нем ботов. Неоспоримым преимуществом языка считается то, что этот язык является серверным. Однако столкнувшись необходимостью в написании чат-бота и встав перед выбором, все же стоит отдать предпочтение одному из вышеописанных языков.

Как вывод из всего вышеописанного, можно сделать следующее, встав перед выбором языка программирования для написания чат-бота стоит учитывать два фактора, а именно какое расчетное количество обращений к чат-боту ожидается и на сколько «сложным» по своей структуре должен быть бот, то какие задачи предстоит решать готовому продукту будет состоять ответ на вопрос, какой язык программирования будет оптимальным. Так же стоит отметить то, что для написания простых ботов существуют и такие решения, как конструкторы, позволяющие без особых знаний языка и затрат времени запустить своего чат-бота, но, если мы рассчитываем на значительный приток пользователей в нашего бота и нам нужен продукт способный на что-то больше, чем показать погоду в нашем регионе – конструктор не наш выбор. Все же думаю каждому очевидно, что для написания чат-бота в мессенджере Telegram больше других подойдет именно Python, за свою простоту, гибкость, наличие готовых синхронных и асинхронных библиотек именно эти факторы позволят нам создать отлично работающий сервис.

* 1. **Выбор платформы для разработки чат-бота.**

- Вконтакте – одна из самых популярных российских соц. сетей, штаб-квартира которой расположена в Санкт-Петербурге. Сайт поддерживает мультиязычность (до 82 языков), но особую популярность завоевал среди русскоязычных пользователей. Соц. сеть вконтакте – позволяет отправлять сообщения с различным контентом, редактировать их, а также создавать собственные страницы и сообщества. Кроме того, платформа вконтакте позиционируется как платформа для бизнеса и его продвижения, а так же, как платформа позволяющая решать повседневные задачи используя мини-приложения. У вконтакте есть API, через который чат-боты могут взаимодействовать с пользователями. Через них можно отправлять текстовые сообщения, фото и видео файлы, местоположения, стикеры и т.п. Но у ботов вконтакте нет кнопок или каких-либо элементов непосредственного взаимодействия.

- Мессенджер Viber – это приложение, позволяющее получать или отправлять текстовые сообщения, звонить или отправлять видео сообщения, различные документы, картинки или видеофайлы, а также может работать с полученными файлами в автономном режиме. В мессенджере присутствуют публичные аккаунты, используемые для бизнеса. Бизнес-аккаунты можно использовать для взаимодействия с пользователями через чат-ботов или CRM системы.

- Мессенджер Telegram – это кроссплатформенная система для мгновенного обмена сообщениями. Позволяет делиться пользователям текстовыми, голосовыми, а также видео сообщениями, реализованы возможности отправки стикеров, фото, видео и прочих форматов файлов, возможно совершать аудио и видео звонки, устраивать прямые трансляции в каналах и (или) группах. Так же функциональность мессенджера может быть значительно расширена за счёт использования чат-ботов, а также стоит отметить, то, что Telegram был одним из первых кто сделал возможным добавление чат-ботов в мессенджер. Многие крупные компании, такие как Сбербанк, Почта России, Медуза, OneTwoTrip и т.п. в России уже используют эту возможность и запустили своих ботов в Telegram.

- Мессенджер WhatsApp – это бесплатный сервис из Америки, для мгновенного обмена сообщениями, с возможностью отправки текстовых, голосовых и видео сообщений, возможностью голосовых и видео звонков, обмена картинками, видео и прочими файлами. Аудитория мессенджера насчитывает более 1.3 миллиарда активных пользователей в течение месяца, а также более ста тысяч действующих ботов. В США WhatsApp является одним из ведущих мессенджеров. Набор функции предоставляемых мессенджером, для разработчиков, считается одним из самым продвинутым в этой сфере. Но стоит отметить, что предоставляемый мессенджером API сервис является платным, а также в настоящее время оплата этого сервиса с территории России заблокирована.

Из всего вышеописанного, можно сделать вывод, что Telegram на территории России сейчас – это единственная платформа, предоставляющая лучшее предложение по созданию чат-ботов, Возможность бесплатного размещения ботов внутри мессенджера, стабильность, регулярные фиксы по безопасности все это отлично подойдет для реализации этого проекта.

* 1. **Выводы по главе 1.**

В этом разделе были разобраны классификации чат-ботов и проведен краткий анализ типов ботов, в общих чертах описаны функции чат-ботов. Рассмотрели основные преимущества использования ботов для бизнеса, а также приведены ряд компаний использующие чат-боты. Разобрались с вариантами доступных платформ для развертывания чат-ботов и доступными для выполнения этой задачи языками программирования. Подводя итог и опираясь на приведенные ранее статистические данные, для разработки проекта мной были выбраны: язык программирования Python, база данных SQLite, среда разработки PyCharm, мессенджер Telegram.

1. **ОПИСАНИЕ МЕТОДОВ РАЗРАБОТКИ ПРИЛОЖЕНИЯ**
   1. **Язык программирования Python как идеальное средство для разработки**
      1. **Разбор необходимых модулей для разработки сервиса**

Из приведенной в первой главе статистики использования языков программирования было ясно, что идеально подходящим под наш проект языком программирования можно смело считать Python. Но кроме самого языка, необходимо так же подумать о том, какие библиотеки использовать для проекта, ведь, как ранее упоминалось, Python содержит в себе огромное количество библиотек, подходящих в том числе для написания чат-бота.

Итак разберемся, что нам понадобиться для выполнения поставленной задачи, разберем по пунктам, что мы планируем делать:

* Подключаться и получать актуальные данные с сайта Wildberries, в момент получения запроса от пользователя о товаре.
* Хранить полученные с сайта данные и предоставлять их пользователю по запросу.
* Уведомлять в автоматическом режиме о изменении данных (изменении цены) в позициях, добавленных пользователем.
* Возможность пользователю самостоятельно добавлять или удалять позиции в любое время.
* Добавить возможность управления данными администраторам сервиса.
* Добавить разделение пользователей на группы доступа с целью дальнейшей возможности монетизации сервиса.

Модуль sqlite3 – библиотека, написанная Герхардом Герингом. Представляет собой SQL интерфейс, совместимый с спецификацией DB-API 2.0, требует SQLite 3.7.15 или более новые версии. Вообще SQLite это C библиотека, которая представляет из себя облегченный вариант базы данных размещенная на диске, она не требует отдельного серверного процесса и дает обращаться к данным используя своеобразный, нестандартный вариант языка запросов SQL. Отлично подходит для небольших некоммерческих проектов и тестирования систем приложений с расчетом на то, что перед релизом база данных будет изменена на PostgreSQL, Oracle, MS SQL или подобные.

Модуль time – встроенный в Python модуль предоставляет функционал связанный со временем.

Модуль pywin32 – библиотека, чья основная роль заключается в том, чтобы предоставить разработчикам простой, быстрый доступ к Windows API, а так же немаловажной чертой этой библиотеки является то, что она позволяет выполнять COM-программирование с помощью Python.

Модуль socket – модуль обеспечивающий доступ к интерфейсу BSD socket. Доступен в многих современных системах Unix, macOS, Windows и т.п.

Модуль datetime – представляет функции для работы с датами и временем, поддерживается арифметика этих переменных для форматирования данных на выходе и манипулирования ими. Модуль помогает обрабатывать события времени такие как годы, месяцы, недели, дни, часы, минуты, секунды и т.д.

Модуль requests – эта библиотека является стандартным инструментом в Python для составления и обработки HTTP запросов. Простой API с легкостью придет на помощь и облегчит трудоемкий процесс создания HTTP запросов и позволит должным образом сосредоточиться на процессе взаимодействия со службами и использовании данных.

Модуль emoji – пакет emoji предлагает преобразование эмодзи в юникоде в названия эмодзи и наоборот, поддерживает абсолютно все наборы кодов, определенных консорциумом Unicode.

Модуль asyncio – мощнейший инструмент для написания параллельного кода с использованием async / await. Эта библиотека используется как основа многих асинхронных фреймворков Python, которые предлагают мощные и высокопроизводительные сетевые, а также веб-серверы, различные библиотеки, используемые для подключения к базам данных и т.д. Данный модуль идеально подходит для высокоуровневого сетевого структурированного кода.

Модуль logging – модуль реализующую гибкую систему для ведения журнала событий, происходящих в приложениях или библиотеках. Модуль предлагает API для ведения журнала, причем поставляет его вместе с модулем стандартной библиотеки. API позволяет фиксировать в журнале события ваши собственные сообщения или сообщения производителей.

Модуль urllib3 – это относительно свежий модуль, тесно связанный с HTTP и разработанный для Python. Он поддерживает загрузку файлов, для которых использовалось многокомпонентное кодирование, так же существует поддержка gzip, объединение пулов соединений и существует поддержка потокобезопасности.

Модуль locale – модуль, позволяющий программистам решать культурные проблемы в своих приложениях, не требуя при этом от разработчика знания особенностей каждой странны в которой планируется выпускать продукт.

Модуль configparser – модуль представляет собой класс ConfigParser, реализующий базовый язык конфигурации и способный обеспечить структуру аналогичную INI.

Модуль aiogram – классный, современный, мощный, а главное полностью асинхронный фреймворк, в написании которого использованы такие модули как asyncio и aiohttp. Он довольно прост в использовании, но при этом позволит сделать чат-ботов быстрее и что не мало важно проще. Имеет FSM, может отвечать в webhook, имеет канал для загрузки файлов с URL и как итог просто качественно ускоряет работу.

* + 1. **Что называется функцией в Python?**

Функцией называют объект, принимающий какие-либо аргументы и возвращающий решение. Обычно, функция определяется оператором def. Рассмотрим пример простейшей функции на рис. 7.

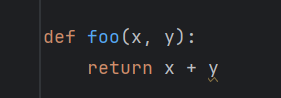


Рисунок 7. Пример функции в Python

Функция в Python может принимать сколько угодно много аргументов, а может вообще не принимать не одного. Аргументы могут быть как обязательными, так и необязательными. Пример такой функции изображен на рис. 8.

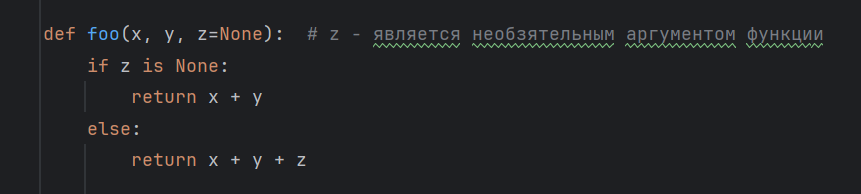


Рисунок 8. Пример функции с необязательным аргументом

* + 1. **Что называется условными операторами и циклами?**

Условные операторы являются основным инструментом выбора в Python. Операторы if / elif / else – выбирают какое действие предпринять, в том или ином случае, как правило опираясь на значения передаваемых, для проверки, переменных. Пример написания условных операторов и их синтаксис изображены на рис. 9.

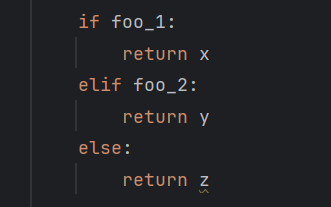


Рисунок 9. Пример написания и синтаксис условных операторов if, elif, else

Циклы в Python являются базовым инструментом, с помощью которых программисты могут быстро выполнять какие-либо повторяющиеся действия, тем самым автоматизирую процесс.

Цикл While – является одним из самых универсальных циклов в языке Python, при том являясь довольно медленным. Он не прекращает выполнение действия до тех пор, пока условие истинно. Пример синтаксиса цикла while изображен на рис. 10.

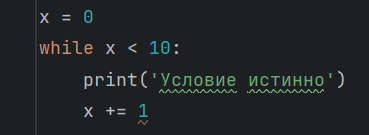


Рисунок 10. Пример синтаксиса цикла while

Цикл For – это более сложный оператор, можно смело утверждать, что он является менее универсальным, нежели цикл while, являясь при этом намного быстрее. Он способен проходить по любому итерируемому (повторяемому) объекту, списку, строке и т.п. выполняя во время каждого прохода код, заложенный в тело цикла. Возможно принудительно прервать цикл, например если нужно выйти из цикла, после достижения пересчетом какого-то значения, выход из цикла осуществляется командой break. Пример синтаксиса цикла for изображен на рис. 11.

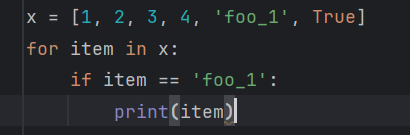


Рисунок 11. Пример синтаксиса цикла for

* + 1. **Что такое исключения?**

Обработка исключений, является одним из самых важных процессов, т.к. не обработанное исключение может вызвать сбой в программе или отображаемых данных, повредить данные или просто на просто сломаться целиком или вызвать сбой в ряде функций. К счастью, писавшие язык Python программисты, за ранее продумали этот вопрос и реализовали способ обработки исключений, такой как try / except / finally. Сообщения об ошибках дают программисту исчерпывающую информацию о том, в чем проблема и куда обратить внимание в случае проблем с работоспособностью кода, обработчики же ошибок дают возможность перехватить и обработать исключения в процессе выполнения программы, а так же, грамотно расставленные обработчики, дают возможность «не упасть» программе, а продолжить работать, даже если возникнет ошибка, передав исключение, через механизм логирования и дав разработчику возможность увидеть и поправить код, не доставив пользователю не приятного опыта использования. Пример самого простого блока обработки исключений представлен на рис. 12.

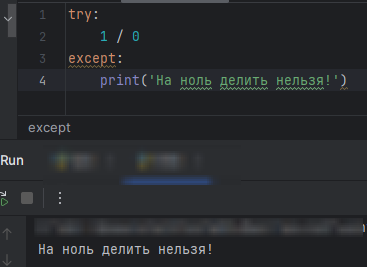


Рисунок 12. Конструкция try / except

* + 1. **Основной метод, используемый для проекта.**

Учитывая то, что наш чат-бот, в качестве основной информации, для своей работы будет использовать данные с торговой площадки wildberries.ru, логично предположить, что основой такого бота станет парсинг данной торговой площадки с целью извлечения данных о продуктах: фото товара, цена и т.п.

Под понятием парсинга понимают метод извлечения, с целью дальнейшей обработки, информации с веб-ресурсов (веб-сайтов). Метод парсинга, как правило, направлен на получение с ресурса неструктурированных данных в HTML формате и преобразования их в структурированный набор данных. Для выполнения парсинга, разработчику необходимо, проанализировать веб-сайт, а точнее структуру его страниц, понять по какой модели, данные попадают в виджеты на страницах сайта и извлечь для себя нужные ссылки, по которым, передавая ключи, можно будет получить, интересующий набор данных, для их последующей обработки и использования. Пример парсинга изображен на рис. 13.

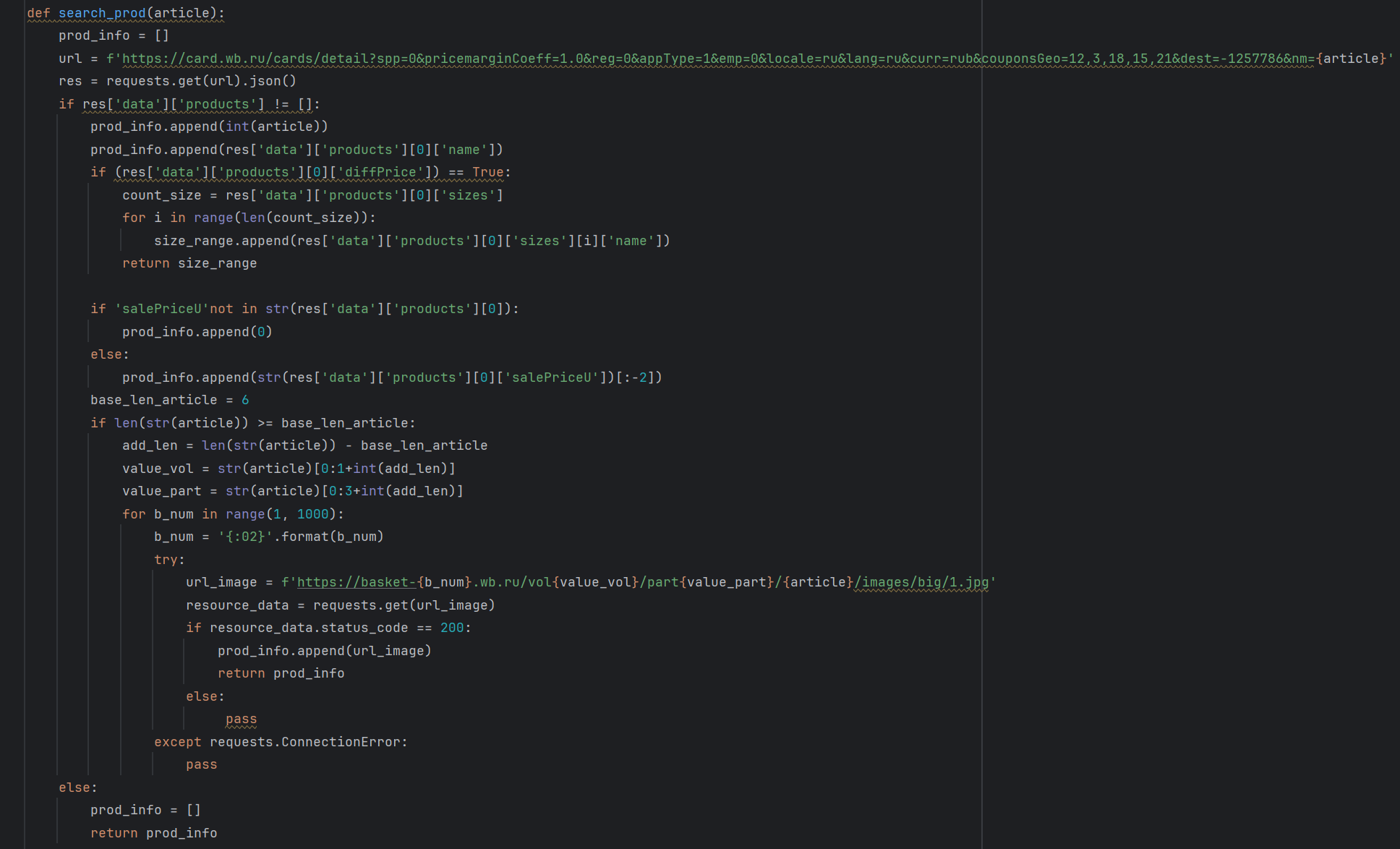


Рисунок 13. Парсинг карточки товара торговой площадки wildberries.ru

* 1. **Описание компании Wildberries**

Торговая площадка / Интернет-магазин Wildberries – на сегодня, является крупнейшей российской компанией. Создана компания была в 2004г. и выбравшая сферу для работы в электронной коммерции в сегменте B2B. Ассортимент площадки начитывает более 3.6 миллионов товаров от более чем 30-ти тысяч как российских, так и импортных брендов и включает в себя одежду, обувь, товары для детей, различную бытовую технику и электронику, различного рода аксессуары, книги, бытовую химию, ювелирные украшения, продукты, товары для дома и т.д. Ежедневно, более 750000 заказов, оформляется именно на торговой площадке Wildberries. Штат сотрудников постоянно растет и на сейчас начитывает около 32000 человек. Wildberries, по их словам, активно набирает молодых специалистов и выпускников, а так же, что их интернет-магазин представляет самые комфортные условия для профессионального развития и карьерного роста. Так же, по данным из аналитического агентства Data Insight, на протяжении трех последних лет, компания Wildberries становиться крупнейшим ретейлером России по объему онлайн-продаж. Так же в компанию на регулярной основе требуются сотрудники из IT отрасли. На рис. 14 изображены фото офиса компании Wildberries.



Рисунок 14. Фото офиса компании Wildberries.

* 1. **Выводы по главе 2.**

В этом разделе мы описали методы разработки, а если быть точнее более подробно углубились в описание главного средства разработки чат-бота, языка Python. Так же мы описали основные функции, операторы, методы для написания качественного продукта. Так же описали, что из себя представляет компания Wildberries, ее сильные стороны, на которых, собственно, и основан выбор компании, на базе которой будет строиться чат-бот.

1. **Описание разработки и Практическая часть реализации сервиса**
   1. **Разработка IT проекта**
      1. **Устав проекта**

• Название проекта: Разработка чат-бота для уведомлений о снижении цены товара, продающегося в интернет-магазине Wildberries на языке Python.

• Ожидаемые результаты проекта: Автоматизация отслеживания снижения цены интересующего товара пользователем, без необходимости прохождения дополнительных регистраций и установки дополнительного софта.

• Ограничения:

» интерфейс мессенджера Telegram

» ограничение срока разработки проекта от стадии планирования до релиза до 28.02.2024г.

• Параметры, запрашиваемые к платформе для разработки проекта:

» Операционная система – предпочтительной системой для разработки и тестирования проекта является Windows 10, 11 (выбор системы обусловлен тесной связью запланированной структуры сервиса для оповещения о изменении цены – при изменении операционной системы, необходимо изменить план по разработке сервисной части приложения)

» Оперативная память – не менее 4Gb

» Процессор – Intel Core i5 / i7 с тактовой частотой не ниже 2.5GHz

» Жесткий диск – объем носителя не менее 120Gb

* + 1. **Структура проекта**

Разработка проекта при помощи методологии Microsoft Business Solutions Methodology включает в себя 6 этапов, такие как:

* Подготовка проекта
* Анализ
* Дизайн
* Разработка / Тестирование
* Развёртывание
* Сопровождение в начале становления проекта

Проект содержит несколько уровней декомпозиции:

* Начало работ
* Подготовка проекта:
  + Предварительное планирование проекта
  + Определение по тем результатам, что ожидаем от проекта, сроков выполнения и границ проекта
* Анализ:
  + Предварительный анализ средств управления сервисом
  + Проанализированы технические / программные средства, необходимые для реализации проекта
* Дизайн:
  + Разработка дизайн решения и макетов интерфейса
* Разработка / Тестирование:
  + Разработка работы бота
  + Установка библиотек, подготовка requirements.txt
  + Создание базы данных, наполнение базы тестовыми данными
  + Формирование кода проекта на языке Python
  + Подготовка к интеграции чат-бота в мессенджер Telegram
  + Подготовка к запуску сервисной части на платформе операционной системы Windows 11
  + Реализация проработанных ранее элементов интерфейса проекта\
  + Проведение тестирования функций проекта / функций сервиса
  + Устранение зафиксированных механизмом логирования неисправностей в коде
* Развертывание проекта:
  + Покупка сервера
  + Размещение и настройка проекта, подготовка сервисов
  + Подготовка инструкций по использованию сервиса
  + Проведение обучения (если необходимо) персонала, занимающегося обслуживанием сервиса
* Завершение
  + 1. **Риски проекта**

Управление рисками на сейчас это одно из самых актуальных и развивающихся направлений менеджмента. Принято считать, что такое управление позволяет более оптимально в плане эффективности и в расчете на затраченные средства контролировать риски, которые будут соответствовать поставленным задачам бизнеса. В табл. 1 рассматриваются риски проекта.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| п/п | Возможные риски | Должность / выполнение работы | Теоретические варианты предотвращения рисков | План действия |
| 1. | Не способность выполнения работы в указанный срок в связи с большим объемом работ | Разработчик | Составление плана и согласование сроков выполнения работы. Разбитие на части отдельных задач | Наем дополнительных сотрудников для реализации проекта в заданный срок |
| Календарные риски | | | | |
| 1. | Не выполнение задач с запланированным дедлайном | Разработка кода на Python | Поиск и наем дополнительного сотрудника, наделенного достаточным уровнем знаний для выполнения задачи в срок | Привлечь на помощь сотрудника, обладающего более обширными знаниями в области разработки на Python |
| 2. | Задачи недостаточно объемные / несоответствующие заявленным | Устранить несоответствия | Добавление дополнительных задач | Необходимо в план добавить задачу о функциональном тестировании |

Последний пункт о добавлении функционального тестирования добавлен с целью предотвращения ошибок, которые могут возникнуть / возникли во время разработки кода проекта. Все те риски, что могут возникнуть / возникли из-за недостатка ресурсов, могут быть предотвращены заранее за счёт введения дополнительного обучения перед началом проекта. Повысив за ранее квалификацию исполняющего персонала, мы сможем обеспечить выполнение проекта качественно и в срок.

* 1. **Практическая часть реализации бота**
     1. **Генерация токена для интеграции чат-бота в мессенджер Telegram.**

Чат-бот для своей работы должен быть интегрирован в Telegram. Для этого у мессенджера реализован служебный бот: BotFather (информация о боте представлена на рис. 15, позволяющий сформировать уникальный токен, благодаря которому можно будет произвести интеграцию проекта в мессенджер.

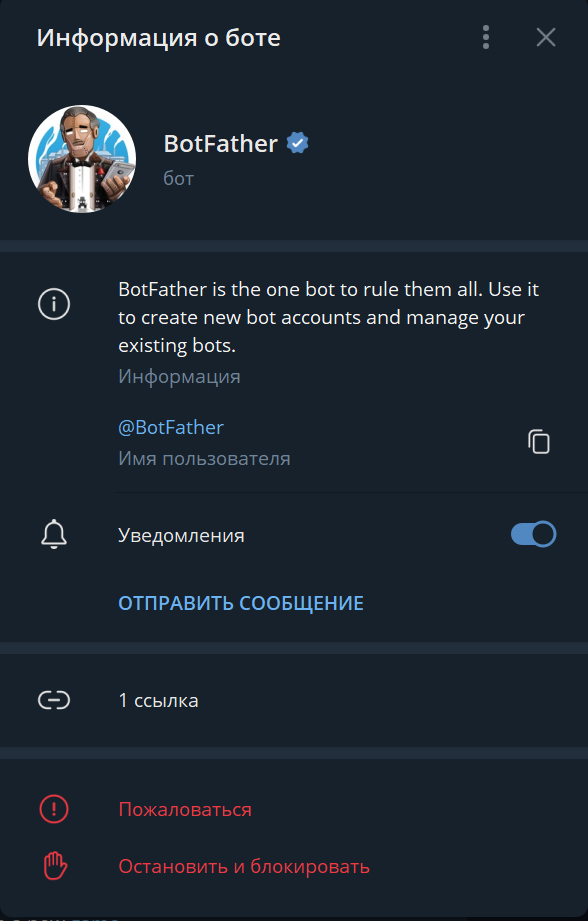


Рисунок 15. Служебный бот Telegram

Бот весьма функционален и имеет множество возможностей, таких как: создание нового бота (генерация имени / токена), просмотр созданных ботов, редактирование созданных web приложении, возможность смены имени действующему боту, возможность добавления доп. меню боту и т.д. со списком всех доступных в BotFather команд можно ознакомиться на рис. 16.



Рисунок 16. Список доступных в служебном боте Telegram команд

Генерируем в BotFather токен и сохраняем его в безопасном месте. Для проекта, место для сохранения данных, будет использоваться .ini файл – эти файлы являются конфигурационными для компьютерного ПО, состоящие из теста, со структурой ключ / значение, а также содержат разделы, включающие в себя эту структуру. Названия файлов произошли от слова инициализация, используемого в ОС MS-DOS, которая в свою очередь и сделала популярными этот тип файлов.

В структуре приложения создаем settings.ini, куда в раздел [bot] помещаем пару ключ значение, где ключ определяем как token, а значение – сгенерированный служебным ботом Telegram токен для нашего чат-бота. Пример файла settings.ini и его структурой можно увидеть на рис. 17.

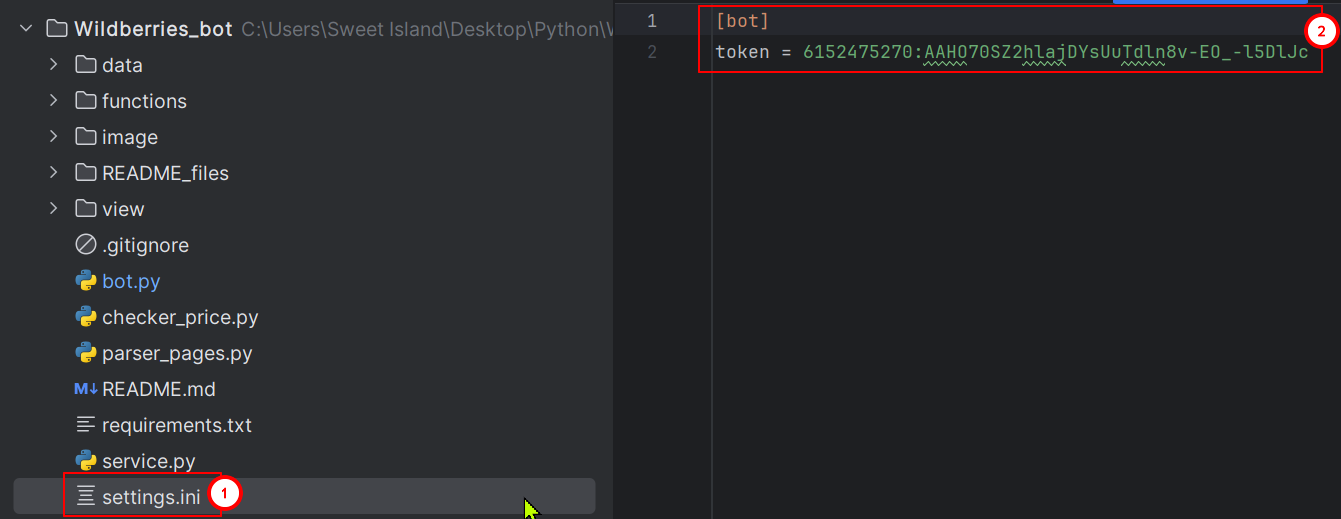


Рисунок 17. Структура файла settings.ini

* + 1. **Парсинг сайта Wildberries.ru**

Ранее мы определились с тем, что для карточки товара в боте, нам хотелось бы видеть фото товара, его актуальную на момент запроса цену (без учета индивидуальных скидок и промо акций), а также название товара. Переходим в интернет-магазин, «проваливаемся» в карточку любого товара, прожав F12 в браузере или кликом правой кнопки мыши по странице, инициализируем механизм просмотра кода страницы. Где путем анализа HTML кода страницы получаем нужные нам URL-ы. Пример получения URL-а фотографии из карточки товара изображен на рис. 18.

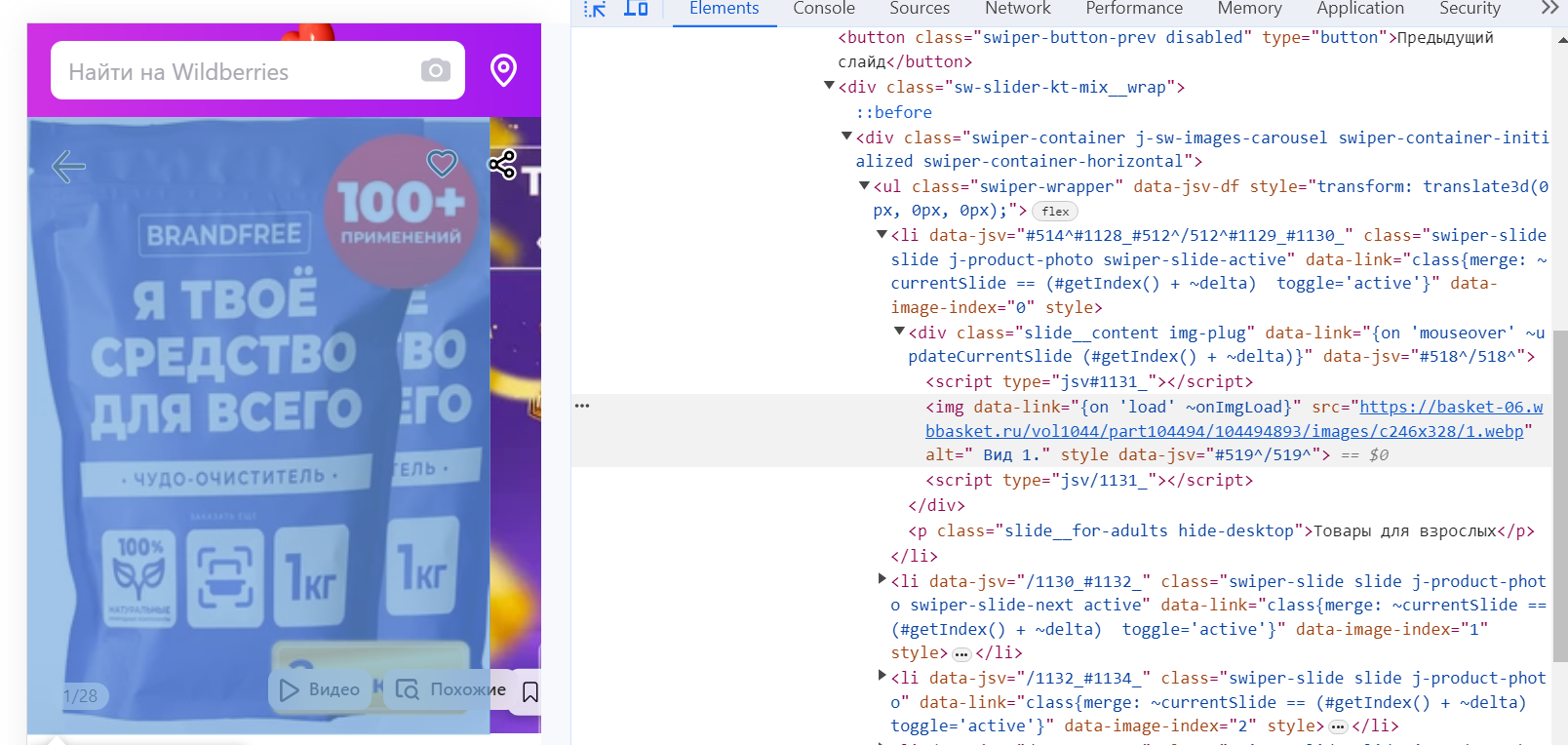


Рисунок 18. Извлечение из кода страницы URL -а изображения.

* + 1. **Создание базы данных и необходимых для чат-бота таблиц**

Местом для хранения была выбрана база данных SQLite. Перейдем к подготовке базы данных, для хранения данных. Как описывалось ранее в разделе 2.1.1 главы 2, для работы с базой данных мы будем использовать внешнюю библиотеку sqlite3, инициализируем установку модуля в виртуальное окружение. После инициализации модуля создадим новую базу данных, назовем ее config.db и создадим там 2 таблицы: «users» и «users\_info». Посмотреть процесс создания базы данных и необходимых таблицы с использованием библиотеки sqlite3 можно на рис. 19.

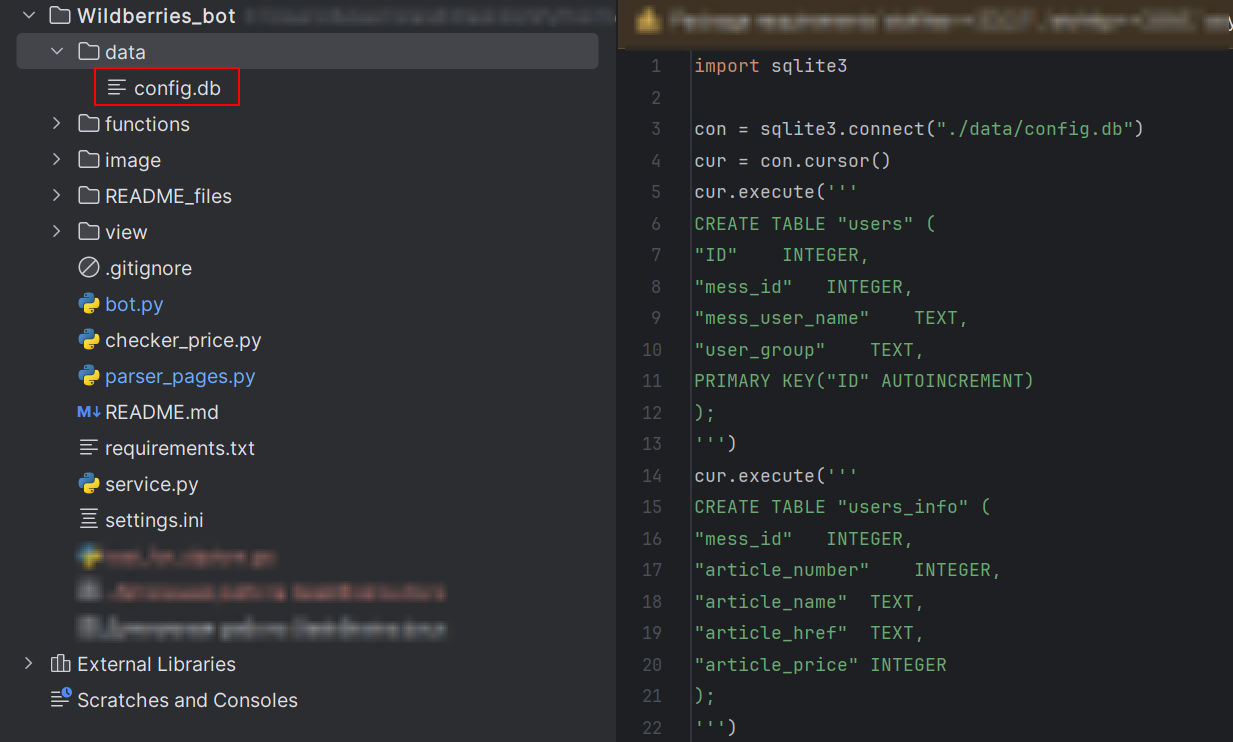


Рисунок 19. Создание базы данных и таблицы для проекта.

Далее мы более подробно рассмотрим структуру созданной базы данных. Структура отражена на рис. 20.



Рисунок 20. База данных чат-бота.

Таблица «users» хранит в себе данные о пользователях, подключившихся в боте, их telegram id и помещает значение в столбец mess\_id, полное имя пользователя и помещает значение в столбец mess\_user\_name и группа доступа и помещает значение в столбец user\_group. Разберем более подробно каждый из сохраняемых параметров.

Telegram ID - некий цифровой код присваивающийся системой мессенджера каждому аккаунту во время его регистрации. Он корректно сохраняет данные по пользователе и образует связь с конкретным пользователем мессенджера. По ID возможно осуществить обратную связь с пользователем, например если понадобиться от лица бота отправить сообщение конкретному пользователю.

Полное имя пользователя – если быть более точным параметр сохраняемый в этом поле это message.from\_user.full\_name, этот параметр выводит полные имя и фамилию пользователя, сохраненные в Telegram (политика мессенджера не обязывает пользователя передавать в качестве этих параметров именно ФИО конкретного пользователя, по этому вместо этих данных часто можно получить Emoji, произвольный набор текста или символов). Мы сохраняем это значение для более корректного обращения к пользователю, если такая необходимость возникнет.

Группа доступа – определяет уровень доступ к боту, т.е. она буквально определяет, какое меню будет отрисовано пользователю, какие уведомления будет получать пользователь и т.п. Так пользователь с группой доступа dev при авторизации получит больше функций и возможностей в плане управления системой бота или базой данных на прямую через бот, чем пользователь группы user. Так же с помощью групп доступа, можно полностью ограничить доступ к боту. Ознакомиться c таблицей «users» можно на рис. 21.

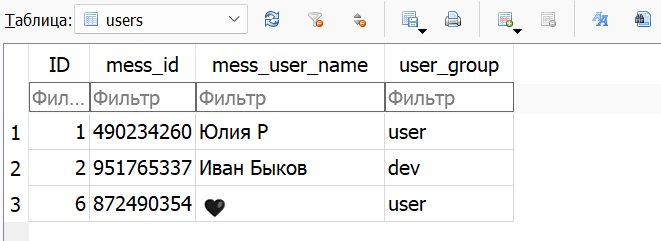


Рисунок 21. Таблица «users»

Таблица «users\_info» хранит в себе данные о позиции, добавленной к отслеживанию пользователем с Telegram ID, а именно telegram id и помещает значение в столбец mess\_id, артикул товара и помещает значение в столбец article\_number, название товара и помещает значение в столбец article\_name, ссылка на картинку товара и помещает значение в столбец article\_href, а также цену товара и помещает значение в столбец article\_price. На рис. 22 можно более подробно ознакомиться с таблицей «users\_info».

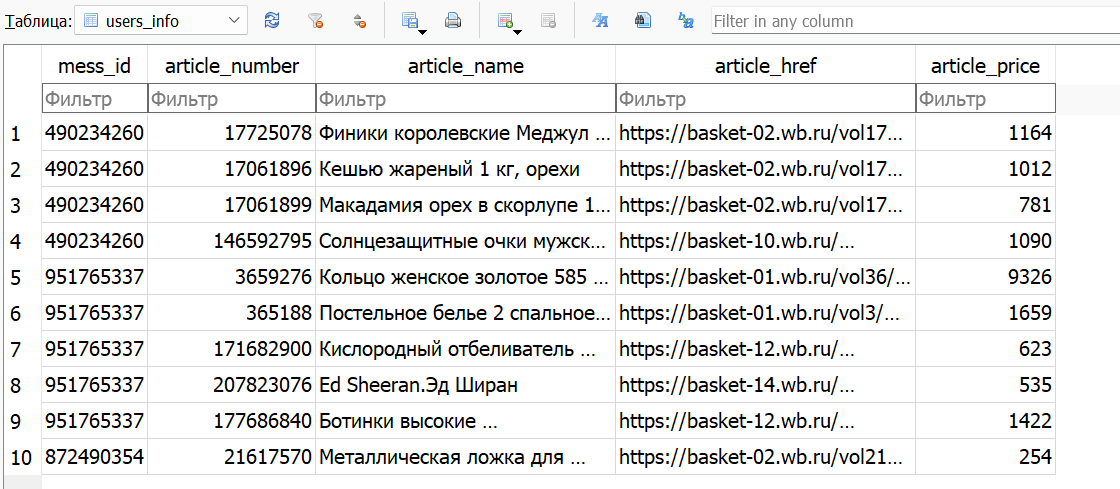


Рисунок 22. Таблица «users\_info»

* + 1. **Сервис для автоматического уведомления о изменении цен.**

Сервис для автоматического уведомления пользователей это важная часть нашего проекта. Добавлять и удалять позиции – важно, но если пользователь не будет получать оповещение при изменении цены на сайте – смысл всего приложения будет утрачен. Сервис мы будем писать, как упоминалось ранее, при помощи библиотеки pywin32. С кодом сервисной части приложения можно ознакомиться на рис. 23.



Рисунок 23. Код сервиса WildberriesNotifService

Разберем подробнее то, как же работает этот сервис. Итак, для начала импортируем все необходимые библиотеки и функции (именно функции CRUD отдельно описанные для работы с данными, представленными в базе данных). Описываем класс сервиса, называем сервис, обозначаем его описание, для дальнейшего удобства использования сервиса и определяемся с таймаутом выполнения основного скрипта сервиса, а именно выставим таймаут равный 60-ти секундам. Наш сервис будет зациклен, у нас нет необходимости обращения к нему, поэтому он будет в автономном режиме выполнять поставленную ему задачу. Ознакомиться более подробно можно на рис. 24.

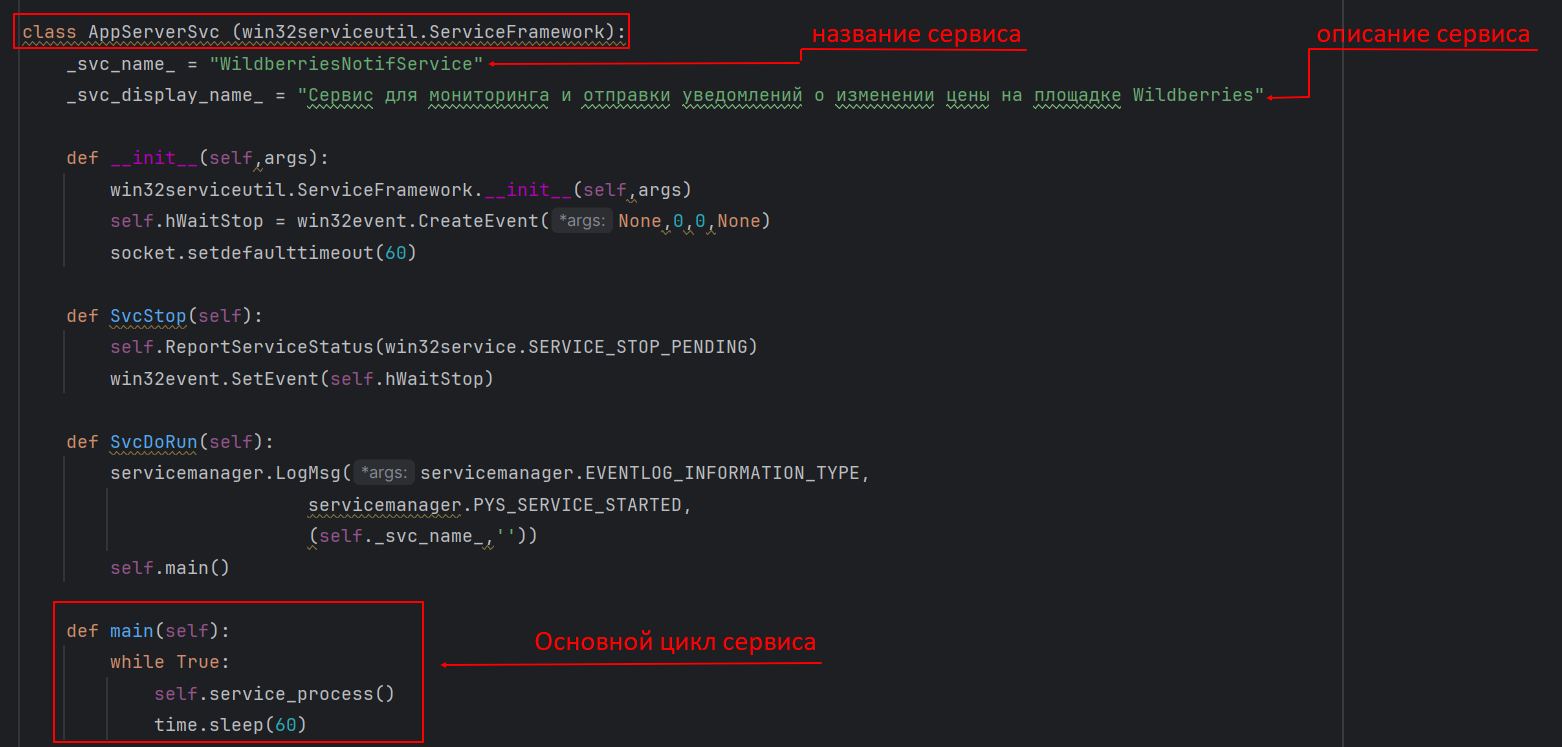


Рисунок 24. Описание основной части сервиса.

Сервис после запуска начинает в цикле while бесконечно запускать функцию service\_process c таймаутами в 60 сек. Функция service\_process считывает все данные из таблицы «user\_info», далее идет на сайт wildberries.ru находит карточку товара и парсит ее – извлекая новое значение цены товара. Если цена товара не изменился, товар пропускается и код переходит к проверке следующего товара. Если же цена стала, ниже используя возможность отправки сообщения, через API сервис мессенджера Telegram, пользователь получит уведомление о изменении цены, так же будет сделана пометка в файл с логами сервиса о успешной или не удачной отправке сообщения пользователю, предварительно изменив цену в базе данных. Более подробно посмотреть на процесс можно на рис. 25.

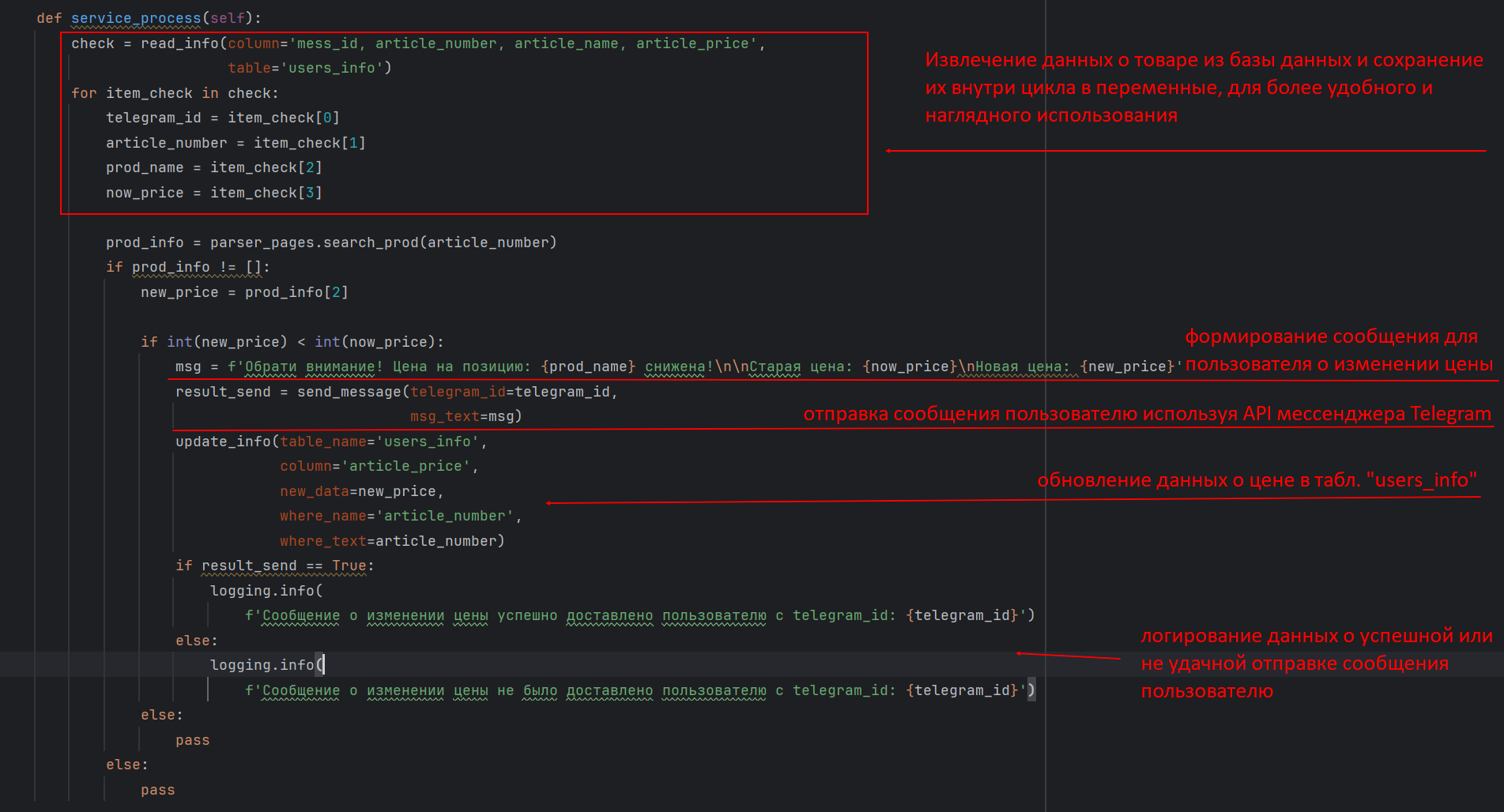


Рисунок 25. Описание работы функции service\_process.

Установка, запуск, остановка и удаление сервиса, происходит путем инсталляции на сервер Python 3.8, установки в него библиотек, зафиксированных в файле requirements.txt. Команды для запуска сервиса и пример использования можно увидеть на рис. 26.

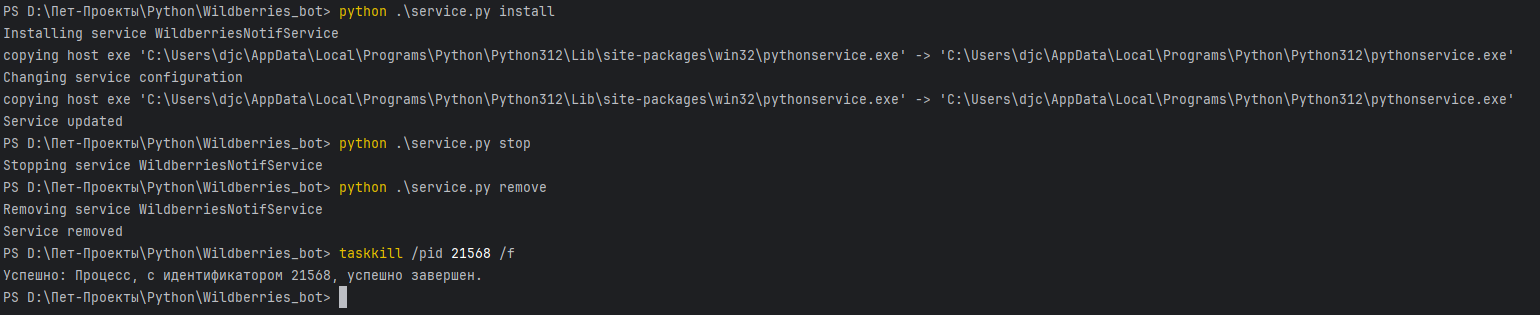


Рисунок 26. Пример команд для управления сервисом.

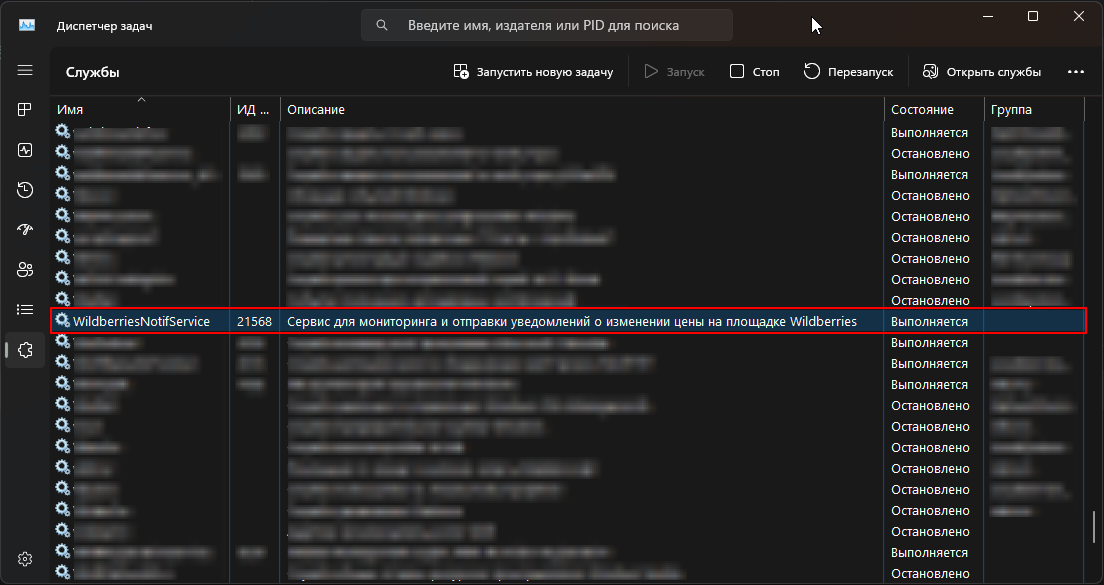
Установка сервиса, запуск, остановка и удаление – требуют повышенных прав, поэтому для этих задач необходимо использовать консоль powershell запущенную с правами администратора. Так же необходимо помнить о том, что сервис зациклен, поэтому в каждый момент времени он активен, а значит его остановка или удаление невозможны штатными командами stop и remove – необходимо будет прибегнуть к помощи команды taskkill и удалить процесс используя его PID, который можно посмотреть, через консоль или в диспетчере приложений. Посмотреть пример запущенного приложения можно на рис.27.

Рисунок 27. Пример запущенного и исправно работающего сервиса WildberriesNotifService

Исправно работающий сервис, будет получать цены из интернет-магазина wildberries.ru сравнивать их с теми ценами, что сохранены в базе данных и в случае снижения оповещать пользователя о изменении цены. Посмотреть пример подобного уведомления можно на рис. 28.

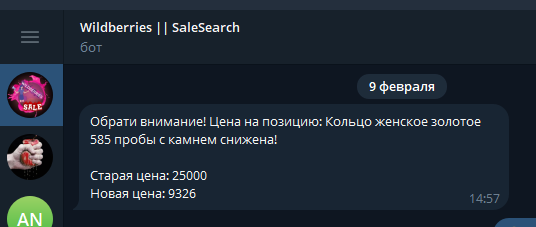


Рисунок 28. Оповещение в чат бота о изменении цены отслеживаемого товара.

Было бы правильнее для управления сервисом написать GUI приложение на PyQT5 или PyQT6, но в виду сжатых сроков выполнения проекта, было решено оставить консольный способ управления сервисом.

* + 1. **Описание работы бота**

Мы определились, что будет использовать асинхронную библиотеку aiogram для создания нашего бота. Начнем с подготовки, импортируем все необходимые для работы библиотеки и определимся с тем, где будем хранить получаемые от пользователя промежуточные данные, данный процесс в теории алгоритмов называется конечным автоматом или FSM – Finite State Mashine). Принцип его работы можно посмотреть на рис. 29.



Рисунок 29. Схематическое обозначение FSM процесса.

В библиотеке aiogram, пожалуй, один из самых удачных вариант реализации FSM. Для нашего проекта хранилищем для состояний выбрано самое базовое хранилище, а именно MemoryStorage – хранилище, использующее для сохранения данных оперативную память, данное хранилище идеально походит для небольших проектов или для реализации примеров, но для большого проекта можно найти более подходящее хранилище или описать свое собственное.

Посмотреть подробнее на то, какие библиотеки импортируются для бота и как реализуется подготовительная часть, можно на рис.30.

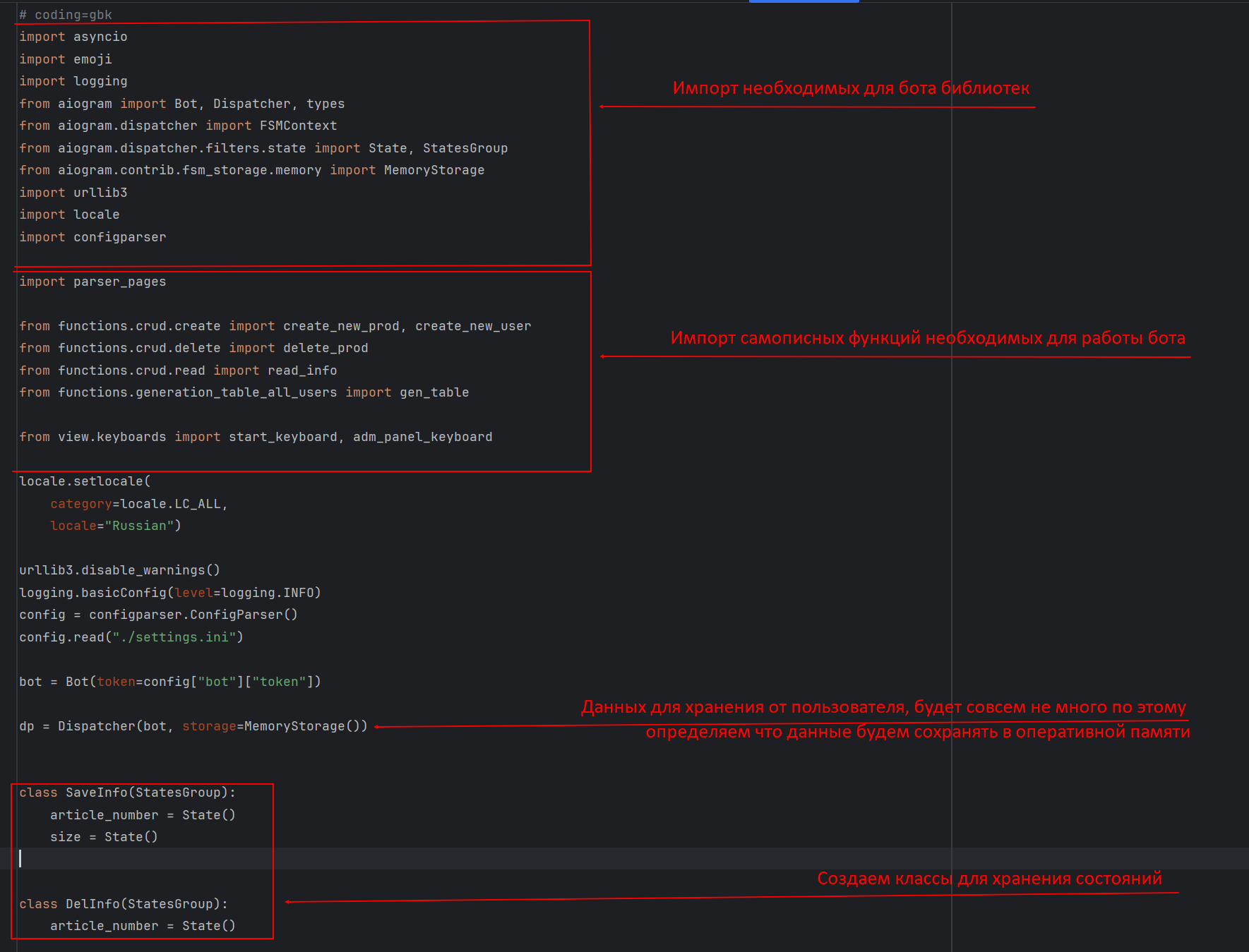


Рисунок 30. Импорт необходимых библиотек и подготовка хранилищ для сохранения состояний.

Управление ботом и его функциями происходит, через команды, базовой командой для любого бота является команда Запустить, бот понимает ее как /start

Отправив команду /start в нашем боте, происходит следующее, первым делом, он пытается найти пользователя в своей базе данных, если совпадений не найдено, он создаст новую строку в таблице «users» и добавит доступные данные о пользователе в систему и вернет пользователю приветственное сообщение и базовую (пользовательскую) клавиатуру. Если же совпадение будет найдено, он просто сообщит пользователю о успешной инициализации и отрисует ему клавиатуру, соответствующую уровню доступа установленному в базе данных пользователю. То, как описана функция /start можно посмотреть на рис.31

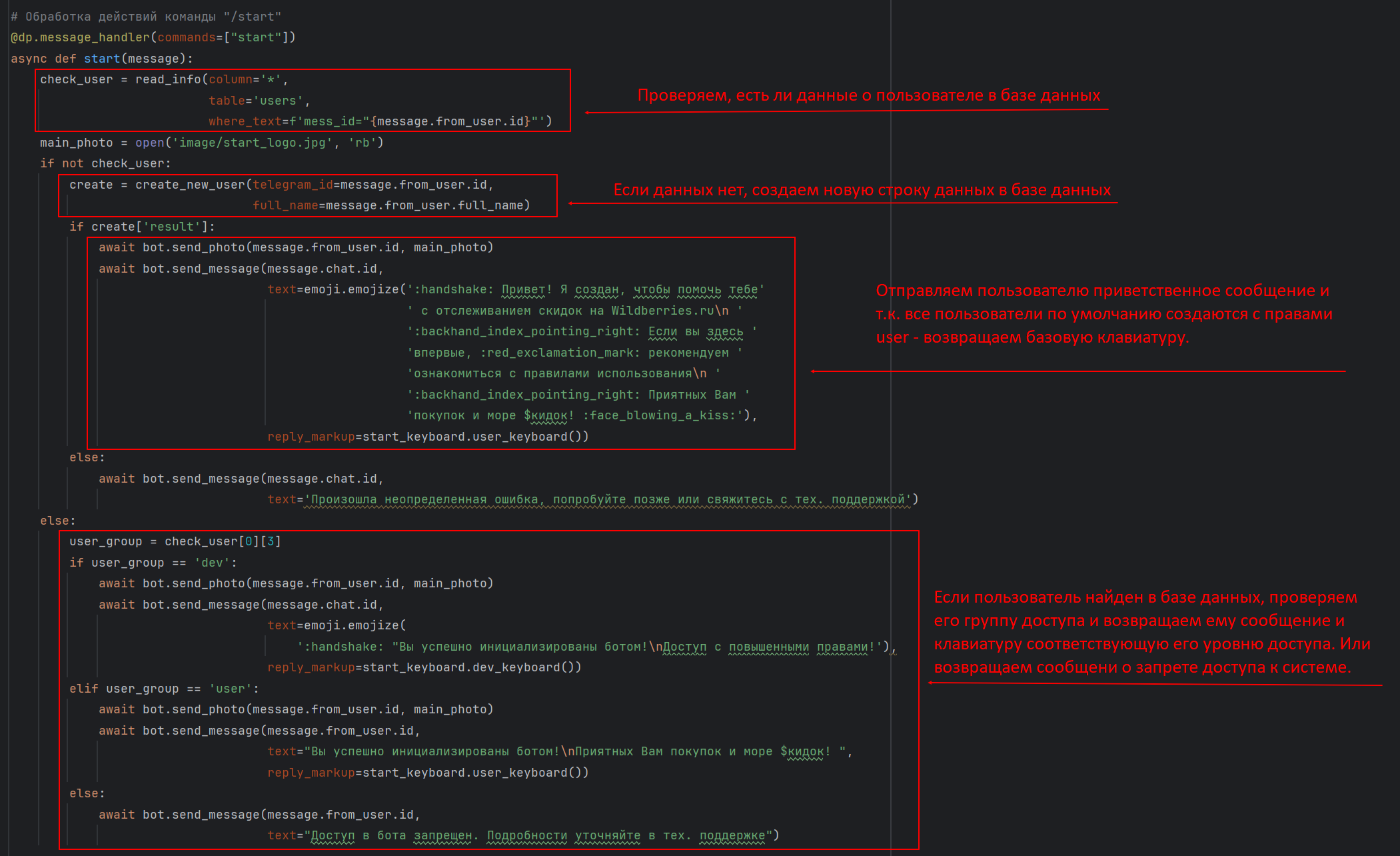


Рисунок 31. Описание команды /start

Кроме того, в боте для пользователей с уровнем доступа user, доступны команды Добавить, Удалить, Список моих хотелок и Правила работы бота.

Для пользователей с уровнем доступа dev, дополнительно доступна кнопка Панель управления. Познакомимся подробнее с каждой из представленных в меню кнопок:

* Добавить: дает возможность передать артикул из интернет-магазина Wildberries.ru выбранного товара и сохраняет его в базу данных, сообщая пользователю о успешном сохранении данных и начале мониторинга цены выбранного им товара.
* Удалить: удаляет строку с товаром из базы данных.
* Список моих хотелок: показывает сохраненные пользователем позиции в базе данных.
* Панель управления: дает возможность посмотреть список всех пользователей базы данных из таблицы «users», так же дает возможность отправки сообщения от лица бота любому пользователю бота.

Дизайн и общий вид бота можно посмотреть на рис. 32.



Рисунок 32. Общий вид чат-бота внутри мессенджера.

* 1. **Вывод по главе 3**

В целом представленных в боте функций достаточно для покрытия отписанных в тех. задании требований по отслеживанию изменения цены товара в интернет-магазине Wildberries.ru, но чат-бот также обладает огромным потенциалом для дальнейшей доработки, добавления функций и обработчиков, консольных и кнопочных команд. В данной главе был описан процесс практической реализации бота, работы его внутренних сервисов, продемонстрированы фрагменты кода и показан алгоритм работы программы.

**ЗАКЛЮЧЕНИЕ**

Автоматизация – это смена затрат сил на рутинные и простые задачи службам способным выполнять их автоматически.

Чат-боты, это современные, полезные средства автоматизации, способные с использованием языка программирования автоматизировать многие процессы.

Функции чат-ботов:

* Добавление автоматизации рутинным процессам
* Поиск, сбор и отправка данных
* Отправка автоответов и авто подтверждений
* Взаимодействие с различными внешними и внутренними системами

Актуальность темы обусловлена тем, что в вечно растущем мире торговли, активном и динамически развивающемся мире, необходимость в сокращении расходов существует особенно остро и многие пользователи, для получения актуальной информации вынуждены затрачивать свое время на рутинные операции: зайти на сайт, посмотреть цену – наш чат-бот решает эту проблему и устраняет рутинную задачу выполняя всю работу за пользователя.

В дипломе были представлены анализы и средства тех. средств по разработке чат-бота, рассмотрены общие понятия, преимущества использования чат-ботов, статистика и аналитические данные по поводу выбора языка для написания бота и приведены другие средства способные помочь в реализации подобного проекта.

**Список использованных Источников**

1. Википедия [Электронный ресурс] – URL: <https://ru.wikipedia.org/wiki/Telegram>
2. Конечные автоматы (FSM) [Электронный ресурс] – URL: <https://mastergroosha.github.io/aiogram-2-guide/fsm/>
3. Работа с SQLite в Python [Электронный ресурс] – URL: <https://habr.com/ru/articles/754400/>
4. sqlite3 — DB-API 2.0 interface for SQLite databases [Электронный ресурс] – URL: <https://docs.python.org/3/library/sqlite3.html>
5. Интернет-магазин Wildberries.ru [Электронный ресурс] – URL: <https://www.wildberries.ru/>
6. О компании «Wildberries» [Электронный ресурс] – URL: <https://career.habr.com/companies/wb>
7. Парсинг сайтов на Python [Электронный ресурс] – URL: <https://idatica.com/blog/parsing-saytov-na-python-rukovodstvo-dlya-novichkov/>
8. Emoji [Электронный ресурс] – URL: <https://emoji-python.readthedocs.io/en/latest/>
9. Циклы в Python: принципы работы и примеры использования [Электронный ресурс] – URL: <https://blog.skillfactory.ru/tsikly-v-python/>
10. AIOGram Framework for Telegram Bot API written in Python 3.7 with asyncio [Электронный ресурс] – URL: <https://sourceforge.net/projects/aiogram.mirror/>
11. За границей Hello World: полный гайд по разработке Telegram ботов с помощью Python и Aiogram 3. Часть 1 [Электронный ресурс] – URL: <https://habr.com/ru/articles/732136/>
12. configparser — Configuration file parser [Электронный ресурс] – URL: <https://docs.python.org/3/library/configparser.html>
13. locale — Internationalization services [Электронный ресурс] – URL: <https://docs.python.org/3/library/locale.html>
14. Guide to Sending HTTP Requests in Python with urllib3 [Электронный ресурс] – URL: <https://stackabuse.com/guide-to-sending-http-requests-in-python-with-urllib3/>
15. logging — Logging facility for Python [Электронный ресурс] – URL: <https://docs.python.org/3/library/logging.html>
16. asyncio — Asynchronous I/O [Электронный ресурс] – URL: <https://docs.python.org/3/library/asyncio.html>
17. Requests в Python – Примеры выполнения HTTP запросов [Электронный ресурс] – URL: <https://python-scripts.com/requests>
18. datetime — Basic date and time types [Электронный ресурс] – URL: <https://docs.python.org/3/library/datetime.html>
19. Python datetime module: Handling dates and time [Электронный ресурс] – URL: <https://blog.logrocket.com/python-datetime-module-handling-dates-time/#:~:text=The%20Python%20datetime%20module%20helps,import%20datetime%20print(dir(datetime))%20datetime%20class>
20. На чем пишут чат-ботов? Руководство по выбору среды разработки ботов для начинающих [Электронный ресурс] – URL: <https://vc.ru/dev/940488-na-chem-pishut-chat-botov-rukovodstvo-po-vyboru-sredy-razrabotki-botov-dlya-nachinayushchih>
21. Поразительная статистика чат-ботов 2024: преимущества, демография, тенденции, маркетинг [Электронный ресурс] – URL: <https://www.bloggersideas.com/ru/chatbot-statistics/>
22. Прогноз популярных языков программирования в 2024 году [Электронный ресурс] – URL: <https://dzen.ru/a/ZNs_tLg8_FCP4bYL>
23. Рейтинг языков программирования TIOBE за январь 2024 года [Электронный ресурс] – URL: <https://www.opennet.ru/opennews/art.shtml?num=60427>