

Министерство образования и молодёжной политики Свердловской области

Департамент образования Администрации города Екатеринбурга

Муниципальное образование «город Екатеринбург»

Чкаловский район

Муниципальное автономное общеобразовательное учреждение

средняя общеобразовательная школа № 200

с углублённым изучением отдельных предметов

Направление: социально-экономическое

Предмет: информационно-коммуникационные технологии

Проект

Создание бота–рассыльщика расписания на примере МАОУ СОШ№200

Автор проекта: Салахов Тимур Маратович, 10 класс

Научный руководитель проекта: Такмаков Игорь Васильевич

учитель Информатики

89617722853

i.takmakov@gmail.com

2022

Оглавление

Введение	
----------------	--

Теоретическая часть

Глава 1 Чат-боты	4
Глава 1.1 Применение чат-ботов и их польза	4
Глава 1.2 Боты-Рассыльщики	5
Глава 2 Парсинг	6
Глава 2.1 Что такое парсинг?	6
Глава 2.2 Возможности парсинга	6
Глава 2.3 Практическое применение парсинга.....	7
Глава 2.4 Легальность парсинга	8
Глава 3 Выбор языка программирования	9
Глава 3.1 C++.....	9
Глава 3.2 Java.....	10
Глава 3.3 Python.....	11

Практическая часть

Глава 4 Бот-Рассыльщик.....	12
Глава 4.1 Особенности и неудобства	12
Глава 4.2 Платформа Вконтакте. VKAPI.....	13
Глава 4.3 Сайт школы. Получение информации	15
Глава 5 Создание программы	16
Глава 6 Готовый продукт.....	17
Глава 7 Заключение	19
Список источников	20
Приложение	21

Введение

В наше время существует много способов доставки информации, которая, зачастую, является важной для адресата. Особенно много способов возникло в течении развития технологий. Например, раньше пользовались почтовыми голубями, затем, все писали друг другу телеграммы, а сейчас достаточно отправить другому человеку сообщение в соц. сети или мессенджере, или написать на его собственную электронную почту.

Практически все организации борются за внимание людей и желают быть популярными. Они хотят, чтобы люди знали о них и пользовались услугами этой организации. Поэтому создание бота-рассыльщика всегда является **важной** темой не только в конкретной стране, но и по всему миру.

Однако, стоит спуститься до более «земных» проблем: обучение. Каждый человек учился в школе, а затем – в ВУЗе. В этих **учебных заведениях** всегда присутствовала **проблема**: расписание уроков\лекций. В школах, после того как ученик перейдёт в среднюю школу, он столкнётся с **неустойчивым расписанием**. То учитель заболит, то будет занят как-то иначе, а ученик не будет знать, что урока не будет и расписание изменится. В ВУЗах проблема похожая: кому-то нужно на одни лекции, кому-то – на другие. В итоге, всё сводится к тому, что каждому учащемуся **необходимо** знать расписание уроков\лекций на следующий день. Поэтому данная проблема является **актуальной**, и её нужно решать в **кратчайшие сроки**.

Объект исследования: Методы, для создания удобного бота-рассыльщика

Предмет исследования: Бот - рассыльщик

Цель: узнать, что такое бот-рассыльщик, какие функции нужно реализовать для его создания, а также создать бота-рассыльщика самостоятельно.

Задачи: найти информацию о ботах-рассыльщиках, где и для чего они могут быть использованы (теоретическая часть). Узнать о необходимом функционале бота (теоретическая часть). Придумать и создать бота-рассыльщика (практическая часть).

Теоретическая часть

Глава 1 Чат-боты

Данный проект заключается в том, чтобы создать чат-бота, который будет иметь минимальный необходимый функционал. Поэтому стоит узнать побольше о самих чат-ботах и какими они бывают.

Глава 1.1 Применение чат-ботов и их польза

Для начала стоит разобраться с понятием «чат-бот». Чат-бот – это программа, которая имитирует реальный разговор с пользователем. Сама программа работает автономно и, в зависимости от надобности, собирает информацию о пользователе, чтобы подстроиться под него.

Многие организации и компании на своём официальном сайте выбирают одно из двух: вкладка Q&A (вопросы и ответы) или чат-бот. Сама вкладка Q&A может быть достаточно информативна, так как в неё помещают самые часто задаваемые вопросы, и, собственно, ответы на эти вопросы. Однако, не все эти вопросы могут быть полезны пользователям, а ответы на все вопросы могут составлять несколько страниц, и то это не гарантирует того, что пользователь найдёт нужный ему ответ.

Чат-бот же может постепенно в течение «диалога» помочь пользователю с решением некоторой проблемы или нахождением нужной информации. Однако, этим чат-боты не ограничиваются. Чат-бот может иметь свой дополнительный функционал, например, регистрация на сайте или прямое обращение в тех. поддержку (иногда это оказывается более удобным вариантом, чем прямой звонок).

Стоит признать, чат-боты – не такая широкая тема для описания, однако и в ней есть свои частные случаи.

Глава 1.2 Боты-рассыльщики

Боты-рассыльщики – частный случай чат-ботов. Как уже понятно из названия, эти боты рассылают некоторую информацию. Для того, чтобы начать их работу, нужна база данных пользователей и, собственно, информация, которую нужно донести до конечного адресата.

Стоит выделить самую главную вещь – разница между ботами-рассыльщиками и чат-ботами. Их разница заключается во взаимодействии с пользователями. Для того, чтобы чат-бот взаимодействовал с пользователем, нужно, чтобы пользователь сделал «первый шаг», т.е. начал диалог с чат-ботом. Но для бота-рассыльщика этого не понадобится, ведь никакой специфичной информации для рассылки не нужно. Достаточно взять адрес пользователя, который хранится в базе данных и уже можно отправлять информацию.

Также, стоит выделить главный минус ботов-рассыльщиков – спам. Так как организации важно только доставить информацию до конечного пользователя, на, допустим, электронную почту конечного пользователя приходит большое количество писем, не интересных ему. Таким образом, боты-рассыльщики порой являются раздражителем для людей, и письма, посылаемые от них, просто отправляются в категорию «спам».

Таким образом, мы узнали о том, что из себя представляет чат-бот и его функционал. Это знание позволит нам реализовать собственный автоответчик, который поможет пользователям, которые впервые сталкиваются с подобными технологиями. **Также,** мы узнали, что боты-рассыльщики не сильно отличаются от чат-ботов. Поэтому, не стоит перебарщивать с рассылкой сообщений, иначе, пользователь просто перестанет пользоваться нашей услугой.

Глава 2 Парсинг

Наш чат-бот должен будет отсылать пользователям расписание их учебного дня. Для этого нужно каким-то способом добывать информацию на сайте школы. Поэтому наша программа будет использовать парсинг.

Глава 2.1 Что такое парсинг?

Для людей, которые многого не знают о промышленном программировании, парсинг является чем-то за гранью понимания. Однако, всё гораздо проще, чем это кажется. Парсинг – жаргонный вариант словосочетания «Синтаксический анализ». Это подразумевает собой сбор информации по какой-либо тематике. Отсюда и глагол – парсить, т.е. собирать информацию. Чтобы более точно понять, что такое «парсинг», достаточно нескольких примеров.

Допустим, что вы в поиске новой работы. Вы заходите на очередную платформу, на которой размещают вакансии и упорно смотрите что работодатели от вас требуют и за какую зарплату. Далее, вы либо ищите какие-либо курсы для того, чтобы стать более компетентным для работы, которую вы выбрали, либо подаёте своё резюме. Таким образом, поиск работы и курсов будет своего рода парсингом.

Однако, оставлять подобную работу на человека будет не столь эффективным решением. На это влияет человеческий фактор – человек может заболеть, медленно работать или собирать не ту информацию. Поэтому, уже давно парсинг автоматизировали. Плюсы автоматизированного парсинга очевидны: скорость получения информации, задавание параметров для выборки информации, меньше ошибок в отчёте и многое другое.

Глава 2.2 Что можно найти с помощью парсинга?

В принципе, с помощью парсинга можно найти любую информацию, которая размещена на сайте, но чаще всего ищут:

- цены
- контакты компании
- описание товаров, их характеристик и в целом контент

Глава 2.3 Способы применения парсинга

В основном, области применения парсинга делятся на две области:

1. Анализ собственной площадки

Анализ собственной площадки нужен, чтобы проверить свой сайт на какие-либо ошибки. Парсинг помогает выявить скрытые ошибки, которые невидны обычному глазу, например, битые ссылки – ссылки, которые уже не являются активными или ведут не на тот источник (бывает, что некоторые ссылки работают, но представляются в некорректном виде).

Такое сканирование необходимо проводить регулярно. Не обязательно делать сканирование слишком часто, но не стоит затягивать с этим. Таким образом, любой пользователь точно сможет использовать всю информацию, что была расположена на сайте. А так как сайт работает исправно, то и отзывы, и оценку сайта заработать не представляется сложным.

2. Анализ конкурентов

Анализ конкурентов вашей области – одно из важнейших задач любого собственника бизнеса. Быстрый сбор информации с других площадок может помочь в оценке товаров и цен конкурентов. Поэтому, вы можете оставаться «на плаву» и быстро реагировать на резкие изменения в вашей области.

Стоит помнить, если вы знаете, что и как у конкурентов, то это не значит, что ничего не надо изменять кроме цен и товаров. Проанализировав площадки других собственников, нужно устранять собственные недочёты и ошибки, которые могла появиться с течением времени.

Также, сканирование позволяет вам составить собственную клиентскую базу, с которой вы можете подстроить свои услуги под нынешние реалии или некоторый круг интересов, что может помочь с увеличением популярности вашего сайта и бизнеса.

Не стоит перебарщивать с парсингом информации, ведь, некоторые ваши действия могут оказаться вне закона.

Глава 2.4 Легальность парсинга

Кратко говоря, всё, что размещено на сайте, можно парсить легально. В конце концов — это то, что создатель сайта выставил в публичный доступ, поэтому, сбор информации с любого открытого источника является законным. А вот, что преследуется законом:

- парсинг с целью DDOS-атаки
- сбор личных данных пользователей, которые находятся не на виду — например, в личном кабинете, указывались при регистрации и т. д.
- парсинг для воровства контента — например, перепост чужих статей под своим именем
- сбор информации, которая составляет государственную или коммерческую тайну

Подводя итог, мы узнали, что такое парсинг, и как добыть информацию так, чтобы у нас не возникло лишних проблем. С помощью этих знаний мы можем создать автоматический парсинг с любого сайта, который существует в сети Интернет, в том числе, и с сайта школы.

Глава 3 Выбор языка программирования

Мы узнали, что такое «чат-боты» и «парсинг». Теперь нужно определиться, на каком языке программирования нам стоит писать нашу программу. Нам нужно взять самые популярные языки программирования, и среди них выбрать тот, что больше всего подходит, аргументируя выбор.

Глава 3.1 C++



Логотип языка программирования C++

C++ — это статистически типизированный язык программирования. Сейчас C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. С его помощью создают операционные системы, разнообразные прикладные программы, драйверы устройств, игры и пр.

К сожалению, каким бы C++ многогранным языком программирования ни был, он больше подходит для создания алгоритмов, нежели для реализации визуальных объектов.

Глава 3.2 Java



Логотип языка программирования Java

Java - современный объектно-ориентированный язык программирования. Программа, написанная на Java, способна выполняться практически на любом компьютере, причём, за одно и то же время. Зная Java, можно создавать мощные мультимедийные приложения для любой платформы.

Java является отличным языком программирования для того, чтобы создавать различные проекты. Однако, Java имеет больше количество нюансов в ООП из-за чего иногда его попросту неудобно использовать в качестве основного языка программирования

Глава 3.3 Python



Логотип языка программирования Python

Python — скриптовый язык общего назначения. Скриптовые языки обычно используются для небольших задач, но Python никак не вписывается в эти рамки. Python является одним из самых популярных языков программирования, поэтому он имеет очень большое количество библиотек. Создатели Python также не дремлют, постепенно подтягивая некоторые методы, которые прижились множеству программистов в других языках программирования.

Python является не таким быстрым языком программирования, например, Java может оказаться в десятки раз быстрее. Однако, читаемость кода и простота языка заглаживают данную проблему. Более того, существуют библиотеки, осуществляющие ускорение работы кода в несколько раз.

Для осуществления проекта я выбрал Python, и на это есть такие причины:

- Простой синтаксис: код легко писать, читать и поддерживать
- Большая встроенная библиотека и множество дополнительных библиотек
- Кроссплатформенность и поддержка почти всех современных систем

Наконец, можно сказать, что мы определились с языком программирования, на котором будет основываться наш проект. Это позволит нам улучшать данный проект в будущем, чтобы он стал намного функциональнее и производительнее.

Глава 4 Бот-рассыльщик

Вот и настала практическая часть проекта – создание бота-рассыльщика. Но нам стоит упомянуть, что есть особые обстоятельства, которые мы не можем игнорировать и должны о них упомянуть.

Глава 4.1 Особенности и неудобства

В любой программе, которая была создана, есть свои «правила», по которым пользователь должен «играть». Примеров хватает не только в компьютере, но и в жизни. Например, учитель говорит, чтобы вы сдали заданное сочинение в письменном варианте, но никак не в печатном, хотя этот формат более удобен и читабелен. Как бы вы ни пытались, вам приходится сдаваться пытаться уговорить учителя принять сочинение в печатном виде и принимаете «правила игры».

Таким образом, программа тоже не сможет автоматически подстроиться под все варианты оформления входных данных. Поэтому стоит упомянуть особенности и неудобства нашей программы.

Особенности и неудобства:

- Сайт учебного заведения – данный проект берёт расписание учебного дня с сайта школы. Не каждая школа имеет расписание на сайте, поэтому программу нужно будет дописывать в зависимости от УЗ. Также, существует вероятность, что сайт УЗ будет временно недоступен.
- Интернет – зачастую, в каждом УЗ существует блокировка интернет-ресурсов. Из-за этого возможны сбои в работе системы.
- Расписание – если у УЗ и есть расписание на сайте, то оформление расписание вряд-ли будет таким-же, поэтому нужно будет либо создавать «правила», либо каким-либо образом подстраивать программу под расписание.
- Платформа – ученики каждого УЗ пользуются той соц. сетью или мессенджером, который им удобен. Однако и УЗ может иметь свою группу в соц. сети или канал в мессенджере, что может быть весомой причиной выбора той или иной платформы. Поэтому выбор платформы, на котором бот будет создан, зависит не только от учеников, но и от самого УЗ.

Глава 4.2 Платформа Вконтакте. VKAPI

Платформа, а в нашем случае, компьютерная платформа – это среда выполнения, в которой должен выполняться фрагмент программного обеспечения или объектный модуль с учётом накладываемых этой средой ограничений и предоставляемых возможностей.

Выбор платформы бота – одна из важных решений практической части. От неё зависит польза и значимость бота. Если выбрать неверную платформу, то услугами бота-рассыльщика будут пользоваться немногие, а, следовательно, проблема не будет решена.

Важно выбрать платформу для закрепления бота перед тем, как начинать создавать программу. От платформы зависит сама структура проекта. Каждая платформа по-разному поддерживает создание ботов. Для одной платформы нужно одни методы, для другой – другие. Поэтому, самый трудный, и в то же время важный шаг для проекта – выбор социальной сети или мессенджера, в качестве платформы для закрепления бота.

К счастью, выбирать пришлось недолго – школа уже имела своё сообщество в социальной сети Вконтакте. Вконтакте является отличным вариантом платформы, т.к. это самая популярная соц. сеть в России. Более того, во Вконтакте уже существует колоссальное количество ботов, подобных тому, что я собираюсь создать. Значит, создание своего бота не будет представляться такой трудной задачей.

Выбрав Python в качестве основного языка программирования, мы не ошиблись, так как в нём была реализована библиотека «vkapi», которая взаимодействует с API Вконтакте.

Но что же такое API? API (англ. Application Programming Interface, рус. программный интерфейс приложения) – инструмент, с помощью которой одна компьютерная программа может взаимодействовать с другой компьютерной программой.

Если проще, то если мы на доске нарисует что-нибудь мелом, то вы и доска будете компьютерными программами, а мел – API. Т.е. вы, с помощью некоторого инструмента оставили информацию на (в) некотором предмете. Так и программы могут взаимодействовать друг с другом с помощью API.

Теперь остаётся только понять, как работать с API Вконтакте. Для этого у Вконтакте есть документация для разработчиков, размещённая в специальном разделе.

Посмотрев на документацию, мы натываемся на список API-сервисов:

- Bots Long Poll API
- Callback API
- User Long Poll API
- Streaming API

Из всех этих сервисов нас удовлетворяют первые два — они взаимодействуют конкретно с сообществом Вконтакте. Теперь нам нужно узнать различие между ними. Документация говорит нам: «**Bots Long Poll API** позволяет работать с событиями из Вашего сообщества в режиме реального времени. В отличие от **Callback API**, очередь из событий хранится на стороне Вконтакте — мы не будем присылать отдельное уведомление для каждого события. В противоположность **User Long Poll**, Bots Long Poll API работает с событиями сообщества, а не пользователя.»

Из этого длинного текста мы можем понять:

1. Callback API и Bots Long Poll API работают **только** с сообществом
2. Если мы будем использовать **Bot Long Poll API**, то Вконтакте возьмёт на себя очередь событий и будет отправлять нам только конкретные уведомления, т.е. нагрузка на бота будет меньше

Таким образом, мы понимаем, что лучшая опция для нас — Bots Long Poll API.

Глава 4.3 Сайт школы. Получение информации

У нас есть база данных и взаимодействие с Вконтакте, чего же нам не хватает? А не хватает нам той информации, которую мы будем рассылать — расписание. Поэтому стоит рассмотреть способы получения расписания.

Как бы то ни было, все возможности получения информации с сайта сводятся к двум опциям:

1. Получение информации через API сайта
2. Получение информации через запрос по ссылке

Первый вариант очень полезный, но только если он есть. Некоторые сайты не имеют собственного API, т.к. не имеют смысла в нём или потому что владелец сайта не задумывался об этом. Так или иначе, некоторые сайты могут и не иметь собственного API. К тому же API у каждого сайта свой, поэтому в зависимости от сайта, программу нужно будет переписывать, что не очень удобно.

Таким образом, стоит посмотреть на второй вариант. Получение информации через запрос по ссылке является не самым удобным, но самым действенным способом получения информации. После перехода по ссылке мы либо попадем на страницу сайта, которую можем проанализировать; либо скачиваем файл, после чего мы также можем что-то с ним сделать.

В нашем случае стоит реализовать второй вариант, так как у сайта школы нету своего API, и нам, получается, не из чего выбирать. В Python есть встроенная библиотека «requests», которая взаимодействует с сайтами и может посылать HTTP-запросы:

- GET — получение ресурса
- POST — создание ресурса
- PUT — обновление ресурса
- DELETE — удаление ресурса

С помощью GET-запроса мы можем получить множество информации, в том числе и сам файл, в случае если ссылка ведёт на скачивание файла. Получается, с помощью GET-запроса мы можем получить расписание уроков, а после, обработать и отправить его.

Глава 5 Создание продукта

Ввиду технических обстоятельств, данный проект не может быть реализован в одной программе. Поэтому, наш бот-рассыльщик расписания будет состоять из двух частей:

1. Автоответчик
2. Таймер

Автоответчик представляет собой ту часть, которая будет отвечать на сообщения, которые пользователь отправил сообществу. Он будет записывать желающих пользователей в общую базу данных, в которой будут храниться пользователи и класс, на рассылку которого они подписались. Таким образом, автоответчик – это визуальная часть программы.

Таймер имеет за собой большой функционал. Таймер, с указанной задержкой, отправляет запрос на сайт школы, тем самым, скачивая файлы с расписаниями учеников. После чего обрабатывает эти данные, что является довольно сложной задачей для программы. А затем, сопоставляет расписания классов с данными, что уже хранятся в той же базе данных, что и список пользователей. В случае, если полученные данные и данные базы данных разнятся, то программа записывает полученные данные в базу, а пользователю пишет, что расписание обновилось.

Таким образом, программа в совокупности может не только мгновенно отвечать на сообщения множества пользователей, но и поддерживать их в курсе изменений в расписании.

Немного фактов:

1. Практическая часть проекта состоит из пяти Python-файлов и одной базы данных (+ 2 Excel-файла).
2. Изначально, проект имел визуальную оболочку, в качестве интерфейса с системой управления. Это подразумевало то, что программа будет установлена на один из компьютеров, которые находятся в учебном заведении и человек, ответственный за программу, сможет комфортно управлять ботом и его функционалом. Однако из-за недоступности социальных сетей в школьной сети, был удалён.

Глава 6. Готовый продукт

После того, как мы разобрали, с помощью чего будет создано наше приложение, откуда мы будем брать информацию и как будет работать наше приложение в целом, остаётся только создать конечный продукт.

Для того, чтобы комфортно разрабатывать новые проекты, разработчики пользуются различными IDE (англ. Integrated Development Environment, рус. Интегрированная Среда Разработки) — комплекс программных средств, используемый программистами для разработки программного обеспечения. Для Python-разработчиков существует популярное решение от компании JetBrains — PyCharm, который я использовал для создания данной программы. Поэтому, структура программы будет показана с помощью этого IDE.

Структура проекта:

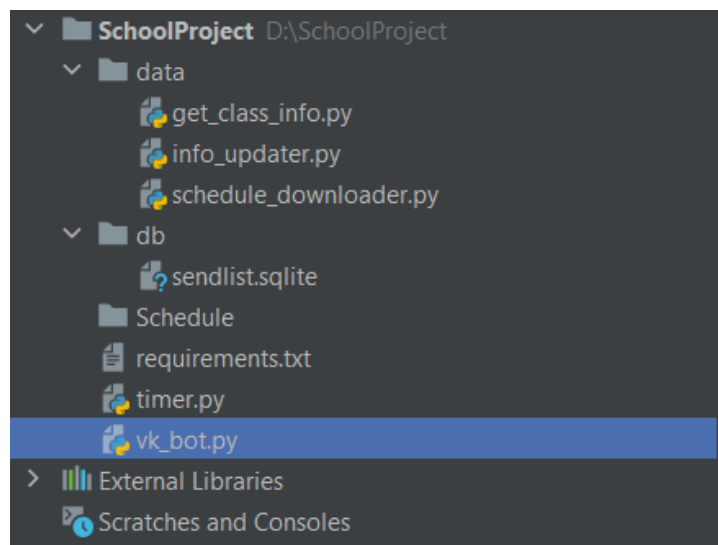
Посмотрев на структуру готового продукта, мы можем увидеть, что наше приложение состоит из пяти Python-файлов и одного файла базы данных (sendlist.sqlite). Поподробнее разберём, какой файл за что отвечает.

Из главы 5 мы можем понять, что главными файлами проекта будут являться автоответчик, взаимодействующий с ВКонтакте (vk_bot.py) и таймер обновления расписания (timer.py). Из главы 5 мы понимаем, что эти два Python-файла запускаются отдельно. Из этих двух программ стоит начать с меньшего — автоответчика.

Автоответчик

Vk_bot.py с помощью API

ВКонтакте принимает сообщения, которые присылаются пользователями в группу ВКонтакте, за которым он закреплён. В зависимости от сообщения, бот добавляет пользователя в базу данных, т.е. добавляет пользователя в лист пользователей, подписавшихся на рассылку или удаляет пользователя из базы данных. Общий объём кода составляет 350 строк.



Структура проекта в PyCharm IDE

Таймер

Timer.py уже имеет большой функционал, так как имеет относительно большой объем кода (183 строки) и получает информацию из других файлов, находящихся в папке «data». Пройдёмся по порядку. Сначала таймер проверяет серверное время (время, которое установлено на сервере) и, если на данный момент, **время** находится в промежутке от **8 часов утра до 17 часов вечера**, то программа замораживается, т.к. это учебное время и появления нового расписания не предвидится (также, чтобы зря не давать лишней нагрузки на сервер). **Иначе**, таймер даёт сигнал файлам из папки «data» по очереди: timer.py -> info_updater.py -> get_class_info.py -> schedule_downloader.py -> get_class_info.py -> info_updater.py -> timer.py

После получения обратного сигнала таймер отправляет пользователям, находящимся в базе данных, их расписание, если их классное расписание изменилось.

Стоит также кратко описать функционал файлов, на которые ссылается таймер:

- info_updater.py – ссылается на get_class_info.py. Сравнивает полученное расписание с тем, что находится в базе данных. Если расписание отличается, то изменяет его в базе данных и добавляет класс в список классов с изменённым расписанием, который отправляется обратно в timer.py. Общий объем кода – 59 строк.
- get_class_info.py – ссылается на schedule_downloader.py. Обработывает excel-файл с расписанием на предстоящий учебный день. Отправляет список с расписанием всех классов, находящихся в данном excel-файле обратно в info_updater.py. Общий объем кода – 116 строк.
- Schedule_downloader.py – скачивает excel-файл с сайта школы, с расписанием школьных занятий в папку «Schedule». Отправляет названия excel-файлов с их местонахождением обратно в get_class_info.py. Общий объем кода – 42 строки.

Таким образом, мы понимаем, как работает наш конечный продукт, а также можем составить схему взаимодействий наших файлов внутри (см. Приложение).

Глава 7. Заключение

В течение создания данного проекта мы преследовали единственную цель – облегчить способ получения информации. В ходе выполнения данного проекта были сделаны следующие выводы:

1. Чат-боты и боты рассылщики тесно связаны между собой, один отвечает на входящие сообщения, другой – просто отправляет сообщения, в любой его форме
2. Не следует злоупотреблять рассылкой, ведь тогда пользователи перестанут пользоваться вашими услугами
3. Парсинг – это метод сбора информации, которым мы и сами достаточно часто пользуемся
4. За парсинг можно понести за собой уголовную ответственность, однако, если пользоваться только теми сайтами, которые видны в сети Интернет и не злоупотреблять парсингом, то ничего плохого с вами не случится
5. У данного проекта, несомненно, есть будущее, которое поможет ему развиваться в различных направлениях

Список источников

1. «Python на примерах. Практический курс по программированию» - Алексей Васильев. 2018 год
2. «Python. Подробный справочник» - Дэвид Бизли
3. <https://blog.ringostat.com/>
4. <https://pandas.pydata.org/>

Приложение

The screenshot shows the Antiplagiat website interface. At the top, there is a header with the Antiplagiat logo, a tariff section (Free), a modules and collections section (1 connected), a balance section (0), and a user section (claymix007@gmail.com). Below the header, there is a navigation bar with 'Главная / Кабинет'. The main content area is titled 'Кабинет' and contains a search bar, a list of actions (Переместить, Удалить, История отчетов), and a table of documents. The table has columns for 'Название', 'Дата загрузки', and 'Оригинальность'. The first document is 'Салахов 01.12' with a date of '02 Дек 2021 21:50' and an originality of '98,48%'. The second document is 'Салахов 23.10' with a date of '23 Ноя 2021 16:15' and an originality of '84,16%'. Both documents have a 'ПОСМОТРЕТЬ РЕЗУЛЬТАТЫ' button next to them.

Название	Дата загрузки	Оригинальность
Салахов 01.12	02 Дек 2021 21:50	98,48%
Салахов 23.10	23 Ноя 2021 16:15	84,16%

Рисунок 1. Проверка на Антиплагиат теоретической части. 98,48%

The screenshot shows the Antiplagiat website interface. At the top, there is a header with the Antiplagiat logo, a tariff section (Free), a modules and collections section (1 connected), a balance section (0), and a user section (claymix007@gmail.com). Below the header, there is a navigation bar with 'Главная / Кабинет'. The main content area is titled 'Кабинет' and contains a search bar, a list of actions (Переместить, Удалить, История отчетов), and a table of documents. The table has columns for 'Название', 'Дата загрузки', and 'Оригинальность'. The first document is 'Салахов 21.12' with a date of '21 Дек 2021 04:47' and an originality of '99,06%'. The second document is 'Салахов 01.12' with a date of '02 Дек 2021 21:50' and an originality of '98,48%'. The third document is 'Салахов 23.10' with a date of '23 Ноя 2021 16:15' and an originality of '84,16%'. All documents have a 'ПОСМОТРЕТЬ РЕЗУЛЬТАТЫ' button next to them.

Название	Дата загрузки	Оригинальность
Салахов 21.12	21 Дек 2021 04:47	99,06%
Салахов 01.12	02 Дек 2021 21:50	98,48%
Салахов 23.10	23 Ноя 2021 16:15	84,16%

Рисунок 2. Проверка на Антиплагиат проекта полностью. 99,06%

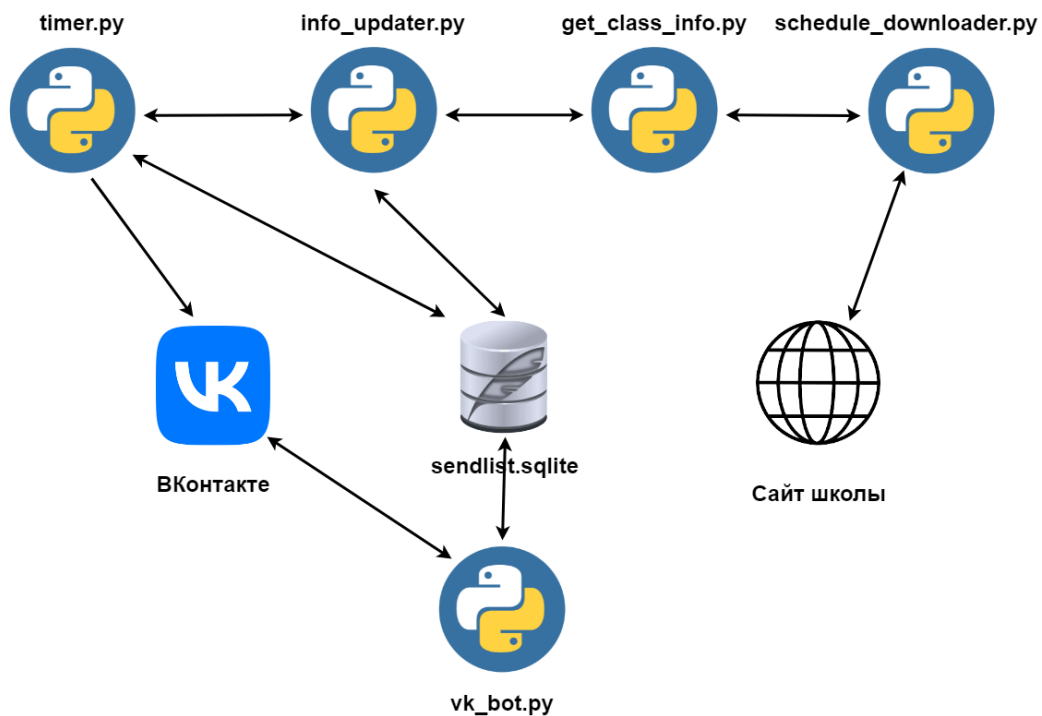


Рисунок 3. Блоксхема взаимодействий файлов проекта

```

138 def timer():
139     """Actual timer of schedule updating. Uses:
140     vk_bot.get_community_info()
141     timer_status()
142     data.info_updater.refresh_lessons()
143     to_seconds()
144     get_delay()
145     """
146
147     com_info = get_community_info()
148
149     try:
150         vk_session = vk_api.VkApi(token=com_info['token'])
151         longpoll = VkBotLongPoll(vk_session, com_info['community_id'])
152
153         vk = vk_session.get_api()
154     except Exception as e:
155         print(e)
156         return
157
158     while True:
159         now = datetime.datetime.now()
160         if timer_status() and (17 <= now.hour < 24 or 0 <= now.hour < 8):
161             classes_to_send = refresh_lessons()
162             print(classes_to_send)
163
164             if classes_to_send:
165                 for i in classes_to_send:
166                     for j in people_to_send(i):
167                         vk.messages.send(user_id=j, message=update_text(i),
168                                         # Don't really know what I should put here, so I put there a random num
169                                         random_id=random.randint(0, 2 ** 64))
170
171             # Means that current time between 8:00 and 17:00
172             # 17:00 is time when usually schedule uploads
173         else:
174             cur_time = to_seconds(hours=now.hour, minutes=now.minute, seconds=now.second)
175             delta = to_seconds(hours=17) - cur_time
176

```

Рисунок 4. Пример рабочего кода (Таймер)

```

51 def schedule_getter(sheet) -> list:
52     sheet = sheet_nan_remover(sheet)
53
54     classes = []
55     classnames_flag = True
56     tmp = []
57
58     for i in sheet:
59         line = i[2:]
60
61         if set(line) == {'-'} and i[1] != i[1]:...
62
63         elif classnames_flag:...
64
65         else:...
66
67     if tmp:
68         classes.extend(tmp)
69
70     return classes
71
72
73 def get_lessons() -> list:
74     request_datetime = get_day_to_check()
75
76     request_url = get_url()
77     result = download(request_datetime, request_url)
78
79     if result:
80         classes_1to4 = result[0]
81         classes_5to11 = result[1]
82
83         excel_1to4 = pd.ExcelFile(classes_1to4)
84         excel_5to11 = pd.ExcelFile(classes_5to11)
85
86         # Parsing from 1 to 4 grades
87         sheet = excel_1to4.parse(excel_1to4.sheet_names[0])
88         sheet = sheet.values[9:52]
89
90

```

Рисунок 5. Пример рабочего кода (Обработчик расписания)

```

97 def main():
98     """
99
100     com_info = get_community_info()
101
102     try:
103         vk_session = vk_api.VkApi(token=com_info['token'])
104         longpoll = VkBotLongPoll(vk_session, com_info['community_id'])
105     except Exception as e:
106         print(e)
107         return
108
109     print('VkBot is ready to work!') # Print line to know if bot is ready
110
111     for event in longpoll.listen():
112         if event.type == VkBotEventType.MESSAGE_NEW:
113             reply = ''
114             console_reply = None
115             user_id = event.obj.message['from_id']
116             text = event.obj.message['text'].split()
117
118             if len(text) == 2 and text[0] == add_person_dict['command']:...
119
120             elif text[0] == remove_person_dict['command']:...
121
122             elif len(text) == 1 and text[0] == command_help_dict['command']:...
123
124             elif len(text) == 4 and text[0] == add_class_dict['command']:...
125
126             elif len(text) == 3 and text[0] == remove_class_dict['command']:...
127             else:
128                 reply = 'Я не знаю, какой ответ дать'
129
130             print(f'{event.obj.message["from_id"]} {text[console_reply]}') # Print lines to get info about bot activity
131             vk = vk_session.get_api()
132             vk.messages.send(user_id=user_id, message=reply,
133                             # Don't really know, what I should put here, so I put there a random num
134                             random_id=random.randint(0, 2 ** 64))
135

```

Рисунок 6. Пример рабочего кода (Автоответчик)