**муниципальное автономное общеобразовательное учреждение**

**города Новосибирска «Гимназия № 14 – образовательный центр «Универсарий»**





**Проект**

**тема: «Создание телеграм-бота на языке Python»**

Автор: Шарипов Руслан Одилович

Руководитель проекта: Носкова Екатерина Юрьевна

Начало работы: сентябрь 2023 г.

Завершение работы: апрель 2024 г.

Новосибирск

2024 год

**Содержание**

Введение 3

1. Язык и среда программирования 4
2. Выбор библиотеки 4
3. Создание бота через @BotFather 7
4. Написание бота 7
5. Запуск бота 9

Заключение 10

Список литературы 11

Приложения 12

Введение

В современном мире бумажные открытки стали пользоваться гораздо меньшей популярностью, а электронные открытки можно все чаще получить от своих близких. Чтобы сделать открытку не такой скучной, я решил создать Телеграм-бота, который будет создавать уникальные подписанные открытки.

Бот Telegram находится в самом Телеграме и является особым пользователем, поведение которого контролирует программа. Такая программа чаще всего пишется на языке программирования Python. Telegram – крайне популярный мессенджер, поэтому использование этого бота будет крайне удобным, так как интерфейс на основе чата интуитивно понятен и многие пользователи с ним знакомы.

**Цель проекта:** создать Телеграм-бота для создания именных открыток на языке Python.

**Задачи:**

**1.** Рассмотреть различные библиотеки Python для создания бота.

**2.** Выбрать библиотеку.

**3.** Создать бота.

1.Язык и среда программирования

Python — это высокоуровневый язык программирования, отличающийся эффективностью, простотой и универсальностью использования. Он широко применяется в разработке веб-приложений и прикладного программного обеспечения, а также в машинном обучении и обработке больших данных. За счет простого и интуитивно понятного синтаксиса является одним из распространенных языков для обучения программированию. Телеграм-бот пишется на Python.

Интегрированная среда разработки (программирования) — комплекс программных средств, используемый программистами для разработки программного обеспечения. Среда разработки включает в себя: текстовый редактор, Транслятор, средства автоматизации сборки, отладчик.

Одной из самых популярных сред разработки на Python является PyCharm, разработанный компанией JetBrains.

PyCharm — это среда программирования для языка Python, или IDE. Это мощный многофункциональный инструмент для разработчиков. PyCharm пользуются более 50% разработчиков на Python, так как он удобный и функциональный.

2.Выбор библиотеки

Чтоб написать Телеграм-бота необходимо использовать специальные библиотеки.

Библиотеки Python — это файлы с шаблонами кода. Их создали для того, чтобы люди не набирали каждый раз заново один и тот же код: достаточно открыть файл, вставить свои данные и получить результат.

Рассмотрим 3 основных варианта.

2.1. python-telegram-bot

Python-telegram-bot - это библиотека Python для создания Телеграм-ботов, разрабатываемая сообществом. Она предоставляет обширный набор функций и простой интерфейс для взаимодействия с Telegram Bot API. Эта библиотека активно поддерживается и обновляется сообществом разработчиков.

Плюсы:

* Полная функциональность: обладает обширным набором инструментов для обработки различных типов сообщений и медиа-файлов.
* Интеграция: легко интегрируется с внешними сервисами и ресурсами.

2.2. aiogram

aiogram – это высокопроизводительная и интуитивно понятная библиотека для создания Телеграм-ботов на Python. Она была разработана с учетом активного участия сообщества. aiogram предоставляет множество инструментов и функций, что делает ее идеальным выбором для создания ботов с расширенной функциональностью. Она активно поддерживается и обновляется сообществом разработчиков, и это гарантирует актуальность и надежность библиотеки.

Плюсы:

* Полная функциональность: предоставляет богатый набор функций для обработки различных типов сообщений и событий.
* Интеграция: эффективно интегрируется с внешними ресурсами и веб-сервисами.

2.3. Telebot

Telebot – это популярная библиотека Python для создания Телеграм-ботов. Она была разработана для простоты и удобства использования, и быстро завоевала популярность среди разработчиков. Эта библиотека предоставляет простой и интуитивно понятный способ создания ботов, что делает ее привлекательным выбором как для начинающих, так и для опытных разработчиков.

Плюсы:

* Основная функциональность: подходит для базовых ботов и проектов, не требующих сложной функциональности.

Минусы:

* Ограниченная интеграция: может потребовать более глубокой настройки для интеграции с внешними сервисами.

2.4. Выбор библиотеки

Я выбрал библиотеку python-telegram-bot, так как я планирую продолжать изучение темы Телеграм ботов, делать ботов сложнее, следовательно, лучше сразу приучать себя к python-telegram-bot или aiogram. Почему не aiogram? Потому, что python-telegram-bot мне показался проще. Также на дополнительных курсах по Python меня учили работать с python-telegram-bot, следовательно знания по поводу использования этой библиотеки у меня есть.

3. Создание бота через @BotFather

Перед началом работы над Телеграм-ботом необходимо составить план:

1. Создание бота через @BotFather
2. Написание кода для бота
   1. Сбор информации у пользователя
   2. Генерация открыток
   3. Отправка открыток пользователю
3. Запуск бота

BotFather(далее BF) — это главный сервис в Телеграмме, через который происходит регистрация всех пользовательских ботов. На любой аккаунт можно зарегистрировать неограниченное количество ботов, единственное условие — это уникальный юзернейм с приставкой «bot», например, «@MyFirstBot» или «@MyFirst\_bot». Что интересно, BF сам является Телеграм-ботом.

Чтобы зарегистрировать бота необходимо написать BF команду /start. BF отправляет инструкцию по пользованию со всеми командами. Чтобы создать бота нам нужно отправить команду /newbot. Далее мы придумываем боту название, в моем случае «открытки бесплатно».

Затем мы должны придумать для бота уникальный username, по которому его можно будет найти. В моем случае это «@PostcardGenerator\_bot».(см. прил. 1)

Бот создан. BF отправляет токен, имея который этим ботом можно управлять. В связи с этим фактом токен необходимо бережно хранить, чтобы злоумышленник ничего не смог сделать с ботом.

1. Написание бота

Для начала опишем структуру программы (см. прил. 2).

Папки «.idea», «.venv», и «\_\_pycache\_\_» нужны для функционирования программы и создаются автоматически. Python файл «main.py» - основной файл, который мы будем запускать. «photo\_generator.py» реализует процесс подписи открыток. В папке «data» хранится все остальное: шрифты, шаблоны для открыток. Подписанные открытки перед отправкой также сохраняются в подкаталогах папки «data», затем отправляются пользователю и удаляются.

* 1. Сбор информации у пользователя

Теперь напишем непосредственно Телеграм-бота в файле main.py. В начале кода необходимо импортировать библиотеки, которые будут использоваться: os для работы с файлами, logging, telegram.ext для фунционирования Телеграм-бота.

Функция main() – основная функция, она будет вызываться при запуске программы и в ней будут вызываться остальные функции.

Добавим необходимую «базу» для бота, от которой будем отталкиваться (см. прил. 3). Добавим ConversationHandler() чтобы задать диалог между ботом и пользователем, и на каждый этап диалога отдельную функцию. В каждой функции будем сохранять ответы в context.user\_data на вопросы, заданные в прошлой функции. Далее необходимо сделать генерацию открыток на основании информации, собранной от пользователя.

* 1. Генерация открыток

Генерация открыток будет осуществляться в файле photo\_generator.py при помощи библиотеки PIL. В зависимости от праздника мы открываем открытку, добавляем на нее текст «от: \_\_\_, кому: \_\_\_» и сохраняем под особым названием. Теперь в файле main.py в функции photo\_generate() Мы вызываем функцию из файла photo\_generator.py. Открытки сгенерированы и сохранены, программе остается их отправить пользователю и удалить.

* 1. Отправка открытки

Отправка осуществляется в функции photo\_generate() при помощи update.message.reply\_media\_group(). После этого программа удаляет их с нашего устройства. Бот Готов. Осталось лишь его запустить

1. Запуск бота

Бот запускается одним нажатием мыши по кнопке «Run» и не выключается до тех пор, пока устройство не выключить.

Заключение

В ходе проекта я научился, как создавать Телеграм-ботов, создал своими руками бота, который будет приносить пользу. Выявил для себя лучшую библиотеку для создания Телеграм-ботов: python-telegram-bot.

Ссылку на GitHub я оставлю в приложении 4.

Список литературы

<https://docs.radist.online/radist.online-docs/nashi-produkty/radist-web/podklyucheniya/telegram-bot/instrukciya-po-sozdaniyu-i-nastroiki-bota-v-botfather>

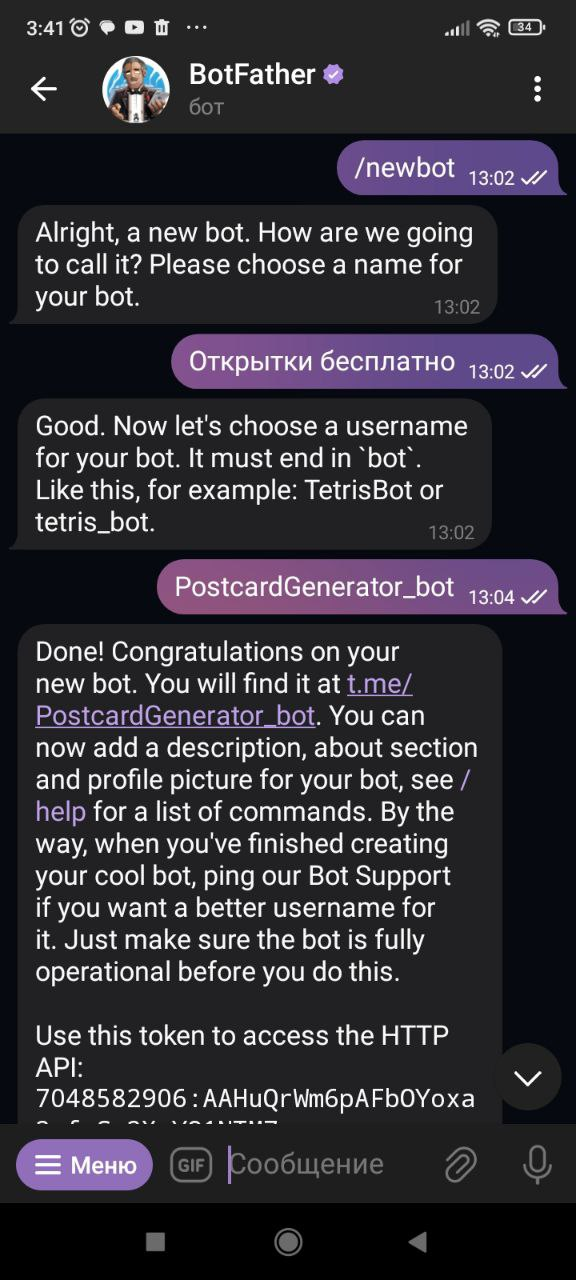
<https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8>

<https://blog.skillfactory.ru/glossary/pycharm/>

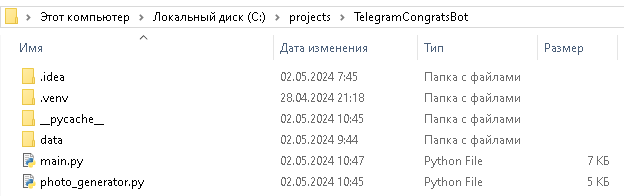
<https://habr.com/ru/companies/otus/articles/771110/>

<https://lms.yandex.ru/courses/1054/groups/8792/lessons/5981>

Приложение 1



Приложение 2



Приложение 3

import logging  
import os  
from telegram.ext import Application, MessageHandler, filters, \  
 CommandHandler, ConversationHandler  
from telegram import ReplyKeyboardMarkup, Update, InputMediaPhoto  
import photo\_generator  
  
  
logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.DEBUG)  
logger = logging.getLogger(\_\_name\_\_)  
  
  
def main():  
 application = Application.builder().token('7048582906:AAHuQrWm6pAFbOYoxa2zfwSc2XgYO1NTM7w').build()  
 application.run\_polling()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Приложение 4

https://github.com/theDarkSideOfRuslan/Celebration\_Telegram\_Bot/tree/master