**++++Департамент образования мэрии города Новосибирска Дворец творчества детей и учащиеся молодежи «Юниор»**

**Городской конкурс проектов учащихся 5 – 8 классов**

Направление: инженерно-технологическое

**Пакетный менеджер на языке программирования Python**

Автор проекта:

Шарипов Руслан

МАОУ Гимназия № 14 – образовательный центр «Универсарий»

9 класс

Консультант проекта:

Носкова Екатерина Юрьевна

учитель информатики

первой квалификационной категории

Контактный телефон руководителя:

+79537650310

г. Новосибирск, 2024

**Введение**

Бывают такие ситуации, когда нет возможности подарить Настоящую открытку близкому человеку. Поэтому я решил создать telegram бота для того, чтобы создавать цифровые открытки, которые никто никому еще не дарил.

Актуальность:

**Цель:**

- создание telegram бота на языке Python

**Задачи:**

- рассмотреть различные варианты, как это можно сделать;

-определиться с методом;

-написать телеграм бота.

**Python, Telegram боты**

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 — это мессенджер (программа для обмена сообщениями), реализованный по клиент-серверной архитектуре. Используя сервер для создания диалога между двумя клиентами, Телеграм пересылает через него или напрямую текстовые сообщения, а также изображения, видео или документы других форматов. Что же такое телеграм-бот?

Представьте, что у нас есть зарегистрированный аккаунт. Все сообщения, присылаемые на него, попадают на вход некоторой программе, а выход этой программы отправляется в виде сообщения тому, от кого пришло исходное сообщение. Это и есть бот. Иными словами, Телеграм-бот — это специальный пользователь, поведением которого управляет некоторая программа. Технически для сервера нет разницы, является данный пользователь человеком или ботом: для сервера оба клиента выглядят одинаково. Эта идеология очень похожа на пользователей-сообществ в vk.com.

Писать код я буду в среде программирования PyCharm.

PyCharm — это среда программирования для языка Python, или IDE. Средами называют программы, в которых можно писать, запускать и отлаживать код, устанавливать новые расширения и дополнительные модули. Это мощный многофункциональный инструмент для разработчиков. PyCharm пользуются более 50% разработчиков на Python, так как он удобный и функциональный.

**Создание и реализация телеграм-бота**

1. Создание бота через @BotFather
2. Выбор библиотеки
3. Написание бота
   1. Сбор информации у пользователя
   2. Генерация открытки
   3. Отправка открытки пользователю
   4. Оформление бота
4. Запуск бота

**1.Создание бота через @BotFather**

@BotFather(далее БФ) – Телеграм-бот, существующий для создания и оформления Телеграм-ботов.

Пишем БФ /start, Он 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».

Бот создан. БФ отправляет токен, имея который этим ботом можно управлять.

**2.Выбор библиотеки**

Основные библиотеки для создания телеграм-бота:

1.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.

2.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.

3.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.

Сравним эти 3 библиотеки.

1.python-telegram-bot:

Полная функциональность: обладает обширным набором инструментов для обработки различных типов сообщений и медиа-файлов.

Интеграция: легко интегрируется с внешними сервисами и ресурсами.

2.aiogram:

Полная функциональность: предоставляет богатый набор функций для обработки различных типов сообщений и событий.

Интеграция: эффективно интегрируется с внешними ресурсами и веб-сервисами.

3.Telebot:

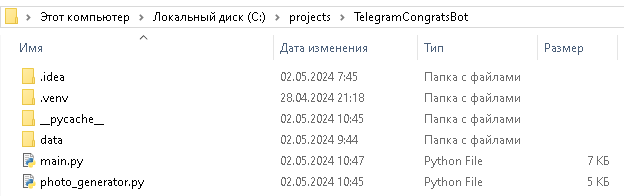
Основная функциональность: подходит для базовых ботов и проектов, не требующих сложной функциональности.

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

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.

**3.Написание бота**

Для начала опишем основную структуру программы.



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.

Теперь напишем необходимую основу для бота в файле main.py:

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\_\_)

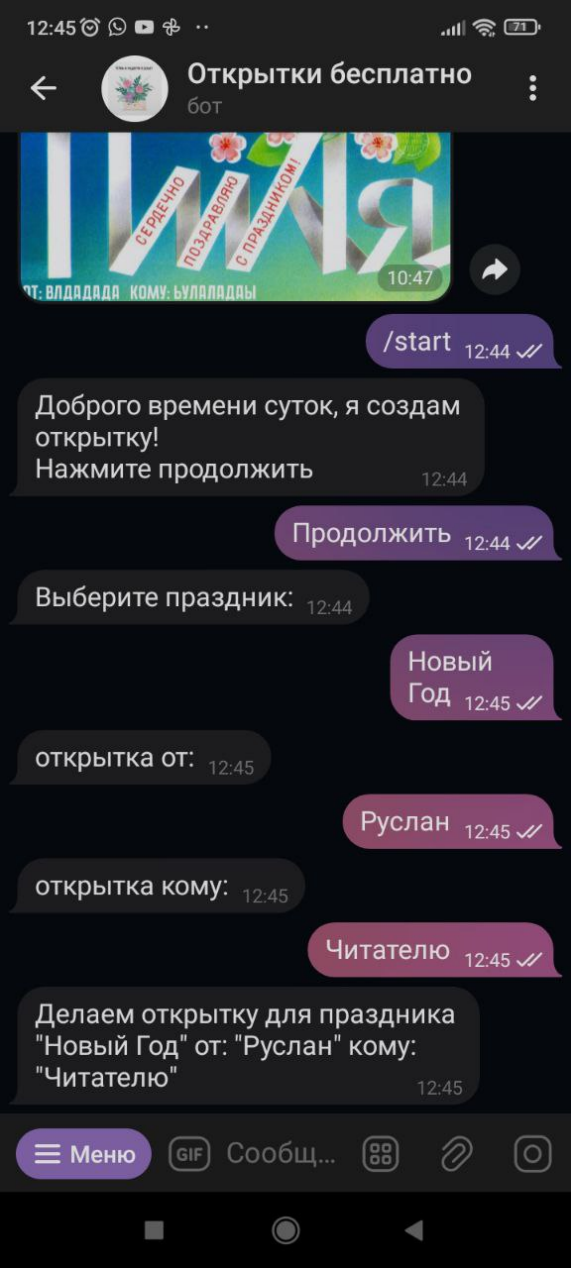
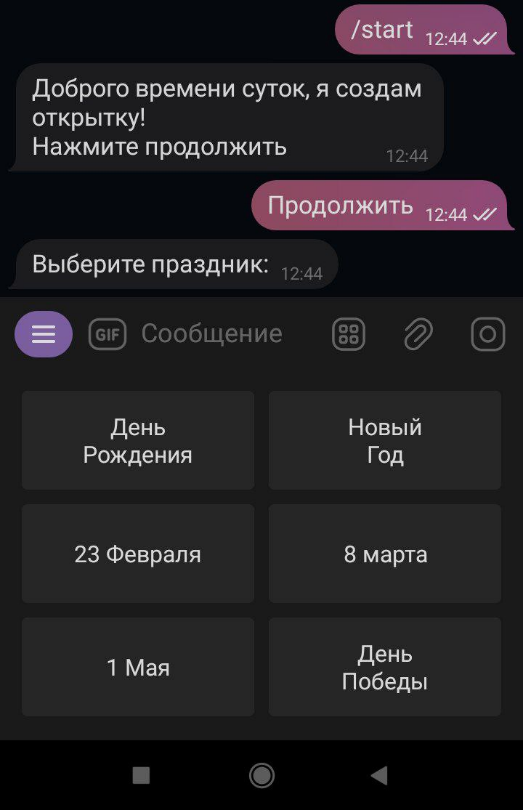
def main():  
 application = Application.builder().token('7048582906:AAHuQrWm6pAFbOYoxa2zfwSc2XgYO1NTM7w').build()  
 application.run\_polling()

if \_\_name\_\_ == '\_\_main\_\_':  
 main()

'7048582906:AAHuQrWm6pAFbOYoxa2zfwSc2XgYO1NTM7w' в коде – токен бота, который мне отправил БФ

3.1.Сбор информации у пользователя

Добавим CommandHandler для команд start и help. Добавим ConversationHandler для создания диалога между ботом и пользователем, зададим параметры states, указывающие, какие сообщения должны идти после каких. Пропишем каждый из этапов диалога в функциях async def, и в возвращать функция будет номер следующего сообщения. Сохранять ответы бот будет в context.user\_data. Теперь при введении команды /start начитается следующий диалог:



3.2.Генерация открытки

Откроем файл photo\_generator.py и пропишем по функции на каждый праздник, которые на вход будут принимать имя отправителя, имя получателя и id чата для идентификации. Здесь при помощи библиотеки PIL мы можем открыть открытку, добавить на нее текст с отправителем и получателем, сохранить и закрыть файл.

3.3.Отправка открытки пользователю

В файле main.py импортируем файл photo\_generator в функции photo\_generate() присвоим переменным day, by, to, id данные, которые вводил пользователь и id чата. Context.user\_data нужно очистить. Теперь отфильтруем правильно введенные дни. На каждый праздник создаем проверку и в зависимости от праздника вызываем функцию из photo\_generator.py. Открытки сгенерированы, отправляем их при помощи update.message.reply\_media\_group() и удаляем из каталога созданные открытки.

3.4.Оформление бота

Заходим на профиль бота и, так как мы создатели, меняем фото бота, описание. В БФ добавляем список команд /start и /help.

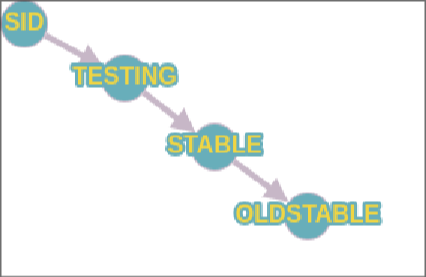
4.Запуск бота

Этот пункт самый простой. В PyCharm запускаем программу(Shift+F10) и бот работает.

**Заключение**

**Я создал бота, запустил, и теперь иногда буду им пользоваться. Программу я загрузил на свой GitHub**

Приложение А

Рисунок 1. Схема веток DPKG&APT.

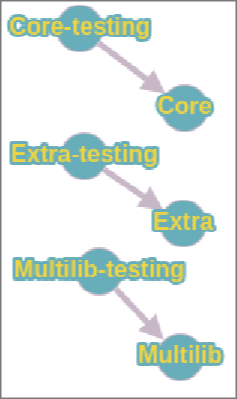
****Рисунок 2. Ветка Pacman.

Рис 3. Схема XBPS-QUERY

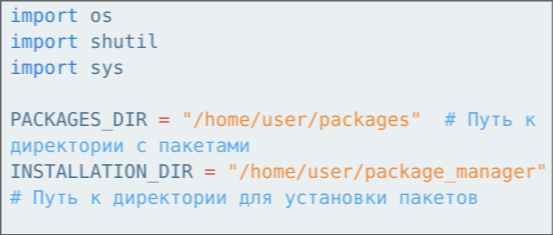
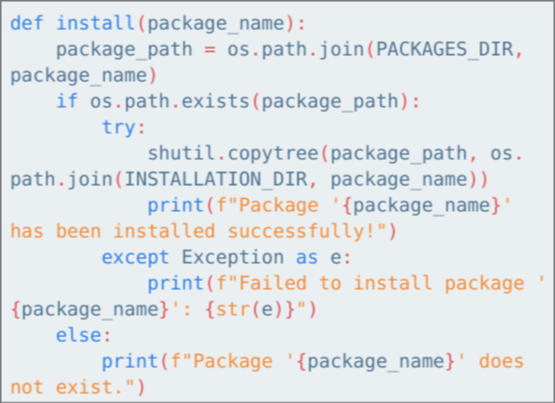
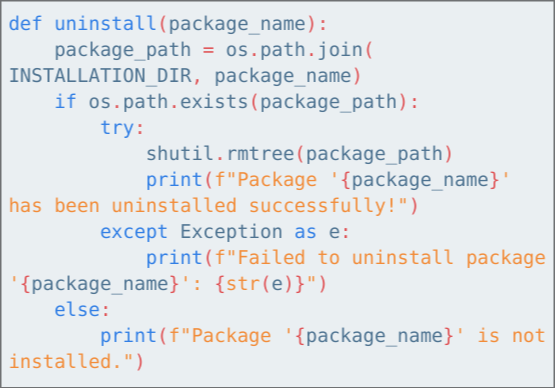
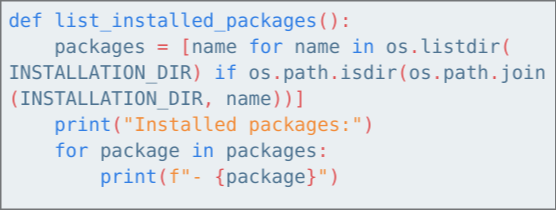
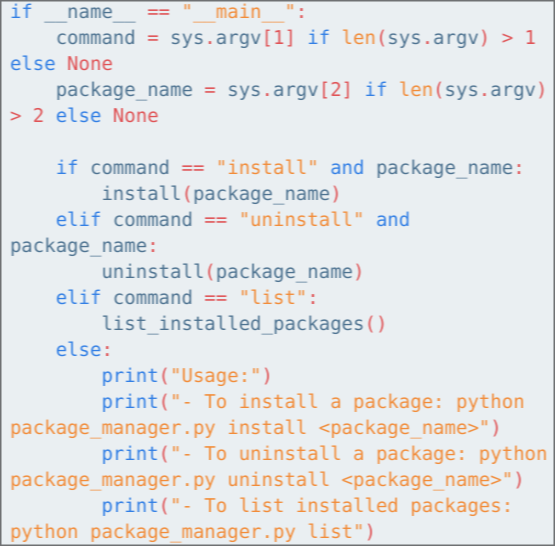
Приложение Б

Рисунок 1. Первая часть кода.

Рисунок 2. Функция установки пакетов.

Рисунок 3. Функция удаления пакетов.

Рисунок 4.Функция вывода списка установленных пакетов.

Рисунок 5.Обработка команд из командой строки.

**Список использованной литературы**

1. <https://graphonline.ru/>
2. <https://losst.pro/obzor-paketnyh-menedzherov-linux#toc-2-rpm-red-hat-package-manager>
3. <https://eternalhost.net/blog/sistemnoe-administrirovanie/paketnye-menedzhery-linux>
4. https://ru.wikibrief.org/wiki/Void\_Linux