**Санкт-Петербургский политехнический университет Петра Великого**

**Институт кибербезопасности и защиты информации**

ЛАБОРАТОРНАЯ РАБОТА № 4

**Реализация Telegram-бота на языке**

**программирования Python**

по дисциплине "Цифровая культура"

Выполнил

студент гр. 4851004/10001 Веремейчук Я. Ю.

<*подпись*>

Помощник преподавателя Данилов В. Д.

<*подпись*>

Санкт-Петербург

2023

**Цель работы**

Получение навыков создания Telegram-ботов на языке программирования Python.

**Задание на работу**

1. Установить модуль telebot для работы с Telegram-ботом [https://pypi.org/project/pyTelegramBotAPI/].

2. Изучить функционал модуля telebot.

3. Получить токен для работы с ботом путем обращения к боту @botfather.

4. Реализовать собственный Telegram-бот, который поддерживает следующие команды: ⎯ /register – команда для регистрации пользователя; после отправки данной команды пользователю необходимо ввести пароль для успешной регистрации. ⎯ /login – команда для прохождения аутентификации; после отправки данной команды пользователю необходимо ввести пароль, введенный при регистрации, для дальнейшего использования команды /predict; ⎯ /predict – команда для проведения бинарной классификации картинки; после отправки данной команды пользователю необходимо отправить боту картинку; в ответ бот присылает ответ – на картинке изображен человек или животное; реализацию бинарной классификации необходимо использовать из лабораторной работы №3. ⎯ /logout – команда для выхода из системы; после отправки данной команды пользователь не может пользоваться командой /predict до тех пор, пока заново не пройдет аутентификацию с помощью /login.

5. Важно, чтобы только зарегистрированный пользователь, прошедший аутентификацию, мог использовать команду /predict.

6. Один из вариантов реализации отслеживания состояний пользователей – это хранение следующих параметров: ID чата, пароль, флаг (осуществлен вход или нет).

7. Ответить на контрольные вопросы

**Ход работы**

Был установлен модуль telebot для работы с Telegram-ботом. Был изучен функционал модуля telebot. Был получен токен для работы с ботом путем обращения к боту @botfather.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 1 – Работа бота

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 2 – Работа бота

Изображение выглядит как трава, крупный рогатый скот, текст, скот

Автоматически созданное описание

Рисунок 3 – Работа бота

Изображение выглядит как текст, одежда, Человеческое лицо, снимок экрана

Автоматически созданное описание

Рисунок 4 – Работа бота

**Ответы на контрольные вопросы**

1. **Что такое аутентификация?**

Аутентификация — процедура проверки подлинности, например проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных.

1. **Что такое авторизация?**

Авторизация — предоставление определенному лицу или группе лиц прав на выполнение определенных действий.

1. **Чем аутентификация отличается от авторизации?**

|  |  |
| --- | --- |
| **Аутентификация** | **Авторизация** |
| Процедура, направленная на подтверждение истинности пользователя или его образа | Процедура, учитывающая присвоение, проверку прав пользователя на выполнение им определенных действий в отношении информационных активов |
| Зависима от данных, которые предоставляет пользователь | Никак не зависит от действий пользователя |
| Приводится в исполнение в рамках одной сессии, не требует повторения | Требуется к выполнению каждый раз, когда возникает необходимость что-то сделать в системе. Может повторяться неограниченное число раз, если не стоит запрет на число попыток |

1. **Для чего нужен токен Telegram-бота?**

Токен нужен, чтобы подключить бота к платформе управления рассылками. Там можно настраивать вовлекающие воронки для работы с аудиторией.

С помощью токена можно контролировать чат-бот, поэтому его важно хранить в безопасном месте.

1. **Каким образом можно осуществить безопасное хранение пароля?**

Зашифрованный файл, менеджер паролей, браузер, внешнее устройство, хэширование.

**Вывод**

Были получены навыки создания Telegram-ботов на языке программирования Python.

**Приложение**

import telebot

import tensorflow as tf

import numpy as np

import matplotlib.pyplot as plt

from keras.api.\_v2.keras.preprocessing import image

bot = telebot.TeleBot("6033305186:AAECFX5-1aV\_pqdBdaUyHnv1noM7iXhGS1c")

global flag

@bot.message\_handler(commands=['start'])

def send\_welcome(message):

bot.reply\_to(message, "Привет, чем помочь?")

@bot.message\_handler(commands=['help'])

def send\_welcome(message):

bot.reply\_to(message, "/register - зарегаться\n /login - ввести пароль\n /predict - бинарная классификация\n /logout - выйти")

@bot.message\_handler(content\_types=['text'])

def start(message):

global flag

if message.text == '/register':

bot.send\_message(message.from\_user.id, "Круто, что ты хочешь зарегаться! Введи новый пароль!")

bot.register\_next\_step\_handler(message, get\_pincode)

elif message.text == '/login':

bot.send\_message(message.from\_user.id, "Чтобы продолжить работу, введи пароль!")

bot.register\_next\_step\_handler(message, check\_pincode)

elif message.text == '/predict':

if flag == 0:

bot.send\_message(message.from\_user.id, "Отказано в доступе!")

else:

bot.send\_message(message.from\_user.id, "Присылайте вашу картинку!")

bot.register\_next\_step\_handler(message, predict\_image)

elif message.text == '/logout':

flag = 0

bot.send\_message(message.from\_user.id, "Вы вышли из системы!")

bot.register\_next\_step\_handler(message, check\_pincode)

else:

bot.send\_message(message.from\_user.id, 'Напиши /help')

pincode = ''

def get\_pincode(message):

global pincode

pincode = message.text

with open('pincodes.txt', 'a') as f:

f.write(pincode)

f.write('\n')

bot.send\_message(message.from\_user.id, 'Пароль сохранен!')

def check\_pincode(message):

global pincode

global flag

flag = 0

pincode = message.text

with open('pincodes.txt', 'r') as f:

for line in f:

if pincode in line:

flag = 1

break

else:

flag = 0

if flag == 1:

bot.send\_message(message.from\_user.id, 'Пароль совпал!')

else:

bot.send\_message(message.from\_user.id, 'Пароль не найден!')

def predict\_image(message):

file\_photo = bot.get\_file(message.photo[len(message.photo) - 1].file\_id)

downloaded\_photo = bot.download\_file(file\_photo.file\_path)

new\_photo = 'D:/Python\_Projects\_Univer/lab3\_input\_file/new\_photo/new\_photo.jpg'

with open(new\_photo, 'wb') as new\_file:

new\_file.write(downloaded\_photo)

model = tf.keras.models.load\_model('model.h5')

img = image.load\_img(new\_photo, target\_size=(200, 200))

x = image.img\_to\_array(img)

plt.imshow(x / 255.)

x = np.expand\_dims(x, axis=0)

images = np.vstack([x])

classes = model.predict(images, batch\_size=10)

print(classes[0])

if classes[0] < 0.5:

bot.send\_message(message.from\_user.id, 'Это коровка!')

else:

bot.send\_message(message.from\_user.id, 'Это человек!')

bot.infinity\_polling()