**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

**Отчет по практической работе**

по дисциплине «Введение в информационные технологии» на тему:

**Telegram-бот**

Выполнил: студент группы БВТ2108

Пузырёва Елизавета Андреевна

Проверил:

Мкртчян Грач Маратович

Москва

2021

1. **Цель работы:**

Создание простого Telegram-бота, который сможет обработать три сообщения, обработать три любые команды, подробно распишет команду /help.

1. **Ход работы:**

* **Создаем новый проект с именем Simple-bot**
* **Загружаем библиотеку для создания Telegram-ботов**

pip3 install pyTelegramBotAPI

* **Регистрируем бота в Telegram**
  + В поисковой строке ищем бота с именем @BotFather



* + Отправляем боту команду/сообщение – /start
  + Отправляем боту команду/сообщение с именем бота, которого хотим зарегистрировать, в нашем случае <Ваша фамилия>\_<Ваше имя>\_bot
  + Для ознакомления с перечнем команд @BotFather отправляем боту команду/сообщение – /help
* **В файл main.py импортируем библиотеки для создания back-end части бота**

import telebot

from telebot import types

* **Из сообщения об успешной регистрации бота копируем токен для управления нашим ботом. В файле main.py оздаем переменную, хранящую в себе токен**

token = "Ваш токен"

* **Создаем объект бота, к которому мы будем в дальнейшем обращаться**

bot = telebot.TeleBot(token)

В класс TeleBot передаем переменную token, для того, чтобы обращаться именно к вашему боту

* **Создаем декоратор, отвечающий за команду /start**

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

def start(message):

keyboard = types.ReplyKeyboardMarkup()

keyboard.row("Хочу", "/help")

bot.send\_message(message.chat.id, 'Привет! Хочешь узнать свежую информацию о МТУСИ?', reply\_markup=keyboard)

Класс ReplyKeyboardMarkup создает пользовательскую клавиатуру с текстовыми кнопками на месте стандартной клавиатуры.

Метод row() заполняет клавиатуру кнопками.

Метод send\_message отправляет пользователю сообщение.

Аргумент message.chat.id используется для того, чтобы бот отправил сообщение тому пользователю, который отправил сообщение, на которое бот в данный момент времени отвечает.

Аргумент reply\_markup=keyboard используется для отправки пользовательской клавиатуры, для ее дальнейшего отображения.

* **Создаем декоратор отвечающий за команду /help**

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

def start\_message(message):

bot.send\_message(message.chat.id, 'Я умею...')

В сообщении вы можете указать что умеет бот, включая команды, на которые он умеет реагировать.

* **Создаем декоратор отвечающий за ответ на сообщение "Хочу"**

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

def answer(message):

if message.text.lower() == "хочу":

bot.send\_message(message.chat.id, 'Тогда тебе сюда – <https://mtuci.ru/>')

Данный декоратор должен стоять ниже, чем декораторы команд, так как в противном случае декораторы команд обрабатываться не будут, потому что команды в своем роде тоже текстовые сообщения.

В этом декораторе аргумент content\_types=['text'] отвечает за реакцию на текстовый тип контента сообщения.

Для проверки конкретного текста используется условная конструкция с условием message.text.lower() == "<текст>". Причем функция lower() отвечает за перевод текста в нижний регистр для удобства использования, и может применяться не только для библиотеки telebot, но и для любых строковых операций и переменных.

* **Создаем декоратор отвечающий за пользовательскую клавиатуру**

markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)

item1=types.KeyboardButton("МТУСИ")

markup.add(item1)

* **Создаем фото профиля бота и описание**

Боту @BotFather отправляем команду /setuserpic , выбираем бота которому хотим поставить фото и отправляем само фото.

Чтобы написать описание бота отправим команду /setdescription и описание которое мы бы хотели видеть.

1. **Вывод:**

В итоге мы создали такой вот простой **Telegram-бот**, который может обработать несколько сообщений, обработать несколько команд, подробно расписать команду /help.

