**Московский государственный технический**

**университет им. Н.Э. Баумана**

**Факультет ИУ**

**Кафедра ИУ5**

**Курс «Основы информатики»**

**Отчет лабораторной работе №6**

Выполнил студент группы ИУ5-36Б:

Турланов В. Е.

Подпись и дата:

Проверил преподаватель каф.:

Нардид А. Н.

Подпись и дата:

Москва, 2024 г

**Разработка бота на основе конечного автомата для Telegram с использованием языка Python.**

**Цель лабораторной работы:** изучение разработки ботов в Telegram.

**Требования к отчету:**

Отчет по лабораторной работе должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

**Задание:**

1. Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

**data\_base.py**

import sqlite3  
  
  
def init\_db():  
 conn = sqlite3.connect('bot\_db.sqlite')  
 cursor = conn.cursor()  
  
 # Создание таблицы для хранения общего числа  
 cursor.execute('''  
 CREATE TABLE IF NOT EXISTS state (  
 id INTEGER PRIMARY KEY,  
 value INTEGER NOT NULL  
 )  
 ''')  
  
 # Создание таблицы для хранения истории команд  
 cursor.execute('''  
 CREATE TABLE IF NOT EXISTS history (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 user\_id INTEGER NOT NULL,  
 command TEXT NOT NULL  
 )  
 ''')  
  
 # Инициализация общего числа, если оно отсутствует  
 cursor.execute('SELECT \* FROM state')  
 if cursor.fetchone() is None:  
 cursor.execute('INSERT INTO state (value) VALUES (0)')  
  
 conn.commit()  
 conn.close()  
  
  
def get\_value():  
 conn = sqlite3.connect('bot\_db.sqlite')  
 cursor = conn.cursor()  
  
 cursor.execute('SELECT value FROM state WHERE id = 1')  
 value = cursor.fetchone()[0]  
  
 conn.close()  
 return value  
  
  
def update\_value(new\_value):  
 conn = sqlite3.connect('bot\_db.sqlite')  
 cursor = conn.cursor()  
  
 cursor.execute('UPDATE state SET value = ? WHERE id = 1', (new\_value,))  
  
 conn.commit()  
 conn.close()  
  
  
def add\_to\_history(user\_id, command):  
 conn = sqlite3.connect('bot\_db.sqlite')  
 cursor = conn.cursor()  
  
 cursor.execute('INSERT INTO history (user\_id, command) VALUES (?, ?)', (user\_id, command))  
  
 conn.commit()  
 conn.close()  
  
  
def get\_history():  
 conn = sqlite3.connect('bot\_db.sqlite')  
 cursor = conn.cursor()  
  
 cursor.execute('SELECT user\_id, command FROM history')  
 history = cursor.fetchall()  
  
 conn.close()  
 return history

**main.py**

import telebot  
from telebot import types  
from database import init\_db, get\_value, update\_value, add\_to\_history, get\_history  
  
# Инициализация базы данных  
init\_db()  
  
  
tg\_bot\_token = '7949443626:AAEJXtUyaWbcWkNQgxv1QxoO9MuM28Z\_ERc'  
bot = telebot.TeleBot(tg\_bot\_token)  
  
  
@bot.message\_handler(commands=['start'])  
def start\_command(message):  
 bot.reply\_to(message, "Добро пожаловать! Используйте кнопки ниже для управления общим числом.")  
 show\_buttons(message.chat.id)  
  
  
def show\_buttons(chat\_id):  
 markup = types.InlineKeyboardMarkup()  
 increase\_button = types.InlineKeyboardButton("Увеличить на 2", callback\_data='increase')  
 decrease\_button = types.InlineKeyboardButton("Уменьшить на 2", callback\_data='decrease')  
 history\_button = types.InlineKeyboardButton("Показать историю", callback\_data='history')  
  
 markup.add(increase\_button, decrease\_button, history\_button)  
  
 bot.send\_message(chat\_id, f"Текущее число: {get\_value()}", reply\_markup=markup)  
  
  
@bot.callback\_query\_handler(func=lambda call: True)  
def handle\_query(call):  
 if call.data == 'increase':  
 new\_value = get\_value() + 2  
 update\_value(new\_value)  
 add\_to\_history(call.from\_user.id, 'increase')  
 bot.answer\_callback\_query(call.id, "Число увеличено на 2.")  
  
 elif call.data == 'decrease':  
 new\_value = get\_value() - 2  
 update\_value(new\_value)  
 add\_to\_history(call.from\_user.id, 'decrease')  
 bot.answer\_callback\_query(call.id, "Число уменьшено на 2.")  
  
 elif call.data == 'history':  
 history = get\_history()  
 history\_message = "История команд:\n"  
 for user\_id, command in history:  
 history\_message += f"Пользователь {user\_id}: {command}\n"  
  
 bot.send\_message(call.message.chat.id, history\_message)  
  
 show\_buttons(call.message.chat.id)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 bot.polling(none\_stop=True)

