Московский государственный технический университет имени Н.Э.Баумана

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет РТ Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по лабораторной работе № 6 по курсу**

**Базовые компоненты интернет-технологий**

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

|  |  |  |
| --- | --- | --- |
| Исполнитель |  |  |
| студент группы РТ5-31б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Титов А.Д. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |
|  |  |  |
| Проверил |  |  |
| Доцент кафедры ИУ5 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гапанюк Ю.Е. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |

Москва -  2021

**Описание задания**

Цель лабораторной работы: изучение разработки ботов в Telegram. Разработать бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

**Текст программы**

**bot.py**

import telebot

from telebot import types

import config

import dbworker

# Создание бота

bot = telebot.TeleBot(config.TOKEN)

# Начало диалога

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

def cmd\_start(message):

bot.send\_message(message.chat.id, 'Я умею выполнять действия над двумя числами!')

dbworker.set(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE), config.States.STATE\_FIRST\_NUM.value)

bot.send\_message(message.chat.id, 'Введите первое число')

# По команде /reset будем сбрасывать состояния, возвращаясь к началу диалога

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

def cmd\_reset(message):

bot.send\_message(message.chat.id, 'Сбрасываем результаты предыдущего ввода.')

dbworker.set(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE), config.States.STATE\_FIRST\_NUM.value)

bot.send\_message(message.chat.id, 'Введите первое число')

# Обработка первого числа

@bot.message\_handler(func=lambda message: dbworker.get(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE)) == config.States.STATE\_FIRST\_NUM.value)

def first\_num(message):

text = message.text

if not text.isdigit():

# Состояние не изменяется, выводится сообщение об ошибке

bot.send\_message(message.chat.id, 'Пожалуйста введите число!')

return

else:

bot.send\_message(message.chat.id, f'Вы ввели первое число {text}')

# Меняем текущее состояние

dbworker.set(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE), config.States.STATE\_SECOND\_NUM.value)

# Сохраняем первое число

dbworker.set(dbworker.make\_key(message.chat.id, config.States.STATE\_FIRST\_NUM.value), text)

bot.send\_message(message.chat.id, 'Введите второе число')

# Обработка второго числа

@bot.message\_handler(func=lambda message: dbworker.get(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE)) == config.States.STATE\_SECOND\_NUM.value)

def second\_num(message):

text = message.text

if not text.isdigit():

# Состояние не изменяется, выводится сообщение об ошибке

bot.send\_message(message.chat.id, 'Пожалуйста введите число!')

else:

bot.send\_message(message.chat.id, f'Вы ввели второе число {text}')

# Меняем текущее состояние

dbworker.set(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE), config.States.STATE\_OPERATION.value)

# Сохраняем первое число

dbworker.set(dbworker.make\_key(message.chat.id, config.States.STATE\_SECOND\_NUM.value), text)

markup = types.ReplyKeyboardMarkup(row\_width=4)

itembtn1 = types.KeyboardButton('+')

itembtn2 = types.KeyboardButton('\*')

itembtn3 = types.KeyboardButton('-')

itembtn4 = types.KeyboardButton('/')

markup.add(itembtn1, itembtn2, itembtn3, itembtn4)

bot.send\_message(message.chat.id, 'Выберите пожалуйста действие', reply\_markup = markup)

# Выбор действия

@bot.message\_handler(func=lambda message: dbworker.get(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE)) == config.States.STATE\_OPERATION.value)

def operation(message):

# Текущее действие

op = message.text

# Читаем операнды из базы данных

v1 = dbworker.get(dbworker.make\_key(message.chat.id, config.States.STATE\_FIRST\_NUM.value))

v2 = dbworker.get(dbworker.make\_key(message.chat.id, config.States.STATE\_SECOND\_NUM.value))

# Выполняем действие

fv1 = float(v1)

fv2 = float(v2)

res = None

markup = types.ReplyKeyboardRemove(selective=False)

try:

if op == '+':

res = fv1 + fv2

elif op == '\*':

res = fv1 \* fv2

elif op == '-':

res = fv1 - fv2

else:

res = fv1 / fv2

res = f'{res:.2f}'

# Выводим результат

bot.send\_message(message.chat.id, f'Результат: {v1}{op}{v2}={str(res)}', reply\_markup = markup)

except:

bot.send\_message(message.chat.id, 'Деление на ноль!', reply\_markup = markup)

# Меняем текущее состояние

dbworker.set(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE), config.States.STATE\_FIRST\_NUM.value)

# Выводим сообщение

bot.send\_message(message.chat.id, 'Введите первое число')

if \_\_name\_\_ == '\_\_main\_\_':

bot.infinity\_polling()

**config.py**

from enum import Enum

# Токент бота

TOKEN = "2141261741:AAG00ycT6v9F1lEDH4ezyzZGAhFQm\_QYcIE"

# Файл базы данных Vedis

db\_file = "db.vdb"

# Ключ записи в БД для текущего состояния

CURRENT\_STATE = "CURRENT\_STATE"

# Состояния автомата

class States(Enum):

STATE\_START = "STATE\_START" # Начало нового диалога

STATE\_FIRST\_NUM = "STATE\_FIRST\_NUM"

STATE\_SECOND\_NUM = "STATE\_SECOND\_NUM"

STATE\_OPERATION = "STATE\_OPERATION"

**dbworker.py**

from vedis import Vedis

import config

# Чтение значения

def get(key):

with Vedis(config.db\_file) as db:

try:

return db[key].decode()

except KeyError:

# в случае ошибки значение по умолчанию - начало диалога

return config.States.STATE\_START.value

# Запись значения

def set(key, value):

with Vedis(config.db\_file) as db:

try:

db[key] = value

return True

except:

# тут желательно как-то обработать ситуацию

return config.States.STATE\_START.value

# Создание ключа для записи и чтения

def make\_key(chatid, keyid):

res = str(chatid) + '\_\_' + str(keyid)

return res

**Примеры выполнения программы**

