



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ГУИМЦ»
КАФЕДРА ИУ5 «Системы обработки информации и управления»

Дисциплина «Базовые компоненты интернет-технологий»
ОТЧЕТ

Лабораторная работа №6
«Разработка бота на основе конечного автомата для Telegram с
использованием языка Python»

Студент: Соловьева А.М., группа ИУ5Ц-53Б
Преподаватель: Гапанюк Ю.Е.

2021 г.

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
1. Описание задания:	3
2. Листинг программы:	3
3. Результат выполнения программы:	6

Цель лабораторной работы: изучение разработки ботов в Telegram.

1. Описание задания:

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

2. Листинг программы:

main.py

```
import telebot
import config
import db
import math
import numpy

bot = telebot.TeleBot(config.token)

@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Решение биквадратного уравнения')
    db.set(db.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: db.get(
    db.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_FIRST_NUM.value)
def first_num(message):
    text = message.text
    try:
        float(text)
        if text == '0':
            bot.send_message(message.chat.id, 'Первое число не может быть ноль!')
            bot.send_message(message.chat.id, 'Введите первое число')
            return
        else:
            # Состояние не изменяется, выводится сообщение об ошибке
            # Меняем текущее состояние
            db.set(db.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_SECOND_NUM.value)
            # Сохраняем первое число
            db.set(db.make_key(message.chat.id,
config.States.STATE_FIRST_NUM.value), text)
            bot.send_message(message.chat.id, 'Введите второе число')
    except ValueError:
        # Состояние не изменяется, выводится сообщение об ошибке
        bot.send_message(message.chat.id, 'Пожалуйста, введите число!')
        return

# Обработка второго числа
@bot.message_handler(func=lambda message: db.get(
    db.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_SECOND_NUM.value)
def second_num(message):
    text = message.text
```

```

    try:
        float(text)
        # Меняем текущее состояние
        db.set(db.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_THIRD_NUM.value)
        # Сохраняем второе число
        db.set(db.make_key(message.chat.id,
config.States.STATE_SECOND_NUM.value), text)
        bot.send_message(message.chat.id, 'Введите третье число')

    except ValueError:
        bot.send_message(message.chat.id, 'Пожалуйста, введите число!')
        return

# Обработка третьего числа
@bot.message_handler(func=lambda message: db.get(
    db.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_THIRD_NUM.value)
def third_num(message):
    text = message.text
    try:
        float(text)
        # Сохраняем третье число
        db.set(db.make_key(message.chat.id, config.States.STATE_THIRD_NUM.value),
text)

        # Нахождение корней
        v1 = db.get(db.make_key(message.chat.id,
config.States.STATE_FIRST_NUM.value))
        v2 = db.get(db.make_key(message.chat.id,
config.States.STATE_SECOND_NUM.value))
        v3 = db.get(db.make_key(message.chat.id,
config.States.STATE_THIRD_NUM.value))
        a = float(v1)
        b = float(v2)
        c = float(v3)
        result = []
        D = b * b - 4 * a * c
        if D == 0.0:
            root = -b / (2.0 * a)
            result.append(root)
        elif D > 0.0:
            sqD = math.sqrt(D)
            root1 = (-b + sqD) / (2.0 * a)
            root2 = (-b - sqD) / (2.0 * a)
            result.append(root1)
            result.append(root2)
        resultfinal = []
        for x in result:
            if x > 0:
                resultfinal.append(numpy.sqrt(x))
                resultfinal.append(-numpy.sqrt(x))
            elif x == 0:
                resultfinal.append(0)
        bot.send_message(message.chat.id, "Имеем корни:{}".format(resultfinal))
        # Меняем текущее состояние
        db.set(db.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_FIRST_NUM.value)
        # Выводим сообщение
        bot.send_message(message.chat.id, 'Введите следующее первое число')
    except ValueError:
        bot.send_message(message.chat.id, 'Введите число!')
        return

```

```
if __name__ == '__main__': # делает нам бесконечный цикл получения данных
    bot.infinity_polling()
```

config.py

```
from enum import Enum

# Токент бота
token = '5043420848:AAFSQjtCTJCbUFlXrh_NVcSvRKtFCG0Qu0Q'

# 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_THIRD_NUM = "STATE_THIRD_NUM"
```

db.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.S_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 False

# Создание ключа для записи и чтения
def make_key(chatid, keyid):
    res = str(chatid) + '__' + str(keyid)
    return res
```

3. Результат выполнения программы:



