# 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 «Информатика и системы управления» Кафедра ИУ5 «Системы обработки информации и управления»**

Лабораторная работа №6

по дисциплине «Базовые компоненты интернет-технологий»

Выполнил: студент группы ИУ5-33Б

Требуков Д.А.

Проверил: Канев А.И.

2021 г.

**Задание:**

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

**Текст 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, 'Решим биквадратное уравнение, с тебя только коэффициенты!')

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, 'Первый коэффициент 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 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 = '2114680737:AAE5w5v9UDxLv51rBFUjPA12IEBraDaT5qY'

# 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" STATE\_OPERATION = "STATE\_OPERATION"

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

**Результаты выполнения:**

