

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»

Курс «Базовые компоненты интернет-технологий»

Отчёт по лабораторной работе №6

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

Выполнил:

студент группы ИУ5-33Б Кузнецов В. А.

подпись: _____, дата: _____

Проверил:

лектор Гапанюк Ю. Е.

подпись: _____, дата: _____

2022 г.

Задание:

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

Текст программы:

dbworker.py

```
from enum import Enum
from vedis import Vedis

# Файл базы данных Vedis
db_file = "db.vdb"
# Ключ записи в БД для текущего состояния
CURRENT_STATE = "CURRENT_STATE"

class States(Enum):
    """Состояния автомата"""
    STATE_COEFFICIENT_A = 0
    STATE_COEFFICIENT_B = 1
    STATE_COEFFICIENT_C = 2

    def __next__(self):
        try:
            return States(self.value + 1)
        except ValueError:
            raise StopIteration

    def get(key):
        """Чтение значения"""
        with Vedis(db_file) as db:
            try:
                return db[key].decode()
            except KeyError:
                # в случае ошибки значение по умолчанию - начало диалога
                return States(0)

    def set(key, value):
        """Запись значения"""
        with Vedis(db_file) as db:
            try:
                db[key] = value
                return True
            except:
                return False

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

equation_solving.py

```
import math

def get_roots(a, b, c):
    result = []

    # Частный случай квадратного уравнения
    if a == 0.0:
        if b != 0.0:
            root = -c / b
            if root > 0:
                result.extend((math.sqrt(root), -math.sqrt(root)))
            if c == 0:
                result.append(0)
        return result

    # Общий случай решения биквадратного уравнения
    D = b * b - 4 * a * c

    if c == 0:
        result.append(0)

    if D == 0.0:
        root = -b / (2.0 * a)
        if root > 0:
            result.extend((math.sqrt(root), -math.sqrt(root)))
    elif D > 0.0:
        sqD = math.sqrt(D)
        root1 = (-b + sqD) / (2.0 * a)
        root2 = (-b - sqD) / (2.0 * a)
        if root1 > 0:
            result.extend((math.sqrt(root1), -math.sqrt(root1)))
        if root2 > 0:
            result.extend((math.sqrt(root2), -math.sqrt(root2)))
    return result

def calc_result(a, b, c):
    # Составить биквадратное уравнение
    result = 'Биквадратное уравнение: '
    if a:
        result += str(a).rstrip('0').rstrip('.') + 'x4'
    if b:
        if b > 0 and a:
            result += '+'
        result += str(b).rstrip('0').rstrip('.') + 'x2'
    if c:
        if c > 0 and (a or b):
            result += '+'
        result += str(c).rstrip('0').rstrip('.')
    elif not a and not b:
        result += '0'
    result += '=0\n'

    # Вычисление корней
    roots = get_roots(a, b, c)

    # Дописать решение
    match len(roots):
```

```

    case 0:
        result += 'Не имеет корней'
    case 1:
        result += 'Имеет один корень:\n{}'.format(*roots)
    case 2:
        result += 'Имеет два корня:\n{}\n{}'.format(*roots)
    case 3:
        result += 'Имеет три корня:\n{}\n{}\n{}'.format(*roots)
    case 4:
        result += 'Имеет четыре корня:\n{}\n{}\n{}\n{}'.format(*roots)

    return result

```

equation_test.py

```

import unittest
from equation_solving import get_roots

class BiquadrateTest(unittest.TestCase):
    def test_no_roots(self):
        expected_answer = set()
        self.assertEqual(expected_answer, set(get_roots(0, 0, 0)))
        self.assertEqual(expected_answer, set(get_roots(0, 1, 4)))
        self.assertEqual(expected_answer, set(get_roots(0, -1, -4)))
        self.assertEqual(expected_answer, set(get_roots(1, 0, 4)))
        self.assertEqual(expected_answer, set(get_roots(-1, 0, -4)))

    def test_one_root(self):
        expected_answer = {0}
        self.assertEqual(expected_answer, set(get_roots(0, 1, 0)))
        self.assertEqual(expected_answer, set(get_roots(1, 0, 0)))
        self.assertEqual(expected_answer, set(get_roots(0, -1, 0)))
        self.assertEqual(expected_answer, set(get_roots(-1, 0, 0)))

    def test_two_roots(self):
        expected_answer = {-2, 2}
        self.assertEqual(expected_answer, set(get_roots(0, 1, -4)))
        self.assertEqual(expected_answer, set(get_roots(0, -1, 4)))
        self.assertEqual(expected_answer, set(get_roots(1, 0, -16)))
        self.assertEqual(expected_answer, set(get_roots(-1, 0, 16)))

    def test_three_roots(self):
        expected_answer = {-2, 0, 2}
        self.assertEqual(expected_answer, set(get_roots(1, -4, 0)))
        self.assertEqual(expected_answer, set(get_roots(-1, 4, 0)))

    def test_four_roots(self):
        expected_answer = {-3, -1, 1, 3}
        self.assertEqual(expected_answer, set(get_roots(1, -10, 9)))
        self.assertEqual(expected_answer, set(get_roots(-1, 10, -9)))

if __name__ == '__main__':
    unittest.main()

```

main.py

```
import os
import telebot
from dotenv import load_dotenv
import dbworker
from equation_solving import calc_result

load_dotenv()
TOKEN = os.environ.get('TOKEN')

# Создание бота
bot = telebot.TeleBot(TOKEN)

# Начало диалога
@bot.message_handler(commands=['start'])
def cmd_start(message):
    bot.send_message(message.chat.id, 'Я решаю биквадратные уравнения!\nОни  
выглядят вот так:  $Ax^4+Bx^2+C=0$ ')
    current_state = dbworker.States.STATE_COEFFICIENT_A
    dbworker.set(dbworker.make_key(message.chat.id, dbworker.CURRENT_STATE),
current_state.value)
    bot.send_message(message.chat.id, f'Введите коэффициент  
{chr(current_state.value + 65)}')

# 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.')
    current_state = dbworker.States.STATE_COEFFICIENT_A
    dbworker.set(dbworker.make_key(message.chat.id, dbworker.CURRENT_STATE),
current_state.value)
    bot.send_message(message.chat.id, f'Введите коэффициент  
{chr(current_state.value + 65)}')

# Считывание коэффициента
@bot.message_handler()
def coefficient(message):
    # Получаем текущее состояние
    current_state =
dbworker.States(int(dbworker.get(dbworker.make_key(message.chat.id,
dbworker.CURRENT_STATE))))

    text = message.text
    if not text.lstrip("-").isdigit():
        # Состояние не изменяется, выводится сообщение об ошибке
        bot.send_message(message.chat.id, 'Пожалуйста введите число!')
        return

    # Сохраняем полученное значение
    # bot.send_message(message.chat.id, f'Вы ввели коэффициент  
{chr(current_state.value + 65)}')
    dbworker.set(dbworker.make_key(message.chat.id, current_state.value),
text)

    # Если есть все три коэффициента, то решаем биквадратное уравнение
    try:
```

```

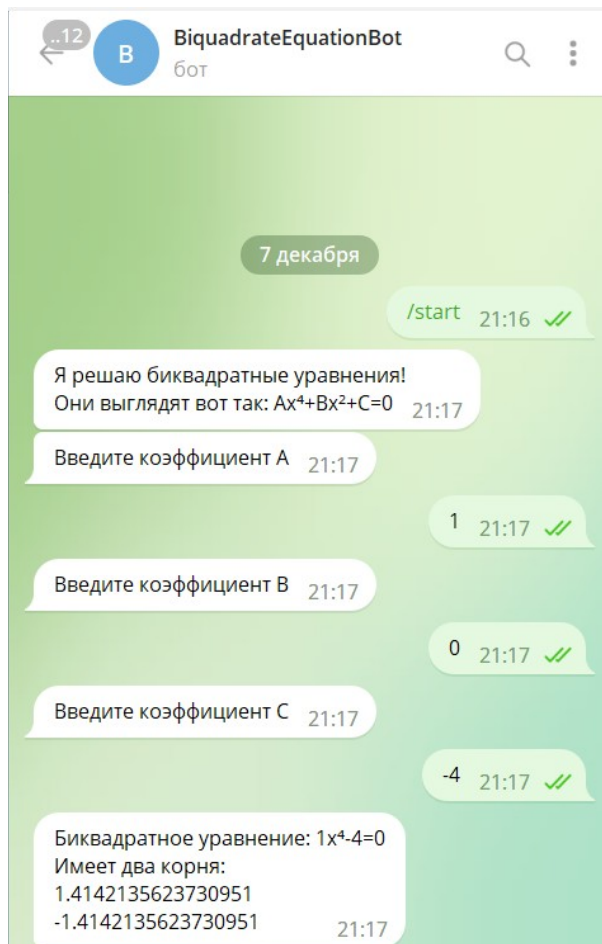
        current_state = next(current_state)
    except StopIteration:
        a = float(dbworker.get(dbworker.make_key(message.chat.id,
dbworker.States.STATE_COEFFICIENT_A.value)))
        b = float(dbworker.get(dbworker.make_key(message.chat.id,
dbworker.States.STATE_COEFFICIENT_B.value)))
        c = float(text)
        bot.send_message(message.chat.id, calc_result(a, b, c))
        current_state = dbworker.States.STATE_COEFFICIENT_A

    # Переходим к следующему состоянию
    dbworker.set(dbworker.make_key(message.chat.id, dbworker.CURRENT_STATE),
current_state.value)
    bot.send_message(message.chat.id, f'Введите коэффициент
{chr(current_state.value + 65)}')

if __name__ == '__main__':
    bot.infinity_polling()

```

Пример выполнения



Введите коэффициент А 21:21

3 21:21 ✓✓

Введите коэффициент В 21:21

-13 21:21 ✓✓

Введите коэффициент С 21:21

4 21:21 ✓✓

Биквадратное уравнение: $3x^4 - 13x^2 + 4 = 0$
Имеет четыре корня:
2
-2
0.5773502691896257
-0.5773502691896257 21:21

Введите коэффициент А 21:21

0 21:21 ✓✓

Введите коэффициент В 21:21

0 21:21 ✓✓

Введите коэффициент С 21:21

0 21:21 ✓✓

Биквадратное уравнение: $0 = 0$
Не имеет корней 21:21

Введите коэффициент А 21:21

0 21:22 ✓✓

Введите коэффициент В 21:22

1 21:22 ✓✓

Введите коэффициент С 21:22

-4 21:22 ✓✓

Биквадратное уравнение: $1x^2 - 4 = 0$
Имеет два корня:
2
-2 21:22