

Факультет РТ Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по домашней работе по курсу  
Базовые компоненты интернет-технологий**

Исполнитель

Студент группы РТ5-31Б

\_\_\_\_\_

Дворкович Ю.А.

“ \_\_\_\_ ” \_\_\_\_\_ 2021 г.

Проверил

Доцент кафедры ИУ5

\_\_\_\_\_

Гапанюк Ю.Е.

“ \_\_\_\_ ” \_\_\_\_\_ 2021 г.

**Содержание:**

1.Описание задания	3
2.Текст программы	3
3.Экранные формы с примерами выполнения программы	9

## 1.Описание задания

1. Модифицируйте код лабораторной работы №6 таким образом, чтобы он был пригоден для модульного тестирования.
2. Используя материалы лабораторной работы №4 создайте модульные тесты с применением TDD - фреймворка (2 теста) и BDD - фреймворка (2 теста).

## 2.Текст программы

*bot.py*

```
import math

import telebot

import config

import dbworker

# Создание бота

bot = telebot.TeleBot(config.TOKEN)

# Начало диалога

@bot.message_handler(commands=['start'])

def cmd_start(message):

    dbworker.set(dbworker.make_key("STATE_FIRST_NUM", config.CURRENT_STATE),
config.States.STATE_FIRST_NUM.value)

    return message

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

@bot.message_handler(commands=['reset'])

def cmd_reset(message):

    dbworker.set(dbworker.make_key("STATE_FIRST_NUM", config.CURRENT_STATE),
config.States.STATE_FIRST_NUM.value)

# 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):
```

```

if not message.isdigit():

    return None

else:

    # Меняем текущее состояние

    dbworker.set(dbworker.make_key("STATE_SECOND_NUM", config.CURRENT_STATE),
config.States.STATE_SECOND_NUM.value)

    # Сохраняем первое число

    dbworker.set(dbworker.make_key("first", config.States.STATE_FIRST_NUM.value), message)

    return message

# 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):

    if not message.isdigit():

        return None

    else:

        # Меняем текущее состояние

        dbworker.set(dbworker.make_key("STATE_OPERATION", config.CURRENT_STATE),
config.States.STATE_OPERATION.value)

        # Сохраняем второе число

        dbworker.set(dbworker.make_key("second", config.States.STATE_SECOND_NUM.value), message)

        return message

# Выбор действия

@bot.message_handler(func=lambda message: dbworker.get(

    dbworker.make_key("STATE_OPERATION", config.CURRENT_STATE)) ==
config.States.STATE_OPERATION.value)

def operation(message):

    # Текущее действие

```

```
op = message

# Читаем операнды из базы данных

v1 = dbworker.get(dbworker.make_key("first", config.States.STATE_FIRST_NUM.value))

v2 = dbworker.get(dbworker.make_key("second", config.States.STATE_SECOND_NUM.value))

# Выполняем действие

fv1 = float(v1)

fv2 = float(v2)

res = None

try:

    if op == '+':

        res = fv1 + fv2

    elif op == '*':

        res = fv1 * fv2

    elif op == '-':

        res = fv1 - fv2

    elif op == '/':

        res = fv1 / fv2

    elif op == '^':

        res = fv1 ** fv2

    elif op == '<>':

        res = math.sqrt(fv1 * fv1 + fv2 * fv2)

except:

    pass

return res
```

*config.py*

```
from enum import Enum
```

```
# Токент бота

TOKEN = '5037121621:AAH3mp2l9SrQhTiq-0zCKVnj1O AQ5_u8uzA'

# 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.S_START.value

# Запись значения

def set(key, value):
```

```

with Vedis(config.db_file) as db:

    try:

        db[key] = value

        return True

    except:

        return config.States.STATE_START.value

# Создание ключа для записи и чтения

def make_key(chatid, keyid):

    res = str(chatid) + '___' + str(keyid)

    return res

```

### *TDD.py*

```

import unittest

import bot

class Test(unittest.TestCase):

    def testOperation(self):

        bot.first_num("3")

        bot.second_num("4")

        self.assertEqual(bot.operation("*"), 12)

        self.assertEqual(bot.operation("<>"), 5)

if __name__ == "__main__":

    unittest.main()

```

### *testBDD.py*

```

from behave import given, when, then

import bot

@given('I send bot message {start}')

def step_impl(context, start: str):

    bot.cmd_start(start)

@when('I send bot first message {firstNum}')

def step_imp2(context, firstNum: str):

    context.firstNum = bot.first_num(firstNum)

```

```

@when('I send bot second message {secondNum}')
def step_imp3(context, secondNum: str):
    context.secondNum = bot.second_num(secondNum)
@then('I send bot operation ^ and get answer {result}')
def step_imp4(context, result: str):
    float(context.firstNum) ** float(context.secondNum) == float(result)
@then('I send bot operation <> and get answer {result}')
def step_imp4(context, result: str):
    math.sqrt(float(context.firstNum)*float(context.firstNum) +
float(context.secondNum)*float(context.secondNum)) == float(result)

```

### *BDD.feature*

Feature: testing bot

Scenario: raise to power digit

Given I send bot message /start  
When I send bot first message 2  
When I send bot second message 5  
Then I send bot operation ^ and get answer 32

Scenario: raise to power digit

Given I send bot message /start  
When I send bot first message 5  
When I send bot second message 12  
Then I send bot operation <> and get answer 13

## **3.Экранные формы с примерами выполнения программы**



Feature: testing bot # features/steps/BDD.feature:1

Scenario: raise to power digit # features/steps/BDD.feature:2  
Given I send bot message /start # features/steps/testBDD.py:6  
When I send bot first message 2 # features/steps/testBDD.py:11  
When I send bot second message 5 # features/steps/testBDD.py:16  
Then I send bot operation ^ and get answer 32 # features/steps/testBDD.py:21

Scenario: sum two digits # features/steps/BDD.feature:7  
Given I send bot message /start # features/steps/testBDD.py:6  
When I send bot first message 5 # features/steps/testBDD.py:11  
When I send bot second message 12 # features/steps/testBDD.py:16  
Then I send bot operation <> and get answer 13 # features/steps/testBDD.py:25

1 feature passed, 0 failed, 0 skipped  
2 scenarios passed, 0 failed, 0 skipped  
8 steps passed, 0 failed, 0 skipped, 0 undefined  
Took 0m0.078s