

**Московский государственный технический  
университет им. Н.Э. Баумана**

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

**Курс «Базовые компоненты интернет-технологий»**

**Домашнее задание**

Выполнил:

Студентка группы ИУ5-34Б  
Федотова Анастасия

Подпись и дата:

Проверил:

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2021 г.

# Постановка задачи

## Задание:

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).

## Текст программы

### bot.py

```
dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_FIRST_WORD.value)
bot.send_message(message.chat.id, 'Вам нужно каждый раз выбирать один вариант из предложенных.')
markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
ibtn1 = types.KeyboardButton('Я женщина')
ibtn2 = types.KeyboardButton('Я мужчина')
ibtn3 = types.KeyboardButton('Я не определился/определилась')
markup.add(ibtn1, ibtn2, ibtn3)
bot.send_message(message.chat.id, 'Выберете один из трех вариантов:', reply_markup=markup)

@bot.message_handler(func=lambda message: dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) == config.States.STATE_FIRST_WORD.value)
def first_word(message):
    text = message.text
    if text == 'Я женщина':
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_FIRST_WORD), 0)
    elif text == 'Я мужчина':
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_FIRST_WORD), 1)
    elif text == 'Я не определился/определилась':
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_FIRST_WORD), 2)
    else:
        markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
        ibtn1 = types.KeyboardButton('Я женщина')
        ibtn2 = types.KeyboardButton('Я мужчина')
        ibtn3 = types.KeyboardButton('Я не определился/определилась')
        markup.add(ibtn1, ibtn2, ibtn3)
        bot.send_message(message.chat.id, 'Выберете один из трех вариантов:', reply_markup=markup)
    return
dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_SECOND_WORD.value)
```

```
import telebot
from telebot import types
from config import get_path
import config
import dbworker

# Создание бота
bot = telebot.TeleBot(config.TOKEN)

@bot.message_handler(commands=['start'])
def cmd_start(message):
    dbworker.set(dbworker.make_key(message.chat.id, config.SENTENCE), '')
    bot.send_message(message.chat.id, 'Сейчас будем решать, что бы выбрать из обуви.')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_FIRST_WORD.value)
    bot.send_message(message.chat.id, 'Вам нужно каждый раз выбирать один вариант из предложенных.')
    markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
    ibtn1 = types.KeyboardButton('Я женщина')
    ibtn2 = types.KeyboardButton('Я мужчина')
    ibtn3 = types.KeyboardButton('Я не определился/определилась')
    markup.add(ibtn1, ibtn2, ibtn3)
    bot.send_message(message.chat.id, 'Выберете один из трех вариантов:', reply_markup=markup)

@bot.message_handler(commands=['reset'])
def cmd_reset(message):
    dbworker.set(dbworker.make_key(message.chat.id, config.SENTENCE), '')
    bot.send_message(message.chat.id, 'Начнем сначала!')
```

```

bot.send_message(message.chat.id, 'Выберете только один из предложенных трех цветов:', reply_markup=markup)

@bot.message_handler(func=lambda message: dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) == config.States.STATE_THIRD_WORD.value)
def third_word(message):
    text = message.text
    if text=="Белый":
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_THIRD_WORD), 0)
    elif text=="Коричневый":
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_THIRD_WORD), 1)
    elif text=="Черный":
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_THIRD_WORD), 2)
    else:
        markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
        ibtn1 = types.KeyboardButton('Белый')
        ibtn2 = types.KeyboardButton('Коричневый')
        ibtn3 = types.KeyboardButton('Черный')
        markup.add(ibtn1, ibtn2, ibtn3)
        bot.send_message(message.chat.id, 'Выберете только один из предложенных трех цветов:', reply_markup=markup)
        return
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_SENTENCE.value)
    markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
    ibtn1 = types.KeyboardButton('38')
    ibtn2 = types.KeyboardButton('39')
    ibtn3 = types.KeyboardButton('40')
    ibtn4 = types.KeyboardButton('41')
    markup.add(ibtn1, ibtn2, ibtn3, ibtn4)
    bot.send_message(message.chat.id, 'Ваш размер ноги:', reply_markup=markup)

markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
ibtn1 = types.KeyboardButton('Шнурки')
ibtn2 = types.KeyboardButton('Молния')
markup.add(ibtn1, ibtn2)
bot.send_message(message.chat.id, 'Выберете вариант застёжки:', reply_markup=markup)

# Обработка второго числа
@bot.message_handler(func=lambda message: dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) == config.States.STATE_SECOND_WORD.value)
def second_word(message):
    text = message.text
    if text=="Молния":
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_SECOND_WORD), 0)
    elif text=="Шнурки":
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_SECOND_WORD), 1)
    else:
        markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
        ibtn1 = types.KeyboardButton('Молния')
        ibtn2 = types.KeyboardButton('Шнурки')
        markup.add(ibtn1, ibtn2)
        bot.send_message(message.chat.id, 'Выберете вариант застёжки:', reply_markup=markup)
        return
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_THIRD_WORD.value)
    markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
    ibtn1 = types.KeyboardButton('Белый')
    ibtn2 = types.KeyboardButton('Коричневый')
    ibtn3 = types.KeyboardButton('Черный')
    markup.add(ibtn1, ibtn2, ibtn3)

@bot.message_handler(func=lambda message: dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) == config.States.STATE_SENTENCE.value)
def again(message):
    text = message.text
    gender=dbworker.get(dbworker.make_key(message.chat.id, config.States.STATE_FIRST_WORD))
    type=dbworker.get(dbworker.make_key(message.chat.id, config.States.STATE_SECOND_WORD))
    color=dbworker.get(dbworker.make_key(message.chat.id, config.States.STATE_THIRD_WORD))
    path=get_path(gender,type,color)
    bot.send_message(message.chat.id, f' Итог: размер "{text}"')
    img = open(path, 'rb')
    bot.send_photo(message.chat.id, img)
    bot.send_message(message.chat.id, 'Может попробовать еще раз!')
    bot.send_message(message.chat.id, 'Вам нужно каждый раз выбирать один вариант из предложенных.')
    markup = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
    ibtn1 = types.KeyboardButton('Я женщина')
    ibtn2 = types.KeyboardButton('Я мужчина')
    ibtn3 = types.KeyboardButton('Я не определился/определилась')
    markup.add(ibtn1, ibtn2, ibtn3)
    bot.send_message(message.chat.id, 'Выберете один из трех вариантов:', reply_markup=markup)
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE), config.States.STATE_FIRST_WORD.value)

if __name__ == '__main__':
    #if os.path.exists('db.vdb'): os.remove('db.vdb')
    bot.infinity_polling()._#работа без остановок

```

```
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:
            return ''

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

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

test.py

```
import config
from config import get_path
import unittest
import os

cur_path = os.path.dirname(os.path.abspath(__file__))

class TestBot(unittest.TestCase):
    def test_1(self):
        w = get_path("1", "1", "0")
        self.assertEqual(w, cur_path+"/img/110.jpg")
    def test_2(self):
        w = get_path("0", "1", "0")
        self.assertEqual(w, cur_path + "/img/010.jpg")
    def test_3(self):
        w = get_path("2", "1", "2")
        self.assertEqual(w, cur_path + "/img/212.jpg")

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

featureBDD.feature

Feature:

One day a test came

Scenario: Test

Given Entered data

When Test\_1!!!!

When Test\_2!!!!

Then Completed

## config.py

```
from enum import Enum
import os

# Токент бота
TOKEN = "5000590462:AAH3nJ0C4mxuE-SySDyMqU01hsE9Rw9N33M"

# Файл базы данных Vedis
db_file = "db.vdb"

# Ключ записи в БД для текущего состояния
CURRENT_STATE = "CURRENT_STATE"

SENTENCE = "SENTENCE"

cur_path = os.path.dirname(os.path.abspath(__file__))

# Состояния автомата
class States(Enum):
    STATE_START = "STATE_START" # Начало нового диалога
    STATE_FIRST_WORD = "STATE_FIRST_WORD"
    STATE_SECOND_WORD = "STATE_SECOND_WORD"
    STATE_THIRD_WORD = "STATE_THIRD_WORD"
    STATE_OPERATION = "STATE_OPERATION"
    STATE_SENTENCE = "STATE_SENTENCE"

def get_path(gender, type, color):
    name = "img/" + gender + type + color + ".jpg"
    return '/'.join([cur_path, name])
```

## stepsBDD.py

```
from behave import given, when, then
from test import TestBot

@given("Entered data")
def start(context):
    context.a = TestBot()

@when("Test 1!!!!")
def calculation(context):
    context.a.test_1()

@when("Test 2!!!!")
def calculation(context):
    context.a.test_2()

@then("Completed")
def end(context):
    pass
```

## Результат выполнения

```
TestBot > test_2()

test x
"/Applications/Python 3.10/IDLE.app/Contents/MacOS/Python" "/Users/mac/Desktop/БКИТ_Федотова/lab 6/test.py"
...
-----
Ran 3 tests in 0.000s
OK
Process finished with exit code 0
```

```
mac@MacBook-Pro-mac lab 6 % /Library/Frameworks/Python.framework/Versions/3.10/bin/behave
Feature: # featureBDD.feature:1
  One day a test came
    Scenario: Test # featureBDD.feature:4
      Given Entered data # steps/stepsBDD.py:4 0.001s
      When Test_1!!!! # steps/stepsBDD.py:8 0.000s
      When Test_2!!!! # steps/stepsBDD.py:13 0.000s
      Then Completed # steps/stepsBDD.py:17 0.000s

1 feature passed, 0 failed, 0 skipped
1 scenario passed, 0 failed, 0 skipped
4 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.001s
mac@MacBook-Pro-mac lab 6 %
```

