**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»**

Домашнее задание

по дисциплине «Базовые компоненты интернет-технологий»

Выполнила:

студентка группы ИУ5-33Б

Пересыпкина М.А.

Проверил:

Канев А.И.

2021 г.

# Постановка задачи:

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

import telebot  
import config  
import dbworker  
  
bot = telebot.TeleBot(config.token)  
  
# Начало диалога  
@bot.message\_handler(commands=["start"])  
def cmd\_start(message):  
 bot.send\_message(message.chat.id, "Привет! Как тебя зовут?")  
 dbworker.set\_state(message.chat.id, config.States.S\_ENTER\_NAME.value)  
  
# 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, "Хорошо, начнём заново. Как тебя зовут?")  
 dbworker.set\_state(message.chat.id, config.States.S\_ENTER\_NAME.value)  
  
@bot.message\_handler(  
 func=lambda message: dbworker.get\_current\_state(message.chat.id) == config.States.S\_ENTER\_NAME.value)  
def user\_entering\_name(message):  
 # В случае с именем не будем ничего проверять, пусть хоть "25671", хоть Евкакий  
 bot.send\_message(message.chat.id, "Отличное имя! Теперь укажи, пожалуйста, сколько тебе лет.")  
 dbworker.set\_state(message.chat.id, config.States.S\_ENTER\_AGE.value)  
  
@bot.message\_handler(func=lambda message: dbworker.get\_current\_state(message.chat.id) == config.States.S\_ENTER\_AGE.value)  
def user\_entering\_age(message):  
 # А вот тут сделаем проверку  
 if not message.text.isdigit():  
 # Состояние не меняем, поэтому только выводим сообщение об ошибке и ждём дальше  
 bot.send\_message(message.chat.id, "Что-то не так, попробуй ещё раз!")  
 return  
 # На данном этапе мы уверены, что message.text можно преобразовать в число, поэтому ничем не рискуем  
 if int(message.text) < 5 or int(message.text) > 100:  
 bot.send\_message(message.chat.id, "Вы указали странный возраст. Напишите его ещё раз.")  
 return  
 else:  
 # Возраст введён корректно, можно идти дальше  
 bot.send\_message(message.chat.id, "Прекрасный возраст! "  
 "Отправь мне какую-нибудь фотографию.")  
 dbworker.set\_state(message.chat.id, config.States.S\_SEND\_PIC.value)  
  
@bot.message\_handler(content\_types=["photo"],  
 func=lambda message: dbworker.get\_current\_state(message.chat.id) == config.States.S\_SEND\_PIC.value)  
def user\_sending\_photo(message):  
 # То, что это фотография, мы уже проверили в хэндлере, никаких дополнительных действий не нужно.  
 bot.send\_message(message.chat.id, "Супер! Больше от тебя ничего не требуется. Если захочешь пообщаться снова - "  
 "отправь команду /start.")  
 dbworker.set\_state(message.chat.id, config.States.S\_START.value)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 bot.infinity\_polling()

config.py

from enum import Enum  
  
token = "5015617089:AAHDaX6zRPxORWEcZsPgMweqHNWKke4X82g"  
db\_file = "database.vdb"  
  
  
class States(Enum):  
 *"""  
 Мы используем БД Vedis, в которой хранимые значения всегда строки,  
 поэтому и тут будем использовать тоже строки (str)  
 """* S\_START = "0" # Начало нового диалога  
 S\_ENTER\_NAME = "1"  
 S\_ENTER\_AGE = "2"  
 S\_SEND\_PIC = "3"

dbworker.py

from vedis import Vedis  
import config  
  
# Пытаемся узнать из базы «состояние» пользователя  
def get\_current\_state(user\_id):  
 with Vedis(config.db\_file) as db:  
 try:  
 return db[user\_id].decode()  
 except KeyError: # Если такого ключа почему-то не оказалось  
 return config.States.S\_START.value # значение по умолчанию - начало диалога  
  
# Сохраняем текущее «состояние» пользователя в нашу базу  
def set\_state(user\_id, value):  
 with Vedis(config.db\_file) as db:  
 try:  
 db[user\_id] = value  
 return True  
 except:  
 # тут желательно как-то обработать ситуацию  
 return False

TDD\_test.py

import unittest  
import sys, os  
from bot import \*  
  
sys.path.append(os.getcwd())  
  
class TestBot(unittest.TestCase):  
 def test\_1(self):  
 self.assertEqual('катя','катя')  
  
 def test\_2(self):  
 self.assertEqual(16, 16)  
  
if \_\_name\_\_== 'main':  
 unittest.main()

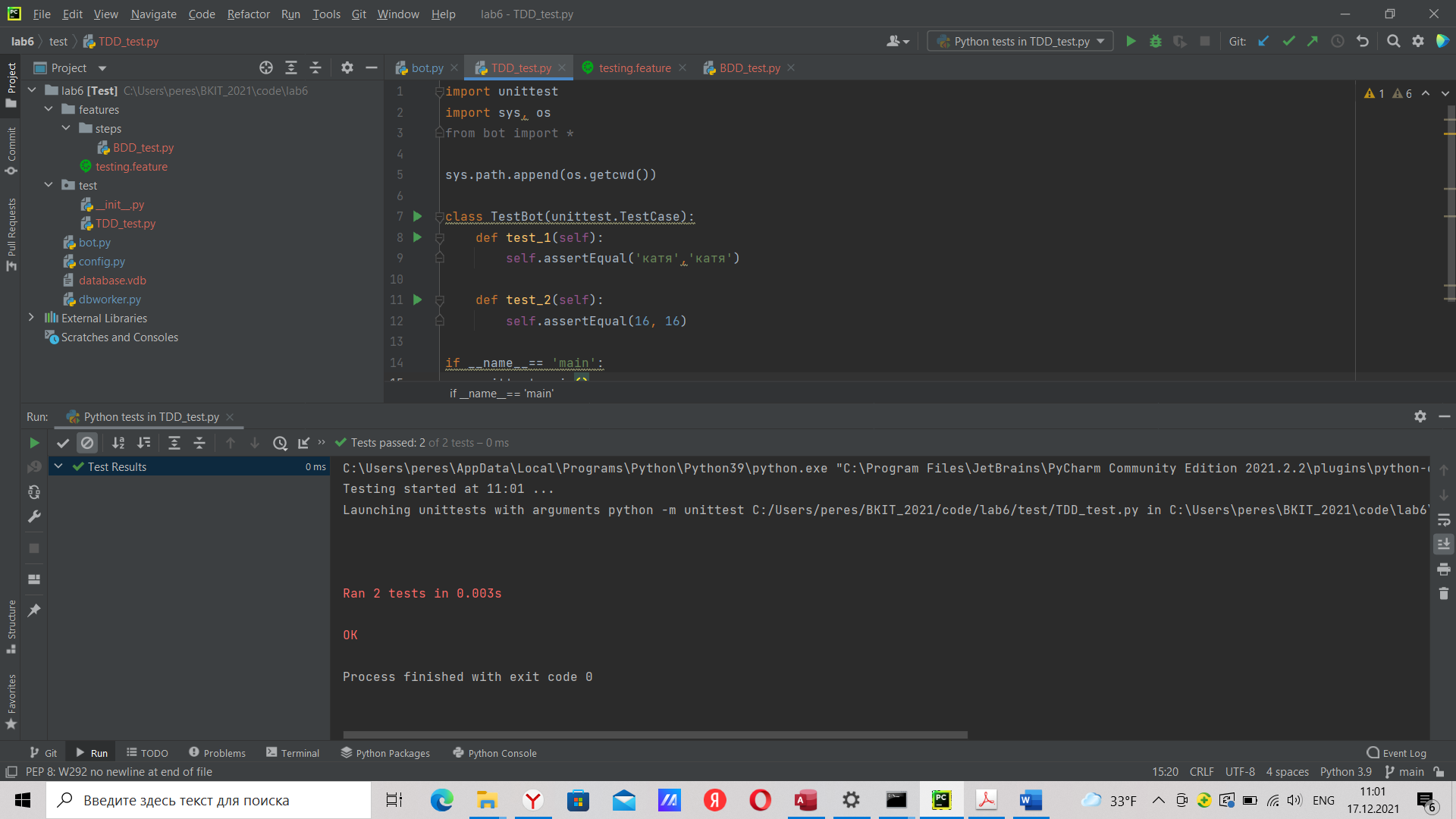
BDD\_test.py

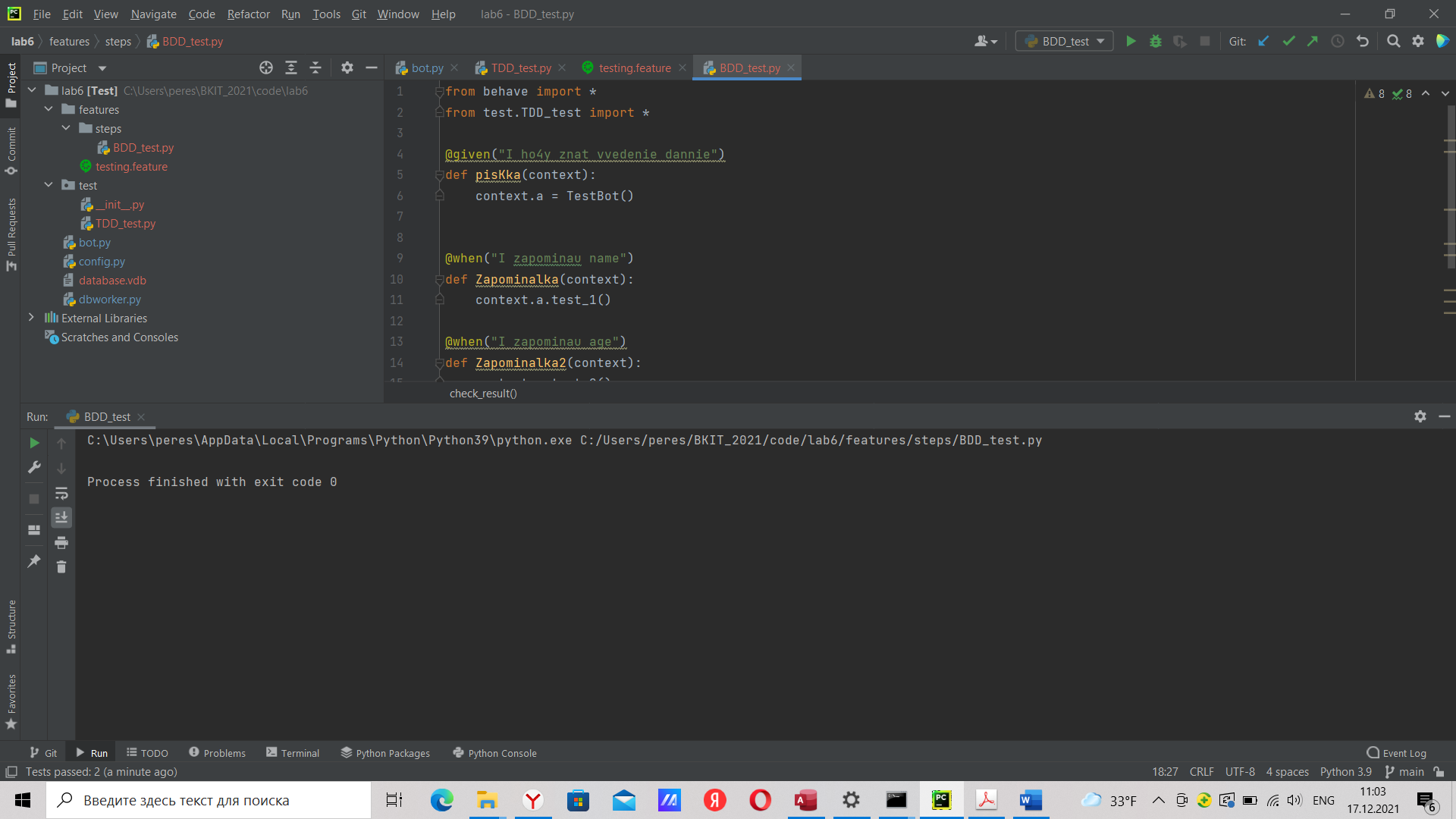
from behave import \*  
from test.TDD\_test import \*  
  
@given("I ho4y znat vvedenie dannie")  
def pisKka(context):  
 context.a = TestBot()  
  
  
@when("I zapominau name")  
def Zapominalka(context):  
 context.a.test\_1()  
  
@when("I zapominau age")  
def Zapominalka2(context):  
 context.a.test\_2()  
  
@then("I xochy 4bee sovpalo!!!!")  
def check\_result(context):  
 pass

testing.feature

Feature: Test  
 Scenario: Test function  
 Given I ho4y znat vvedenie dannie  
 When I zapominau name  
 When I zapominau age  
 Then I xochy 4bee sovpalo!!!!

**Экранные формы с примерами выполнения программы:**

****

****