

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

**Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

Отчёт по домашнему заданию

Выполнил:

студент группы РТ5-31Б
Андреев Виктор

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2021 г

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).

Текст программы

main.py:

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

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

# Начало диалога
@bot.message_handler(commands=['start'])
def cmd_start(message):
    bot.send_message(message.chat.id, 'Я умею выполнять действия над двумя числами!')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_FIRST_NUM.value)
    bot.send_message(message.chat.id, 'Введите первое число')

# 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.')
    dbworker.set(dbworker.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: dbworker.get(
    dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_FIRST_NUM.value)
def first_num(message):
    text = message.text
    if not text.isdigit():
        # Состояние не изменяется, выводится сообщение об ошибке
        bot.send_message(message.chat.id, 'Пожалуйста введите число!')
        return
    else:
        bot.send_message(message.chat.id, f'Вы ввели первое число {text}')
        # Меняем текущее состояние
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_SECOND_NUM.value)
        # Сохраняем первое число
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_FIRST_NUM.value),
text)
        bot.send_message(message.chat.id, 'Введите второе число')

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

```

        return
    else:
        bot.send_message(message.chat.id, f'Вы ввели второе число {text}')
        # Меняем текущее состояние
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_OPERATION.value)
        # Сохраняем первое число
        dbworker.set(dbworker.make_key(message.chat.id, config.States.STATE_SECOND_NUM.value),
text)

        markup = types.ReplyKeyboardMarkup(row_width=2)
        itembtn1 = types.KeyboardButton('+')
        itembtn2 = types.KeyboardButton('*')
        itembtn3 = types.KeyboardButton('-')
        itembtn4 = types.KeyboardButton("/")
        markup.add(itembtn1, itembtn2, itembtn3, itembtn4)
        bot.send_message(message.chat.id, 'Выберите пожалуйста действие', reply_markup=markup)

# Выбор действия
@bot.message_handler(func=lambda message: dbworker.get(
    dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_OPERATION.value)
def operation(message):
    # Текущее действие
    op = message.text
    # Читаем операнды из базы данных
    v1 = dbworker.get(dbworker.make_key(message.chat.id, config.States.STATE_FIRST_NUM.value))
    v2 = dbworker.get(dbworker.make_key(message.chat.id,
config.States.STATE_SECOND_NUM.value))
    # Выполняем действие
    fv1 = float(v1)
    fv2 = float(v2)
    #res = 0
    if op == '+':
        res = summa(fv1, fv2)
    elif op == '*':
        res = raznost(fv1, fv2)
    elif op == '-':
        res = proizv(fv1, fv2)
    elif op == '/':
        res = chastnoe(fv1, fv2)
    else:
        res = "Error"
    # Выводим результат
    markup = types.ReplyKeyboardRemove(selective=False)
    bot.send_message(message.chat.id, f'Результат: {v1}{op}{v2}={str(res)}',
reply_markup=markup)
    # Меняем текущее состояние
    dbworker.set(dbworker.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')

def summa(a, b):
    return a + b

def raznost(a, b):
    return a - b

def proizv(a, b):
    return a * b

def chastnoe(a, b):
    if b == 0:

```

```

        return "Error"
    else:
        return a / b

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

```

config.py:

```

from enum import Enum

# Токент бота
TOKEN = "5094082950:AAHfbYkSZEiZ0QZHiIsnbsRohPIFmXRl3RE"

# 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:
            # 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

```

TDDtest.py:

```
import unittest

from main import summa
from main import raznost
from main import proizv
from main import chastnoe

class TDDtestFunc(unittest.TestCase):
    def testFunc(self):
        self.assertEqual(summa(3, 2), 5)
        self.assertEqual(raznost(3, 2), 1)
        self.assertEqual(proizv(3, 2), 6)
        self.assertEqual(chastnoe(3, 2), 1.5)

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

BDDfeat.feature:

```
Feature: testing the function from main
    Scenario: testing the function chastnoe(3, 2)
        Given I put values [3, 2] into the function
        Then I get 1.5

    Scenario: testing the function chastnoe(3, 0)
        Given I put values [3, 0] into the function
        Then I get Error
```

BDDtest.py:

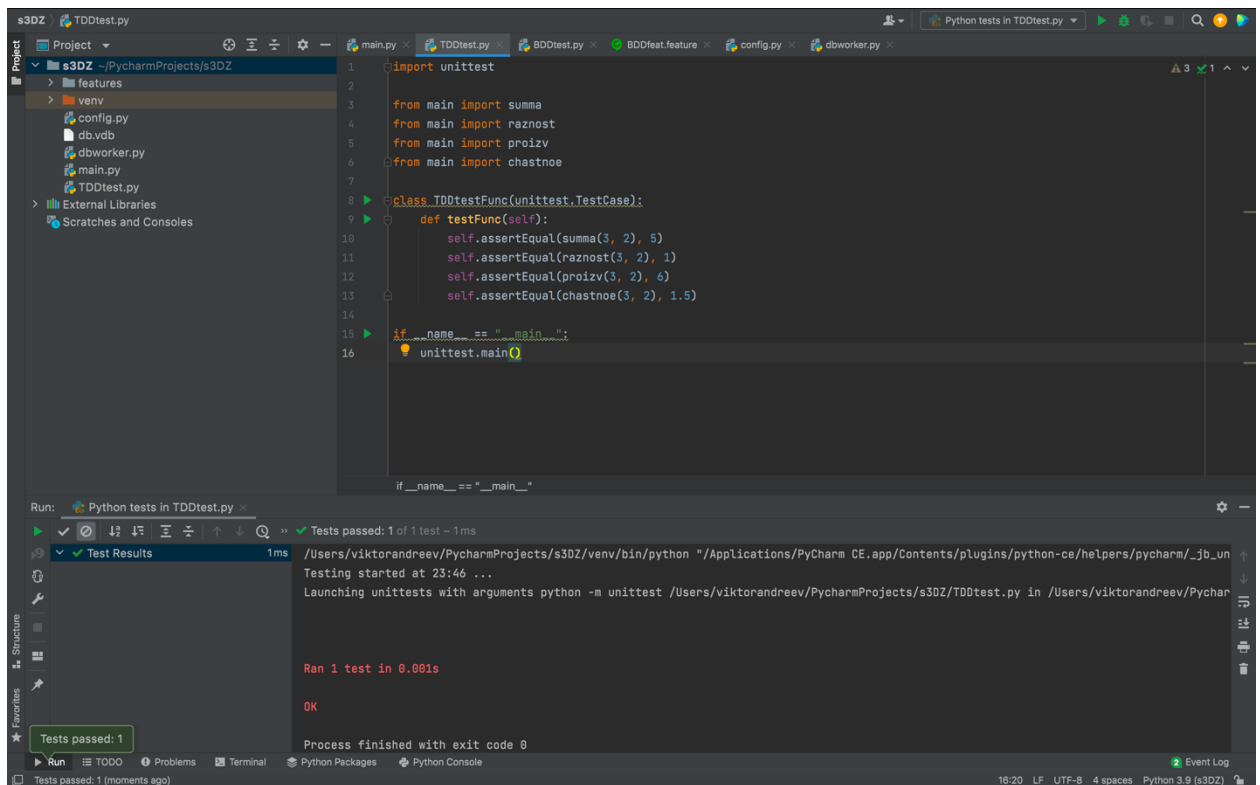
```
from behave import given, then
from main import chastnoe

@given('I put values {values} into the function')
def step_impl(context, values: str):
    values = list(map(int, values.replace("[", "").replace("]", "").split(", ")))
    context.result = chastnoe(values[0], values[1])

@then('I get {result}')
def step_impl(context, result: str):
```

assert str(context.result) == result

Примеры работы программы

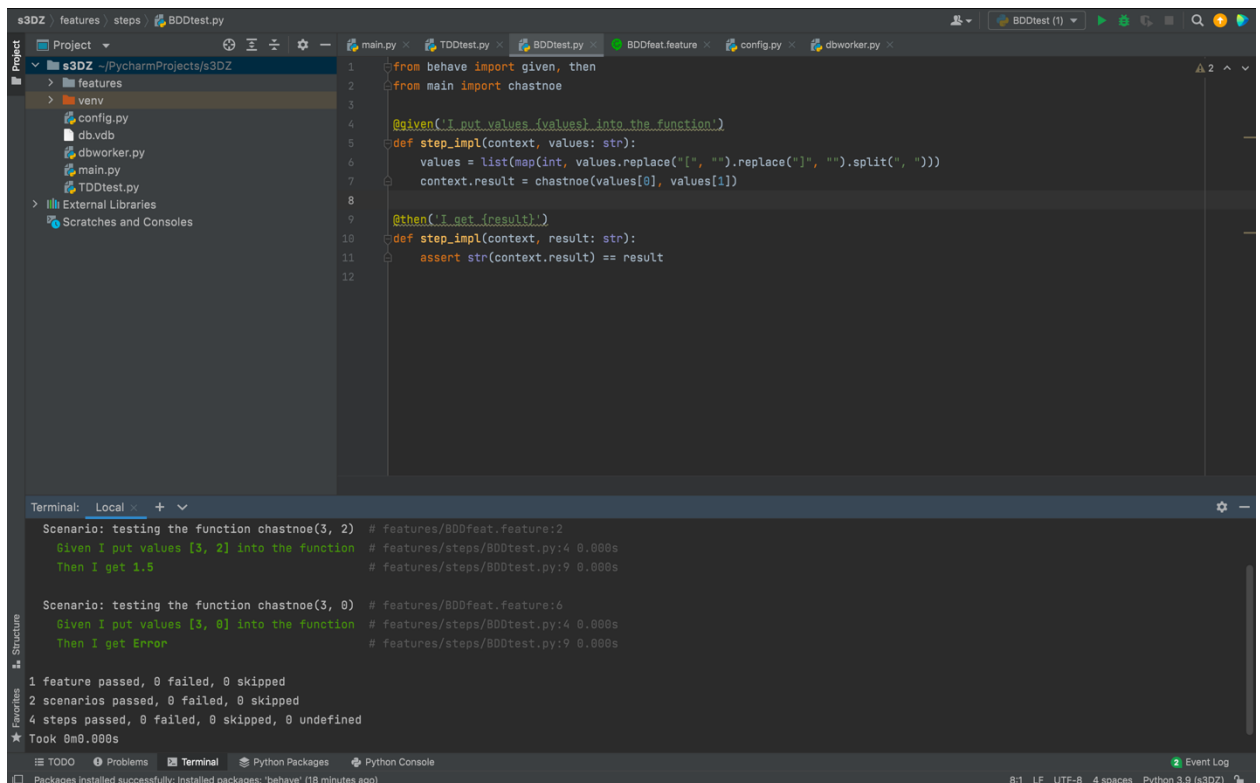


The screenshot shows the PyCharm IDE with a project named 's3DZ'. The file 'TDDtest.py' is open, containing the following code:

```
1 import unittest
2
3 from main import summa
4 from main import raznost
5 from main import proizv
6 from main import chastnoe
7
8 class TDDtestFunc(unittest.TestCase):
9     def testFunc(self):
10         self.assertEqual(summa(3, 2), 5)
11         self.assertEqual(raznost(3, 2), 1)
12         self.assertEqual(proizv(3, 2), 6)
13         self.assertEqual(chastnoe(3, 2), 1.5)
14
15 if __name__ == "__main__":
16     unittest.main()
```

The Run window at the bottom shows the test results:

```
Run: Python tests in TDDtest.py
Tests passed: 1 of 1 test - 1 ms
Testing started at 23:46 ...
Launching unittests with arguments python -m unittest /Users/viktorandreev/PycharmProjects/s3DZ/TDDtest.py in /Users/viktorandreev/Pychar
Ran 1 test in 0.001s
OK
Process finished with exit code 0
```



The screenshot shows the PyCharm IDE with a project named 's3DZ'. The file 'BDDtest.py' is open, containing the following code:

```
1 from behave import given, then
2 from main import chastnoe
3
4 @given('I put values {values} into the function')
5 def step_impl(context, values: str):
6     values = list(map(int, values.replace("[", "").replace("]", "").split(" ")))
7     context.result = chastnoe(values[0], values[1])
8
9 @then('I get {result}')
10 def step_impl(context, result: str):
11     assert str(context.result) == result
```

The Terminal window at the bottom shows the test results:

```
Terminal: Local
Scenario: testing the function chastnoe(3, 2) # features/BDDfeat.feature:2
  Given I put values [3, 2] into the function # features/steps/BDDtest.py:4 0.000s
  Then I get 1.5 # features/steps/BDDtest.py:9 0.000s

Scenario: testing the function chastnoe(3, 0) # features/BDDfeat.feature:6
  Given I put values [3, 0] into the function # features/steps/BDDtest.py:4 0.000s
  Then I get Error # features/steps/BDDtest.py:9 0.000s

1 feature passed, 0 failed, 0 skipped
2 scenarios passed, 0 failed, 0 skipped
4 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.000s
```