**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Домашнее задание

## по дисциплине «Базовые компоненты интернет-технологий»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33 |  | преподаватель каф. ИУ5 |
| Ефременко Д.С. |  | Канев А.И. |
| Подпись и дата: |  | Подпись и дата: |

2021

### 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:

HW.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):  
 name = message.from\_user.first\_name  
 bot.send\_message(message.chat.id, 'Приветствую тебя, '+ name)  
 bot.send\_message(message.chat.id, 'Я бот-калькулятор!')  
 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)  
 global res  
  
 if op == '/' and fv2 == 0:  
 bot.send\_message(message.chat.id, 'На ноль делить нельзя!')  
 dbworker.set(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE), config.States.STATE\_SECOND\_NUM.value)  
 bot.send\_message(message.chat.id, 'Введите второе число')  
 return  
 res = final(op, fv1, fv2)  
 # Выводим результат  
 markup = types.ReplyKeyboardRemove(selective=False)  
 bot.send\_message(message.chat.id, f'Результат: {v1}{op}{v2}={str(res)}', reply\_markup=markup)  
 markup = types.ReplyKeyboardMarkup(row\_width=2)  
 itembtn1 = types.KeyboardButton('MR')  
 itembtn2 = types.KeyboardButton('MC')  
 itembtn3 = types.KeyboardButton('M+')  
 itembtn4 = types.KeyboardButton('M-')  
 itembtn5 = types.KeyboardButton('MS')  
 itembtn6 = types.KeyboardButton('Продолжить вычисления')  
 markup.add(itembtn1, itembtn2, itembtn3, itembtn4, itembtn5, itembtn6)  
 bot.send\_message(message.chat.id, 'Выберите пожалуйста действие', reply\_markup=markup)  
 dbworker.set(dbworker.make\_key(message.chat.id, config.CURRENT\_STATE), config.States.STATE\_EXTRA.value)  
  
def final(op, fv1, fv2):  
 if op == '+':  
 ans = fv1 + fv2  
 elif op == '\*':  
 ans = fv1 \* fv2  
 elif op == '/' and fv2 != 0:  
 ans = fv1 / fv2  
 elif op == '-':  
 ans = fv1 - fv2  
 return ans  
  
@bot.message\_handler(func=lambda message: dbworker.get(  
 dbworker.make\_key(message.chat.id, config.CURRENT\_STATE)) == config.States.STATE\_EXTRA.value)  
def extra(message):  
 result = res  
 extra\_op = message.text  
 if extra\_op == 'MS':  
 dbworker.set(dbworker.make\_key(message.chat.id, config.States.STATE\_EXTRA.value), str(result))  
 bot.send\_message(message.chat.id, 'Число успешно сохранено')  
 elif extra\_op == 'MR':  
 v3 = dbworker.get(dbworker.make\_key(message.chat.id, config.States.STATE\_EXTRA.value))  
 bot.send\_message(message.chat.id, 'Число, сохраненное в памяти:' + v3)  
 elif extra\_op == 'MC':  
 result = 0  
 dbworker.set(dbworker.make\_key(message.chat.id, config.States.STATE\_EXTRA.value), result)  
 bot.send\_message(message.chat.id, 'Число удалено из памяти')  
 elif extra\_op == 'Продолжить вычисления':  
 # 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')  
 elif extra\_op == 'M-':  
 k1 = float(dbworker.get(dbworker.make\_key(message.chat.id, config.States.STATE\_EXTRA.value)))  
 result = k1 - res  
 bot.send\_message(message.chat.id, f'Результат: {k1}-{res}={str(result)}')  
 elif extra\_op == 'M+':  
 k1 = float(dbworker.get(dbworker.make\_key(message.chat.id, config.States.STATE\_EXTRA.value)))  
 result = k1 + res  
 bot.send\_message(message.chat.id, f'Результат: {k1}+{res}={str(result)}')  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 bot.infinity\_polling()

HW\_TDD.py

import unittest  
import sys, os  
  
sys.path.append(os.getcwd())  
from HW import \*  
  
class Calc\_Test(unittest.TestCase):  
 def test\_1(self):  
 self.assertEqual(final('+', -2, 1), -1)  
 def test\_2(self):  
 self.assertEqual(final('-', 5, 1), 4)  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

HW\_BDD.py

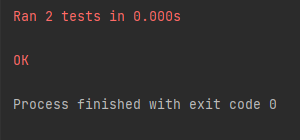
from behave import \*  
  
from HW\_TDD import \*  
  
@given('Bot')  
def first\_step(context):  
 context.a = Calc\_Test()  
  
@when('test\_1 return OK')  
def test\_1(context):  
 context.a.test\_1()  
  
@when('test\_2 return OK')  
def test\_2(context):  
 context.a.test\_2()  
  
@then('all right')  
def last\_step(context):  
 pass

myfeature.feature

Feature: Test  
 Scenario: Bot testing  
 Given Bot  
 When test\_1 return OK  
 And test\_2 return OK  
 Then all right

Результаты выполнения:

TDD:



BDD:

