



**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»**

**Курс «Базовые компоненты интернет-технологий»
Отчет по домашнему заданию**

«Домашнее задание»

**Выполнил:
студент группы ИУ5-33Б
Климов Н.С.**

**Проверил:
Канев А.И.**

2021 г.

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:

Файл main.py

```
# This Python file uses the following encoding: utf-8
import os
import telebot
from telebot import types
TOKEN = '5070861888:AAFyocuSFXNXASY1flsiW2IR2IBtyXnVs4o'
Serial = ["/Ужасы", "/Детектив"]
Film = ["/Комедия", "/Боевик"]
bot = telebot.TeleBot(TOKEN) # упрощаем обращение через API, чтобы каждый
раз не писать всю строку
markup = types.ReplyKeyboardMarkup() # табличка кнопочек под полем ввода
сообщения у юзера

@bot.message_handler(commands=['start'])
def handle_info(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Выберите сериал или фильм:')
    user_markup.row("/Сериал", "/Фильм")
    bot.send_message(message.chat.id, """Добро пожаловать:
    """,
                    reply_markup=user_markup) # ну и отправляем
пользователю сообщение, чтобы он знал, что все работает
    # в конце reply_markup = user_markup прикрепляем те самые кнопочки

@bot.message_handler(commands=['Сериал', 'Фильм'])
def handle_info(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    user_markup.row("/Сериал", "/Фильм") # первая строка (раз, два)
    bot.send_message(message.chat.id, "Пожалуйста нажмите кнопку",
                    reply_markup=user_markup)

@bot.message_handler(commands=['Сериал'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Выберите жанр:')
    user_markup.row("/Ужасы", "/Детектив")
    user_markup.row("/Сериал", "/Фильм")
    bot.send_message(message.chat.id, "Пожалуйста нажмите кнопку",
                    reply_markup=user_markup)

@bot.message_handler(commands=['Фильм'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Выберите жанр:')
    user_markup.row("/Комедия", "/Боевик")
    user_markup.row("/Сериал", "/Фильм")
    bot.send_message(message.chat.id, "Пожалуйста нажмите кнопку",
                    reply_markup=user_markup)
```

```

@bot.message_handler(commands=['Ужасы'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Выберете_сериал:')
    user_markup.row("/Королевы_Крика", "/Салем")
    user_markup.row("/Сериал")
    bot.send_message(message.chat.id, "Пожалуйста нажмите кнопку",
reply_markup=user_markup)

@bot.message_handler(commands=['Детектив'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Выберете_сериал:')
    user_markup.row("/Шерлок", "/Менталист")
    user_markup.row("/Сериал")
    bot.send_message(message.chat.id, "Пожалуйста нажмите кнопку",
reply_markup=user_markup)

@bot.message_handler(commands=['Комедия'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Выберете_фильм:')
    user_markup.row("/Американский_Пирог", "/Шпион")
    user_markup.row("/Фильм")
    bot.send_message(message.chat.id, "Пожалуйста нажмите кнопку",
reply_markup=user_markup)

@bot.message_handler(commands=['Боевик'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Выберете_фильм:')
    user_markup.row("/Нерв", "/Охота")
    user_markup.row("/Фильм")
    bot.send_message(message.chat.id, "Пожалуйста нажмите кнопку",
reply_markup=user_markup)

@bot.message_handler(commands=['Королевы_Крика'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, '«Королевы Крика» (англ: Scream Queens)
– американский телесериал канала FOX, созданный Райаном Мёрфи. Перед нами
слэшер, хорошо перемешанный с чёрной комедией и сатирой. Отчасти, это пародия
на жанр, даже название отсылает к понятию «Королева Крика», обозначающему
амплуа актрис хорроров. Среди самых известных можно выделить Нив Кемпбелл,
Кэтрин Изабель и Мэри-Элизабет Уинстэд. Одна из таких, Джейми Ли Кёртис,
звезда серии Хэллоуин, сыграла здесь одну из главных ролей.')
    user_markup.row("/Ужасы")
    bot.send_message(message.chat.id, "Конец описания",
reply_markup=user_markup)

@bot.message_handler(commands=['Салем'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, '«Салем» (англ. Salem) — американский
драматический телесериал в жанре исторического фэнтези, созданный Адамом
Саймоном и Брэнноном Брагой, который транслировался на телеканале WGN America
с 20 апреля 2014 года по 18 января 2017 года[1]. Это первое оригинальное шоу
WGN America[2]. Сериал, в котором главные роли исполняют Джанет Монтомери и
Шейн Уэст, вдохновлён в действительности происходившими судебными процессами
над салемскими ведьмами в конце XVII века.')
    user_markup.row("/Ужасы")
    bot.send_message(message.chat.id, "Конец описания",
reply_markup=user_markup)

```

```

@bot.message_handler(commands=['Шерлок'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Шерлок (Sherlock) — телесериал BBC, рассказывающий общеизвестную историю Шерлока Холмса на новый лад. Первый сезон, состоящий из трёх серий по 90 минут, стартовал в июле 2010 года. Второй — в январе 2012. Третий — в январе 2014. Спецвыпуск — в январе 2016. Четвёртый сезон — в январе 2017. Пятого сезона в планах нет, но формально шоу не закрывали.')
    user_markup.row("/Детектив")
    bot.send_message(message.chat.id, "Конец описания",
reply_markup=user_markup)

@bot.message_handler(commands=['Менталист'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, '«Менталист» (англ. The Mentalist) — американский телевизионный сериал, который транслировался на телеканале «CBS» с 23 сентября 2008 года по 18 февраля 2015 года. Создан английским сценаристом и режиссёром Бруно Хеллером, который также является исполнительным продюсером. Сериал транслировался на «CBS» на протяжении семи сезонов, насчитывая в общей сложности 151 эпизод')
    user_markup.row("/Детектив")
    bot.send_message(message.chat.id, "Конец описания",
reply_markup=user_markup)

@bot.message_handler(commands=['Американский Пирот'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, '«Американский пирот» (англ. American Pie) — серия комедийных фильмов. Первый фильм был выпущен 9 июля 1999 года студией Universal Pictures, став поп-культурным явлением. Затем вышли два продолжения (2001 и 2003 годах). С 2005 по 2009 вышли 4 спин-офф фильма. В 2012 году вышел четвёртый фильм из оригинальной серии. В дальнейшем неоднократно сообщалось о возможном выходе в будущем пятой части франшизы[1][2][3]. В 2020 году вышел новый спин-офф.')
    user_markup.row("/Ужасы")
    bot.send_message(message.chat.id, "Конец описания",
reply_markup=user_markup)

@bot.message_handler(commands=['Шпион'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, 'Сьюзан Купер всю свою жизнь мечтала стать секретным агентом и даже устроилась работать в ЦРУ. Однако, дальше сотрудника самой низкой ступени ей так и не удалось продвинуться. Понимая, что осуществить свою мечту практически невозможно, она все-таки продолжает надеяться на лучшее и с нетерпением ожидает своего шанса. И вскоре судьба дарит ей такую возможность. Лучший агент проваливает секретное задание, в ходе которого он должен был выяснить местонахождение ядерной бомбы, но, совершенно случайно, застрелил единственного человека, владеющего этой информацией. Через некоторое время службе разведки становится известно, что Рейна Боянова знает, где находится бомба, но она также знает в лицо всех секретных агентов ЦРУ. Теперь у руководства не остается другого выбора, как воспользоваться услугами Сьюзан Купер, но сможет ли она справиться со столь сложной миссией?..')
    user_markup.row("/Комедия")
    bot.send_message(message.chat.id, "Конец описания",
reply_markup=user_markup)

@bot.message_handler(commands=['Нерв'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)

```

```

        bot.send_message(message.chat.id, 'Поддавшись на уговоры одноклассниц, тихоня Ви открывает для себя мир популярной онлайн-игры «Нерв», где участники выполняют задания разной степени сложности, которые придумывают для них другие пользователи. В случае успеха игроки получают очень неплохие деньги. Ви проявляет неожиданную для самой себя смелость и отсутствие комплексов, и в очередном раунде знакомится с парнем по имени Иэн. Красивая пара нравится зрителям игры, и они решают, что следующие задания Ви и Иэн должны выполнять только вместе. Ставки растут, задания становятся все опаснее, и Ви постепенно понимает, что выйти из игры по собственной воле не получится. Те, кто считают себя игроками, на самом деле всего лишь пешки в руках таинственного сообщества Наблюдателей. И никто из участников игры не знает, какое задание станет следующим – или последним.')
        user_markup.row("/Боевик")
        bot.send_message(message.chat.id, "Конец описания",
reply_markup=user_markup)

@bot.message_handler(commands=['Охота'])
def handle_stop(message):
    user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
    bot.send_message(message.chat.id, '«Охота» (англ. The Hunt) – американский триллер[a] режиссёра Крэйга Зобела по сценарию Ника Кьюза и Деймона Линделофа, основанный на рассказе Ричарда Коннелла «Самая опасная дичь». В главных ролях: Бетти Гилпин, Айк Баринхолц, Эмма Робертс и Хилари Суэнк. Продюсером фильма стал Джейсон Блум со своей компанией Blumhouse Productions[15]. Зобел и Линделоф говорили, что фильм служит сатирой на глубокое политическое разделение между американскими левыми и правыми[5].')
    user_markup.row("/Боевик")
    bot.send_message(message.chat.id, "Конец описания",
reply_markup=user_markup)

try:
    bot.polling(none_stop=True, interval=0)
except IndexError:
    bot.polling(none_stop=True, interval=0)
except TypeError:
    bot.polling(none_stop=True, interval=0)

def check_Serial(text):
    if text in Serial:
        return 1
    else:
        return 0

def check_Film(text):
    if text in Film:
        return 1
    else:
        return 0

bot.infinity_polling()

```

Файл tdd.py

```

import unittest
import sys, os

sys.path.append(os.getcwd())
from main import *

```

```
class myTest(unittest.TestCase):
    def test_1(self):
        pass
        self.assertEqual(check_Serial("/Ужасы"), 1)
        self.assertEqual(check_Serial("/Мелодрамма"), 0)
    def test_2(self):
        pass
        self.assertEqual(check_Film("/Комедия"), 1)
        self.assertEqual(check_Film("/Артхаус"), 0)
```

Файл feature.feature

```
Feature: Test
  Scenario: Bot testing
    Given Bot
    When test_1 return OK
    And test_2 return OK
    Then all right
```

Файл bdd.py

```
from behave import *

from tdd import *

@given('Bot')
def first_step(context):
    context.a = myTest()

@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
```

Экранные формы с примерами выполнения программы:

Результаты работы tdd.py:



