



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»**

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

Выполнил:
студент группы
ИУ5Ц-52Б Дзауров И.А.
_____, “__” _____ 2022 г.

Проверил:
преподаватель кафедры
ИУ5 - Гапанюк Ю.Е.
_____, “__” _____ 2022 г.

Москва, 2022 г.

Описание задания

1. Модифицируйте код лабораторной работы №5 или №6 таким образом, чтобы он был пригоден для модульного тестирования.

Структура проекта:

```
main.py
TDDTest.py
```

Основной файл - [main.py](#):

```
-----
# This is a sample Python script.
import os
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton # pip install aiogram
from aiogram import Dispatcher, Bot, executor, types
from random import randint, choice
from string import ascii_letters, digits, punctuation

API_TOKEN = "___"

# инициализация бота...
bot = Bot(token=API_TOKEN)
dispatcher = Dispatcher(bot)

# Создание клавиатуры
btn_random = KeyboardButton("🎲 Рандомное число")
btn_pass = KeyboardButton('🔑 Придумать пароль')
btn_other = KeyboardButton("💠 Другое")
btn_photo = KeyboardButton("💠 Герб")

main_menu = ReplyKeyboardMarkup(resize_keyboard=True).add(btn_random,
btn_pass, btn_photo, btn_other)

btn_info = KeyboardButton('ℹ️ Информация')
btn_main = KeyboardButton('🟡 Главное меню')

other_menu = ReplyKeyboardMarkup(resize_keyboard=True).add(btn_info,
btn_main)

@dispatcher.message_handler(commands=['start'])
async def start(message: types.Message):
    await bot.send_message(message.from_user.id, f"👋 Привет, {message.from_user.first_name}!", reply_markup=main_menu)

@dispatcher.message_handler()
async def messages(message: types.Message):
    if message.text == '🎲 Рандомное число':
        await bot.send_message(message.from_user.id, f'🎲 Рандомное число: {randint(0, 100)}')
    elif message.text == '🔑 Придумать пароль':
        password = "".join([choice(str(digits + ascii_letters + punctuation)) for _ in range(24)])
        await bot.send_message(message.from_user.id, f'🔑 Надёжный пароль: {password}')
```

```

elif message.text == '◆ Гепб':
    text = 'simple_text'
    img_url = 'Telegram.png'
    await bot.send_message(message.from_user.id, f'{text}<a
href="{img_url}">.</a>', parse_mode=ParseMode.HTML)
elif message.text == '◆ Другое':
    await bot.send_message(message.from_user.id, '◆ Открываю...',
reply_markup=other_menu)
elif message.text == '📄 Информация':
    await bot.send_message(message.from_user.id, f'👤 Ваше имя:
{message.from_user.first_name}\n\nБот создан by Ibragim 🤖')
elif message.text == '🟡 Главное меню':
    await bot.send_message(message.from_user.id, '🟡 Открываю меню...',
reply_markup=main_menu)
else:
    await bot.send_message(message.from_user.id, f'😞 Бот вас не понял...
:(')

if __name__ == '__main__':
    executor.start_polling(dispatcher, skip_updates=True)

```

TDD-тест - TDDTest.py:

```

import unittest

from main import get_roots, get_roots_biquadratic

class TestGetRoots(unittest.TestCase):
    def testGetRoots(self):
        self.assertEqual(get_roots_biquadratic(get_roots(4, -5, 1)), [1.0, -1.0,
0.5, -0.5])
        self.assertEqual(get_roots_biquadratic(get_roots(1, -2, -8)), [2.0, -
2.0])
        self.assertEqual(get_roots_biquadratic(get_roots(1, 1, 1)), [])

    def testValue(self):
        with self.assertRaises(ValueError) as e:
            get_roots_biquadratic(get_roots(0, 33, 9))

    def testType(self):
        with (self.assertRaises(TypeError)) as e:
            get_roots_biquadratic(get_roots(7, "D", 4))

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

```

Экранные формы с примерами выполнения программы

