

Московский государственный технический университет им. Н.Э. Баумана

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Базовые компоненты интернет-технологий»
Отчет по лабораторной работе №5**

Выполнил:
студент группы ИУ5-31Б
Егошин Дмитрий
Павлович

Подпись: _____

Дата: _____

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий
Евгеньевич

Подпись: _____

Дата: _____

Москва, 2021 г.

Лабораторная работа №5

Описание задания

1. Необходимо для произвольной предметной области реализовать от одного до трех шаблонов проектирования: один порождающий, один структурный и один поведенческий. Для сдачи лабораторной работы в минимальном варианте достаточно реализовать один паттерн.
2. Вместо реализации паттерна Вы можете написать тесты для своей программы решения биквадратного уравнения. В этом случае, возможно, Вам потребуется доработать программу решения биквадратного уравнения, чтобы она была пригодна для модульного тестирования.
3. В модульных тестах необходимо применить следующие технологии:
 - TDD - фреймворк.
 - BDD - фреймворк.
 - Создание Mock-объектов.

Текст программы

1.main.py

```
import telebot
from telebot import types

TOKEN = '5041319492:AAGrjQOxJ9etN8CmymAKWcnE9CcqKcVy42o'
bot = telebot.TeleBot(TOKEN)
global config
config = ['Первое число', 'Второе число', 'Действие', 'Посчитать']
global cases
cases = ['first', 'second', 'action', 'res']
global call
call = ''
global actions
actions = ['plus', 'minus', 'multiply', 'divide']

@bot.message_handler(commands='start')
def start(message):
    msg = 'Добро пожаловать.'
    markup = types.InlineKeyboardMarkup()
    btn = types.InlineKeyboardButton('Начать работу', callback_data='work')
    markup.add(btn)
    bot.send_message(message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == 'work')
def work(message):
    msg = 'Введите данные'
    markup = types.InlineKeyboardMarkup(row_width=1)
    for i in range(4):
        btn = types.InlineKeyboardButton(f'{config[i]}',
        callback_data=f'{cases[i]}')
```

```

        markup.add(btn)
        btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
        markup.add(btn)
        bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == cases[0] or
message.data == cases[1])
def nums(message):
    global call
    call = cases[cases.index(message.data)]
    msg = 'Введите число'
    bot.send_message(message.message.chat.id, msg)

@bot.callback_query_handler(lambda message: message.data == cases[2])
def action(message):
    msg = 'Выберите действие'
    markup = types.InlineKeyboardMarkup(row_width=2)
    btn = types.InlineKeyboardButton('+', callback_data='plus')
    btn1 = types.InlineKeyboardButton('-', callback_data='minus')
    btn2 = types.InlineKeyboardButton('*', callback_data='multiply')
    btn3 = types.InlineKeyboardButton('/', callback_data='divide')
    markup.add(btn, btn1, btn2, btn3)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data in actions)
def act(message):
    if message.data == actions[0]:
        config[2] = actions[0]
    elif message.data == actions[1]:
        config[2] = actions[1]
    elif message.data == actions[2]:
        config[2] = actions[2]
    elif message.data == actions[3]:
        config[2] = actions[3]
    markup = types.InlineKeyboardMarkup(row_width=1)
    msg = 'Введите данные'
    for i in range(4):
        if not config[i].isdigit() and not config[i] in actions:
            btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
            markup.add(btn)
    btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
    markup.add(btn)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == cases[3])
def res(message):
    if config[0].isdigit() and config[1].isdigit() and (config[2] in
actions):
        if config[2] == 'plus':
            config[3] = float(config[0]) + float(config[1])
            msg = f'{float(config[0])} + {float(config[1])} = {config[3]}'
        elif config[2] == 'minus':
            config[3] = float(config[0]) - float(config[1])
            msg = f'{float(config[0])} - {float(config[1])} = {config[3]}'
        elif config[2] == 'multiply':
            config[3] = float(config[0]) * float(config[1])
            msg = f'{float(config[0])} * {float(config[1])} = {config[3]}'
        elif config[2] == 'divide':

```

```

        if config[1] == '0':
            msg = 'Деление на ноль невозможно.'
        else:
            config[3] = float(config[0]) / float(config[1])
            msg = f'{float(config[0])} / {float(config[1])} =
{config[3]}'
            markup = types.InlineKeyboardMarkup()
            btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
            markup.add(btn)
            bot.send_message(message.message.chat.id, msg, reply_markup=markup)
        else:
            msg = 'Недостаточно данных'
            markup = types.InlineKeyboardMarkup()
            for i in range(4):
                if not config[i].isdigit() and not config[i] in actions:
                    btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
                    markup.add(btn)
            btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
            markup.add(btn)
            bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.callback_query_handler(lambda message: message.data == 'reset')
def reset(message):
    msg = 'Данные сброшены.'
    global config
    config = ['Первое число', 'Второе число', 'Действие', 'Посчитать']
    markup = types.InlineKeyboardMarkup()
    btn = types.InlineKeyboardButton('Продолжить', callback_data='work')
    markup.add(btn)
    bot.send_message(message.message.chat.id, msg, reply_markup=markup)

@bot.message_handler(content_types='text')
def text(message):
    if (call == cases[0] or call == cases[1]) and message.text.isdigit():
        if call == cases[0]:
            config[0] = message.text
            bot.send_message(message.chat.id, f'Вы ввели первое число
{float(message.text)}')
        elif call == cases[1]:
            config[1] = message.text
            bot.send_message(message.chat.id, f'Вы ввели второе число
{float(message.text)}')
        markup = types.InlineKeyboardMarkup(row_width=1)
        msg = 'Введите данные'
        for i in range(4):
            if not config[i].isdigit() and not config[i] in actions:
                btn = types.InlineKeyboardButton(f'{config[i]}',
callback_data=f'{cases[i]}')
                markup.add(btn)
        btn = types.InlineKeyboardButton('Сбросить', callback_data='reset')
        markup.add(btn)
        bot.send_message(message.chat.id, msg, reply_markup=markup)
    if (call == cases[0] or call == cases[1]) and not message.text.isdigit():
        msg = 'Ошибка, введите число'
        bot.send_message(message.chat.id, msg)

bot.polling(none_stop=True)

```

Результат выполнения работы



