Московский государственный технический университет им. Н.Э. Баумана

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

Курс «Базовые компоненты интернет-технологиий»

Лабораторная работа №6.

Выполнил:
студент группы ИУ5-34Б:
Гордеев Матвей Владиславович
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий Евгеньевич
Подпись и дата:

2022

Задание:

1. Модифицируйте код лабораторной работы №5 или №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).

Решение задачи

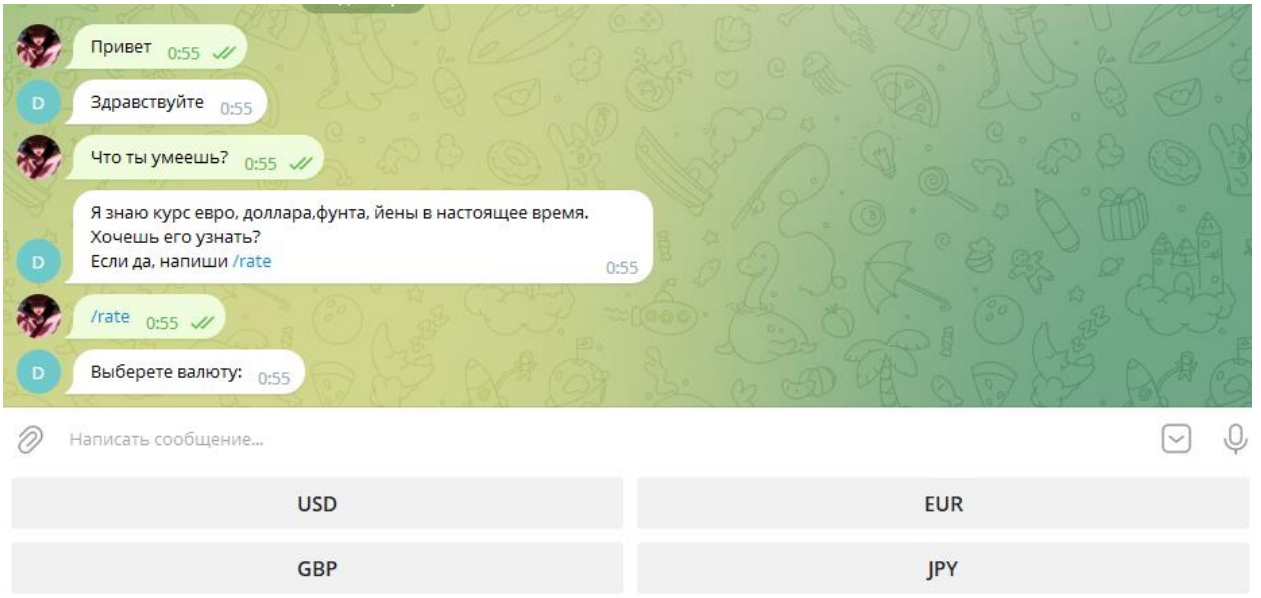
bot.py

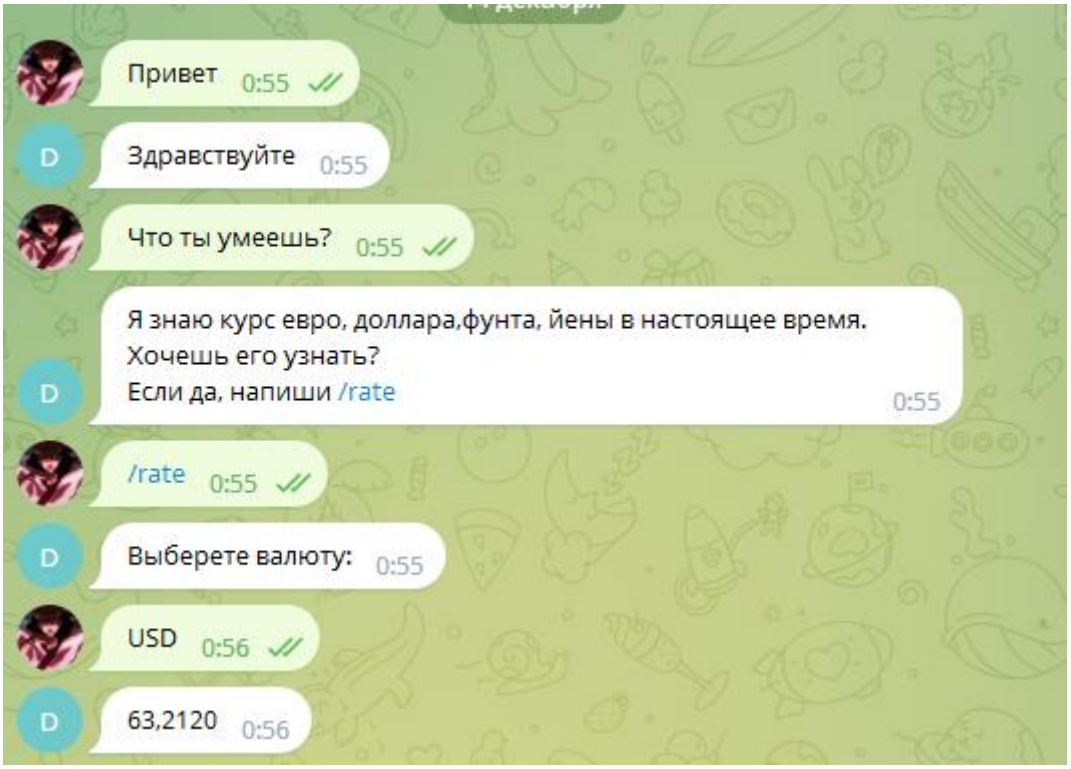
```
import telebot
import random
import requests
import datetime
import pytz
from telebot import types
from bs4 import BeautifulSoup
TOKEN= 'Скрыт для сдачи отчёта'

bot=telebot.TeleBot(TOKEN)


response = requests.get("https://www.cbr.ru/scripts/XML_daily_eng.asp")



soup=BeautifulSoup(response.text,"html.parser")


quests1=['Как дела?','как дела?','Как твои дела?','как твои дела?','Ты
как?','ты как?','Как сама?','как сама?','Как сам?','как сам?','Как
поживаешь?','как поживаешь?','Как настроение?','как настроение?']
answers1=["Отлично!","Замечательно!","Превосходно!","Лучше
некуда!","Хорошо..."]
quests2=['Что делаешь?','что делаешь?','че делаешь?','Че делаешь?','чем
занимаешься?','Чем занимаешься?','чё делаешь?','Чё делаешь?','чо
делаешь?','Чо делаешь?','чё делаеш?','Чё делаеш?','чо делаеш?','Чо делаеш?']
answers2=["Учусь","Отдыхаю","Тренеруюсь","С тобой общаюсь"]
quests3=['Назови своё имя','как тебя зовут?','кто ты?',"назови своё имя",'Как
тебя зовут?','Кто ты?','Ты кто?','ты кто?']
randans=['Не понимаю тебя, прости','Затрудняюсь сказать что-либо на этот
счёт',"Пожалуйста, переформулирой свой вопрос!"]
randans2=["Я знаю курс евро, доллара,фунта, йены в настоящее время. Хочешь
его узнать?\nЕсли да, напиши /rate","Тебе интересен курс евро, доллара,фунта,
йены в настоящее время?\nЕсли да, напиши /rate"]
randoms=[randans,randans2]
answers4=['Ок','Ладно','Ну ладно','Окей']
quests5=["Привет","привет","Хай","хай","Здарова","здарова","Добрый
день","добрый день","Добрый вечер","добрый
вечер","Здравствуйте","здравствуйте","здравствуй","Здравствуй",
          "Сап","сап","Hi","hi","whats up","Whats up","Приветик","приветик"]
answers5=["Привет","привет","Хай","хай","Здравствуйте","здравствуйте","здравс
твуй","Здравствуй","Приветик","приветик"]
quests6=["Что будешь делать?","что будешь делать?"]
answers6=["Буду отдыхать",'Пойду погуляю','Посмотрю сериал']
quests7=['Что ты умеешь?','что ты умеешь?','что умеешь?','Что умеешь?','Что
ты знаешь?','что ты знаешь?']
quests8=['Пока','пока','Чао','чао','До встречи','до
встречи','Досвидания','досвидания','До свидания','до
```

```
свидания','Досвидание','досвидание']
answers8=['Пока','Чао','До встречи']



@bot.message_handler(commands=['start'])
def welcome(message):
    bot.send_message(message.chat.id,"Привет, {0.first_name}.\nМеня зовут
{1.first_name}.\n".format(message.from_user,bot.get_me()))



@bot.message_handler(commands=['help'])
def help_command(message):
    bot.send_message(message.chat.id,'Ты можешь узнать у меня курс
валют(/rate) или же можем просто поговорить. Если ты забыл телефон или часы,
могу подсказать точное время(/time)')


@bot.message_handler(commands=['time'])
def time_command(message):
    tz=pytz.timezone('Europe/Moscow')
    date = datetime.datetime.now(tz)
    bot.send_message(message.chat.id,"Точное время по Мск: ")
    bot.send_message(message.chat.id,date.strftime('%H:%M'))




@bot.message_handler(commands=['rate'])
def rates(message):
    markup=types.ReplyKeyboardMarkup(resize_keyboard=True,row_width=2)
    item1=types.KeyboardButton('USD')
    item2 = types.KeyboardButton('EUR')
    item3 = types.KeyboardButton('GBP')
    item4 = types.KeyboardButton('JPY')
    markup.add(item1,item2,item3,item4)
    newmsg=bot.send_message(message.chat.id,'Выберете валюту:
',reply_markup=markup)

    bot.register_next_step_handler(newmsg, process_coin_step)


def printCoin(buy):
    return '*Курс к рублю:* '+str(buy)
def process_coin_step(message):
    markup=types.ReplyKeyboardRemove(selective=False)
    if(message.text=='USD'):
        full_page = soup.find(id='R01235').value.string
    elif (message.text == 'EUR'):
        full_page = soup.find(id='R01239').value.string
    elif (message.text == 'GBP'):
        full_page = soup.find(id='R01035').value.string
    elif (message.text == 'JPY'):
        full_page = soup.find(id='R01820').value.string

    bot.send_message(message.chat.id, full_page, reply_markup=markup,
parse_mode="Markdown")


@bot.message_handler(content_types=['text'])
def noname(message):
    flag=0
    if flag==0 and (((message.text.count('Сколько')>0 or
```

```
message.text.count('сколько')>0) and (message.text.count('время')>0 or
message.text.count('Время')>0 or message.text.count('Времени')>0 or
message.text.count('времени')>0)) or message.text.count('который час')>0 or
message.text.count('Который час'))>0:
       tz = pytz.timezone('Europe/Moscow')
       date = datetime.datetime.now(tz)
       bot.send_message(message.chat.id,'Точное время ')
       bot.send_message(message.chat.id,date.strftime('%H:%M'))

    for i in range(0,len(quests1)):
        if message.text==quests1[i]:
            bot.send_message(message.chat.id,random.choice(answers1))
            flag=1
    for j in range(0,len(quests2)):
        if message.text==quests2[j]:
            bot.send_message(message.chat.id,random.choice(answers2))
            flag=1
    for h in range(0,len(quests3)):
        if message.text==quests3[h]:
            bot.send_message(message.chat.id,
                             "Запоминай, {0.first_name}.\nМеня зовут
{1.first_name}.\n".format(message.from_user,

bot.get_me()))
            bot.send_message(message.chat.id, random.choice(randans2))
            flag=1

    for y in range(0,len(quests7)):
        if message.text==quests7[y]:
            bot.send_message(message.chat.id,random.choice(randans2))
            flag=1

    for u in range(0,len(quests6)):
        if message.text==quests6[u]:
            bot.send_message(message.chat.id,random.choice(answers6))
            flag=1

    for g in range(0,len(quests5)):
        if message.text==quests5[g]:
            bot.send_message(message.chat.id,random.choice(answers5))
            flag=1
    for r in range(0,len(quests8)):
        if message.text==quests8[r]:
            bot.send_message(message.chat.id,random.choice(answers8))
            flag=1

    if message.text.count('хаха')>0 and flag==0:
        bot.send_message(message.chat.id,'Хех)')
        flag=1

    elif flag==0 and ((message.text=='Нет' or message.text=='нет' or
message.text=='да' or message.text=='Да')):
        bot.send_message(message.chat.id,random.choice(answers4))

    elif flag==0 and ((message.text=='Кто твой создатель?')):
        bot.send_message(message.chat.id,'Гордеев Матвей, группа ИУ5-34Б')

bot.polling(none_stop=True)
```

Результат


Привет 0:55
Здравствуйте 0:55
Что ты умеешь? 0:55
Я знаю курс евро, доллара,фунта, йены в настоящее время.
Хочешь его узнать?
Если да, напиши /rate 0:55
/rate 0:55
Выберете валюту: 0:55
Написать сообщение...
USD
EUR
GBP
JPY


Привет 0:55
Здравствуйте 0:55
Что ты умеешь? 0:55
Я знаю курс евро, доллара,фунта, йены в настоящее время.
Хочешь его узнать?
Если да, напиши /rate 0:55
/rate 0:55
Выберете валюту: 0:55
USD 0:56
63,2120 0:56

Bot.py(с сохранением состояния)

```
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 = fv1 + fv2
    elif op == '*':
        res = fv1 * fv2
    elif op=="-":
        res=fv1-fv2
    elif op=="/":
        res=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)
    # 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')


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

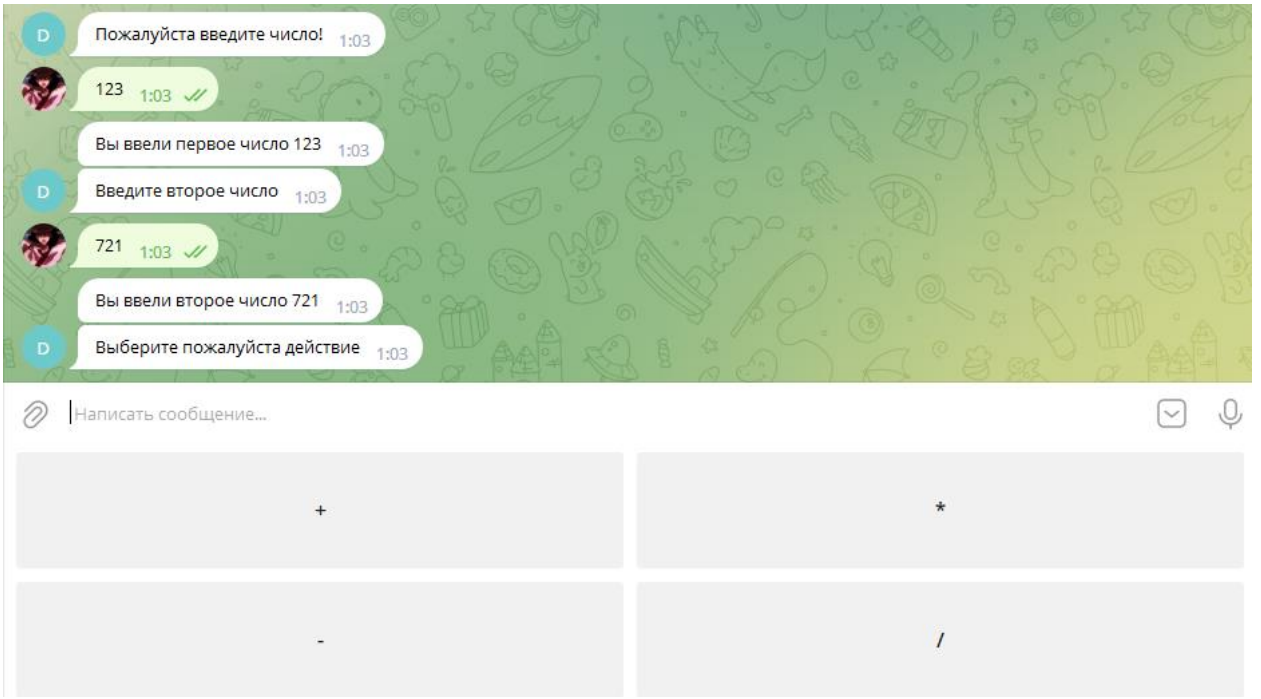
config.py

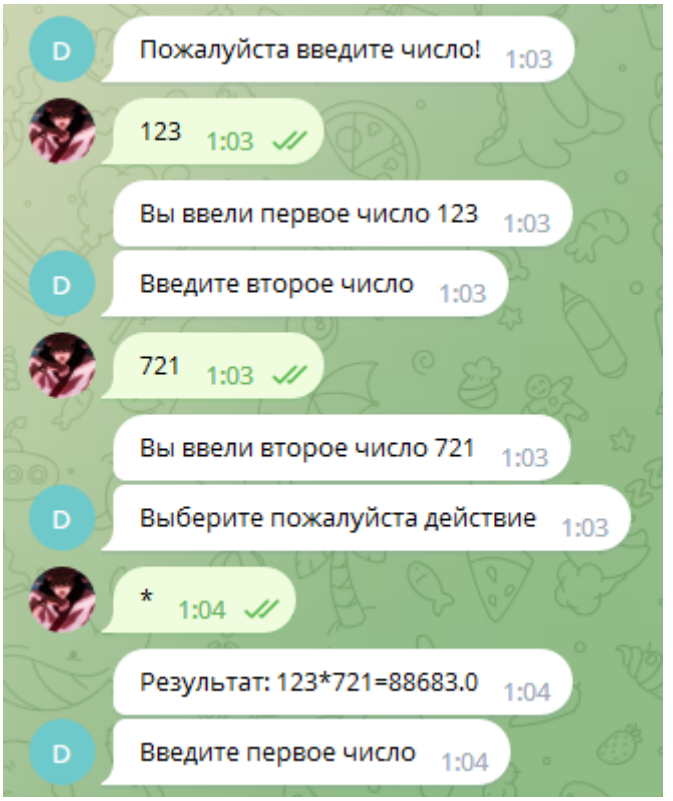
```
from enum import Enum

# Токент бота
TOKEN = "Скрыт для представления отчёта"

# 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.STATE_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
```

Результат


Пожалуйста введите число! 1:03
123 1:03
Вы ввели первое число 123 1:03
Введите второе число 1:03
721 1:03
Вы ввели второе число 721 1:03
Выберите пожалуйста действие 1:03
Написать сообщение...
+
*
-
/


Пожалуйста введите число! 1:03
123 1:03
Вы ввели первое число 123 1:03
Введите второе число 1:03
721 1:03
Вы ввели второе число 721 1:03
Выберите пожалуйста действие 1:03
* 1:04
Результат: 123*721=88683.0 1:04
Введите первое число 1:04