

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 «Системы обработки информации и управления»

Домашнее задание

по дисциплине «Базовые компоненты интернет-технологий» «Изучение возможностей создания ботов в Telegram и их тестирования.

>>

Выполнил:

Студент группы ИУ5-35Б

Титов Е.А.

Постановка задачи

- 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

```
def operation(fv1, fv2, op):
    # Выполняем действие
    res = 0
    if op == 'Перевести':
        res = fv1*fv2
    elif \ op == 'Стоимость комбика gibson в евро':
        res = int(500.0/fv2)
    return res
def num_input(text):
    cond = False
        a = float(text)
    except ValueError:
        cond = True
    if not cond and a <= 0:</pre>
        cond = True
    if cond:
        return 'CURRENT_STATE'
        return 'STATE COURSE'
```

Файл test.py

Файл dbworker.py

```
from vedis import Vedis
import config
# Чтение значения
def get(key):
    with Vedis(config.db_file) as db:
            return db[key].decode()
        except KeyError:
            # в случае ошибки значение по умолчанию - начало диалога
            return config.States.S_START.value
# Запись значения
def set(key, value):
    with Vedis(config.db_file) as db:
            db[key] = value
            return True
        except:
            return False
# Создание ключа для записи и чтения
def make_key(chatid, keyid):
    res = \underline{str}(chatid) + '\_' + \underline{str}(keyid)
    return res
```

Файл config.py

```
# Токент бота

TOKEN = "5009819170:AAHKKrGMlaGEiGuwCxH03Y_1G6qa8ZGuvwQ"

# 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_EUR = "STATE_EUR"
```

```
STATE_COURSE = "STATE_COURSE"

STATE CONV = "STATE CONV"
```

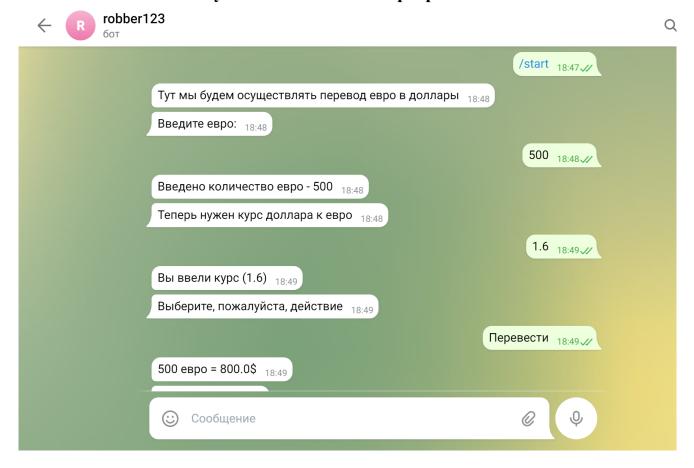
Файл bot.py

```
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):
    bot.send_message(message.chat.id, 'Тут мы будем осуществлять перевод евро в
доллары')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_EUR.value)
    bot.send_message(message.chat.id, 'Введите евро:')
# 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, 'Сброс результатов предыдущего ввода.')
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_EUR.value)
    bot.send_message(message.chat.id, 'Введите евро:')
# Обработка первого числа
@bot.message_handler(func=Lambda message: dbworker.get(
    dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.STATE_EUR.value)
def first_num(message):
    text = message.text
    cond = False
        a = float(text)
    except ValueError:
        cond = True
    if not cond and a <= 0:</pre>
        cond = True
    if cond:
        # Состояние не изменяется, выводится сообщение об ошибке
```

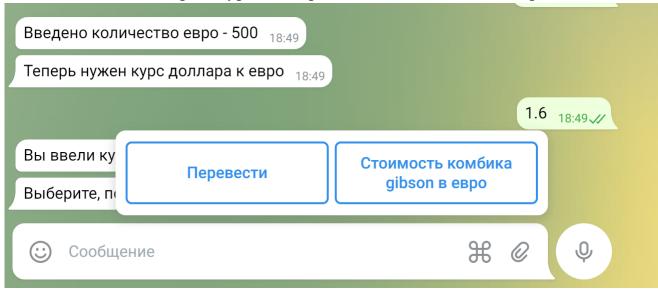
```
bot.send message(message.chat.id, 'Введите положительное число:')
        bot.send_message(message.chat.id, f'Введено количество евро - {text}')
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_COURSE.value)
        dbworker.set(dbworker.make key(message.chat.id,
config.States.STATE_EUR.value), text)
        bot.send_message(message.chat.id, 'Теперь нужен курс доллара к евро')
# Обработка второго числа
@bot.message_handler(func=Lambda message: <u>dbworker</u>.get(
    dbworker.make key(message.chat.id, config.CURRENT STATE)) ==
config.States.STATE COURSE.value)
def second_num(message):
   text = message.text
    cond = False
        a = float(text)
    except ValueError:
        cond = True
    if not cond and a <= 0:</pre>
        cond = True
    if cond:
        # Состояние не изменяется, выводится сообщение об ошибке
        bot.send_message(message.chat.id, 'Введите положительное число:')
        bot.send_message(message.chat.id, f'Вы ввели курс ({text})')
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE CONV.value)
        # Сохраняем первое число
        dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_COURSE.value), text)
        markup = types.ReplyKeyboardMarkup(row_width=2)
        itembtn1 = types.KeyboardButton('Перевести')
        itembtn2 = types. KeyboardButton('Стоимость комбика gibson в евро')
        markup.add(itembtn1, itembtn2)
        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 CONV.value)
def operation(message):
   # Текущее действие
    op = message.text
   # Читаем операнды из базы данных
```

```
val1 = dbworker.get(dbworker.make key(message.chat.id,
config.States.STATE_EUR.value))
    val2 = dbworker.get(dbworker.make_key(message.chat.id,
config.States.STATE COURSE.value))
    # Выполняем действие
    fv1 = float(val1)
    fv2 = float(val2)
    res = 0
    if op == 'Перевести':
        res = fv1*fv2
    elif op == 'Стоимость комбика gibson в евро':
        res = 500.0/fv2
    # Выводим результат
    markup = types.ReplyKeyboardRemove(selective=False)
    if op == 'Перевести':
        bot.send_message(message.chat.id, f'{val1} eBpo = {res}$',
reply_markup=markup)
        bot.send_message(message.chat.id, f'Komбик стоит примерно {int(res)} евро',
reply_markup=markup)
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.STATE_EUR.value)
    bot.send_message(message.chat.id, 'Введите евро')
bot.polling()
```

Результат выполнения программы



После ввода кол-ва евро и курса бот предлагает действие на выбор:



теперь нужен курс доллара к евро	18:49
	1.6 18:49
Вы ввели курс (1.6) _{18:49}	
Выберите, пожалуйста, действие	18:49
	Стоимость комбика gibson в евро _{18:50}
Комбик стоит примерно 312 евро	18:50
Введите евро 18:50	
Сообщение	© Q

1.7			
Ran 2 tests	in 0.000s		
ок			