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)

**Отчет по домашнему заданию по курсу**

**Базовые компоненты интернет-технологий**

**«Тестирование Telegram бота»**

|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| студент группы ИУ5-34Б |  |
| Братухин Александр |  |
| Подпись и дата:  13.12.21 | Подпись и дата: |

Москва, 2021

**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 **config.py**

from enum import Enum

TOKEN = токен, полученный от @BotFather

db\_file = 'db.vdb'

S\_CUR = "CURRENT"

class States(Enum):

    S\_START = "START"

    S\_TYPE = "TYPE"

    S\_PRICE = "PRICE"

    S\_COUNTRY = "COUNTRY"

Файл **dbworker.py**

from vedis import Vedis

import config

def get(key):

    with Vedis(config.db\_file) as db:

        try:

            return db[key].decode()

        except KeyError:

            return config.States.S\_START.value

def set(key, value):

    with Vedis(config.db\_file) as db:

        try:

            db[key] = value

            return True

        except:

            return False

def make\_key(id, st):

    return f"{id}\_{st}"

Файл **functions.py**

import os

cur\_path = os.path.dirname(os.path.abspath(\_\_file\_\_))

def path\_picture(type, price, country):

    if type == 'Грузовой':

        if price == 'Бюджетный':

            if country == 'Отечественный':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\до 1.2 млн Бюджет\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\до 1.2 млн Бюджет\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\до 1.2 млн Бюджет\\3.jpeg')

        elif price == 'Средний':

            if country == 'Отечественный':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\1.2-3 средний\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\1.2-3 средний\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\1.2-3 средний\\3.jpg')

        elif price == 'Престиж':

            if country == 'Отечественный':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\3 и больше (престиж)\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\3 и больше (престиж)\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Грузовые\\3 и больше (престиж)\\3.jpg')

    elif type == 'Семейный':

        if price == 'Бюджетный':

            if country == 'Отечественный':

                return os.path.join(cur\_path, 'бот Автомобили\Семейные\\до 1.2 млн Бюджет\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Семейные\\до 1.2 млн Бюджет\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Семейные\\до 1.2 млн Бюджет\\3.jpg')

        elif price == 'Средний':

            if country == 'Отечественный':

               return os.path.join(cur\_path, 'бот Автомобили\Семейные\\1.2-3 средний\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Семейные\\1.2-3 средний\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Семейные\\1.2-3 средний\\3.jpg')

        elif price == 'Престиж':

            if country == 'Отечественный':

                return os.path.join(cur\_path, 'бот Автомобили\Семейные\\3 и больше (престиж)\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Семейные\\3 и больше (престиж)\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Семейные\\3 и больше (престиж)\\3.jpg')

    elif type == 'Спортивный':

        if price == 'Бюджетный':

            if country == 'Отечественный':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\до 1.2 млн Бюджет\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\до 1.2 млн Бюджет\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\до 1.2 млн Бюджет\\3.jpg')

        elif price == 'Средний':

            if country == 'Отечественный':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\\1.2-3 средний\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\\1.2-3 средний\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\\1.2-3 средний\\3.jpg')

        elif price == 'Престиж':

            if country == 'Отечественный':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\\3 и больше (престиж)\\1.jpg')

            elif country == 'Европейский':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\\3 и больше (престиж)\\2.jpg')

            elif country == 'Азиатский':

                return os.path.join(cur\_path, 'бот Автомобили\Спортивные\\3 и больше (престиж)\\3.jpg')

Файл **bot.py**

import telebot

from telebot import types

import config

import dbworker

import functions as functions

bot = telebot.TeleBot(config.TOKEN)

@bot.message\_handler(commands=['start'])

def welcom(message):

    markup = types.ReplyKeyboardMarkup(row\_width=1, resize\_keyboard=True)

    markup.add(types.KeyboardButton('Привет, бот Автомобили!'))

    dbworker.set(dbworker.make\_key(message.chat.id, config.S\_CUR), config.States.S\_TYPE.value)

    bot.send\_message(message.chat.id, 'Добро пожаловать в бот Автомобили!', reply\_markup=markup)

@bot.message\_handler(func=lambda message:

dbworker.get(dbworker.make\_key(message.chat.id, config.S\_CUR)) == config.States.S\_TYPE.value)

def type\_car(message):

    if message.text == 'Грузовой' or message.text == 'Семейный' or message.text == 'Спортивный':

        if message.text == 'Грузовой':

            dbworker.set(dbworker.make\_key(message.chat.id, config.States.S\_TYPE.value), 'Грузовой')

            dbworker.set(dbworker.make\_key(message.chat.id, config.S\_CUR), config.States.S\_PRICE.value)

        elif message.text == 'Семейный':

            dbworker.set(dbworker.make\_key(message.chat.id, config.States.S\_TYPE.value), 'Семейный')

            dbworker.set(dbworker.make\_key(message.chat.id, config.S\_CUR), config.States.S\_PRICE.value)

        elif message.text == 'Спортивный':

            dbworker.set(dbworker.make\_key(message.chat.id, config.States.S\_TYPE.value), 'Спортивный')

            dbworker.set(dbworker.make\_key(message.chat.id, config.S\_CUR), config.States.S\_PRICE.value)

        markup1 = types.ReplyKeyboardMarkup(row\_width=3, resize\_keyboard=True)

        botton1\_1 = types.KeyboardButton('Бюджетный')

        botton2\_2 = types.KeyboardButton('Средний')

        botton3\_3 = types.KeyboardButton('Престиж')

        markup1.add(botton1\_1, botton2\_2, botton3\_3)

        bot.send\_message(message.chat.id, 'Выберите ценовой диапазон: ', reply\_markup=markup1)

    else:

        markup = types.ReplyKeyboardMarkup(row\_width=3, resize\_keyboard=True)

        botton1 = types.KeyboardButton('Грузовой')

        botton2 = types.KeyboardButton('Семейный')

        botton3 = types.KeyboardButton('Спортивный')

        markup.add(botton1, botton2, botton3)

        bot.send\_message(message.chat.id, 'Выберите тип авто: ', reply\_markup=markup)

@bot.message\_handler(func=lambda message:

dbworker.get(dbworker.make\_key(message.chat.id, config.S\_CUR)) == config.States.S\_PRICE.value)

def car\_price(message):

    if message.text == 'Бюджетный' or message.text == 'Средний' or message.text == 'Престиж':

        if message.text == 'Бюджетный':

            dbworker.set(dbworker.make\_key(message.chat.id, config.States.S\_PRICE.value), 'Бюджетный')

            dbworker.set(dbworker.make\_key(message.chat.id, config.S\_CUR), config.States.S\_COUNTRY.value)

        elif message.text == 'Средний':

            dbworker.set(dbworker.make\_key(message.chat.id, config.States.S\_PRICE.value), 'Средний')

            dbworker.set(dbworker.make\_key(message.chat.id, config.S\_CUR), config.States.S\_COUNTRY.value)

        elif message.text == 'Престиж':

            dbworker.set(dbworker.make\_key(message.chat.id, config.States.S\_PRICE.value), 'Престиж')

            dbworker.set(dbworker.make\_key(message.chat.id, config.S\_CUR), config.States.S\_COUNTRY.value)

        markup1 = types.ReplyKeyboardMarkup(row\_width=3, resize\_keyboard=True)

        botton1\_1 = types.KeyboardButton('Отечественный')

        botton2\_2 = types.KeyboardButton('Европейский')

        botton3\_3 = types.KeyboardButton('Азиатский')

        markup1.add(botton1\_1, botton2\_2, botton3\_3)

        bot.send\_message(message.chat.id, 'Выберите страну: ', reply\_markup=markup1)

    else:

        markup = types.ReplyKeyboardMarkup(row\_width=3, resize\_keyboard=True)

        botton1 = types.KeyboardButton('Бюджетный')

        botton2 = types.KeyboardButton('Средний')

        botton3 = types.KeyboardButton('Престиж')

        markup.add(botton1, botton2, botton3)

        bot.send\_message(message.chat.id, 'Выберите ценовой диапазон: ', reply\_markup=markup)

@bot.message\_handler(func=lambda message:

dbworker.get(dbworker.make\_key(message.chat.id, config.S\_CUR)) == config.States.S\_COUNTRY.value)

def car\_price(message):

    path = functions.path\_picture(dbworker.get(dbworker.make\_key(message.chat.id, config.States.S\_TYPE.value)),

    dbworker.get(dbworker.make\_key(message.chat.id, config.States.S\_PRICE.value)),

    message.text)

    with open(path, 'rb') as img:

        bot.send\_photo(message.chat.id, img)

bot.infinity\_polling()

Файл **testTDD.py**

import unittest

from functions import path\_picture

class TestTDD(unittest.TestCase):

    def test1(self):

        self.assertEqual(path\_picture('Семейный', 'Престиж', 'Отечественный'),

        r'd:\python\_projects\my\_bot\бот Автомобили\Семейные\3 и больше (престиж)\1.jpg')

    def test2(self):

        self.assertEqual(path\_picture('Грузовой', 'Средний', 'Азиатский'),

        r'd:\python\_projects\my\_bot\бот Автомобили\Грузовые\1.2-3 средний\3.jpg')

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

Файл **steps.py**

from behave import given, when, then

from functions import path\_picture

@given("I have the data {type}, {price} and {country}")

def have\_numbers(context, type, price, country):

    context.type = type

    context.price = price

    context.country = country

@when("I find the path")

def calculate\_roots(context):

    context.path = path\_picture(context.type, context.price, context.country)

@then("I expect the result to be {path}")

def expect\_result(context, path):

    print(path)

    print(context.path)

    assert context.path == path

Файл **testBDD.feature**

Feature: file for test my telebot

    Scenario: Test telebot1

        Given I have the data Спортивный, Престиж and Европейский

        When I find the path

        Then I expect the result to be D:\python\_projects\my\_bot\бот Автомобили\Спортивные\3 и больше (престиж)\2.jpg

    Scenario: Test telebot2

        Given I have the data Семейный, Бюджетный and Отечественный

        When I find the path

        Then I expect the result to be D:\python\_projects\my\_bot\бот Автомобили\Семейные\до 1.2 млн Бюджет\1.jpg

**Пример выполнения программы**

