Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ	**«Информатики и систем управления»**
КАФЕДРА	ИУ5

Дисциплина «Разработка интернет-приложений»

Отчет по лабораторной работе №4
«Шаблоны проектирования и модульное тестирование в Python.»

Студент	группы ИУ5-52Б	Гришин Илья
Преподаватель		Гапанюк Ю.Е.

Цель лабораторной работы: изучение реализации шаблонов проектирования и возможностей модульного тестирования в языке Python.

1. Задание

1. Необходимо для произвольной предметной области реализовать три шаблона проектирования: один порождающий, один структурный и один поведенческий.

2. Для каждой реализации шаблона необходимо написать модульный тест. В модульных тестах необходимо применить следующие технологии:

- TDD - фреймворк.
- BDD - фреймворк.

2. Текст программ

Порождающий паттерн проектирования «строитель»:

stroitel.py

```
from __future__ import annotations
from abc import ABC, abstractmethod, abstractproperty
from typing import Any


class Builder(ABC):

    @abstractproperty
    def product(self) -> None:
        pass

    @abstractmethod
    def setEngine(self) -> None:
        pass

    @abstractmethod
    def setFuel(self) -> None:
        pass

    @abstractmethod
    def setMTransmission(self) -> None:
        pass

    @abstractmethod
    def setATransmission(self) -> None:
        pass

    @abstractmethod
    def setConditioner(self) -> None:
```

```
        pass

    @abstractmethod
    def setOnBoardComp(self) -> None:
        pass

    @abstractmethod
    def setFogLights(self) -> None:
        pass


class ConcreteBuilderCar(Builder):

    def __init__(self) -> None:
        self.reset()

    def reset(self) -> None:
        self._product = ProductCar()

    @property
    def product(self) -> ProductCar:
        product = self._product
        self.reset()
        return product

    def setEngine(self) -> None:
        self._product.add("Двигатель легкового автомобиля")

    def setFuel(self) -> None:
        self._product.add("Бензин")

    def setMTransmission(self) -> None:
        self._product.add("МКП")
        return True

    def setATransmission(self) -> None:
        self._product.add("АКП")

    def setConditioner(self) -> None:
        self._product.add("Кондиционер")

    def setOnBoardComp(self) -> None:
        self._product.add("Бортовой компьютер")

    def setFogLights(self) -> None:
        self._product.add("Противотуманные фары")

class ConcreteBuilderTruck(Builder):

    def __init__(self) -> None:
        self.reset()
```

```
    def reset(self) -> None:
        self._product = ProductTruck()

    @property
    def product(self) -> ProductTruck:
        product = self._product
        self.reset()
        return product

    def setEngine(self) -> None:
        self._product.add("Двигатель грузового автомобиля")

    def setFuel(self) -> None:
        self._product.add("Дизель")

    def setMTransmission(self) -> None:
        self._product.add("МКП")
        return True

    def setATransmission(self) -> None:
        self._product.add("АКП")

    def setConditioner(self) -> None:
        self._product.add("Кондиционер")

    def setOnBoardComp(self) -> None:
        self._product.add("Бортовой компьютер")

    def setFogLights(self) -> None:
        self._product.add("Противотуманные фары")

class ProductCar():
    def __init__(self) -> None:
        self.parts = []

    def add(self, part: Any) -> None:
        self.parts.append(part)

    def list_parts(self) -> None:
        print(f"Комплектация легкового автомобиля: {', '.join(self.parts)}", end=
"")
        return f"{', '.join(self.parts)}"

class ProductTruck():
    def __init__(self) -> None:
        self.parts = []

    def add(self, part: Any) -> None:
        self.parts.append(part)
```

```
    def list_parts(self) -> None:
        print(f"Комплектация грузового автомобиля: {', '.join(self.parts)}", end=
"")
        return f"{', '.join(self.parts)}"

class Director:
    def __init__(self) -> None:
        self._builder = None

    @property
    def builder(self) -> Builder:
        return self._builder

    @builder.setter
    def builder(self, builder: Builder) -> None:
        self._builder = builder

    def build_basic(self) -> None:
        print("Базовая конфигурация автомобиля (Basic): ")
        self.builder.setEngine()
        self.builder.setFuel()
        self.builder.setMTransmission()

    def build_standart(self) -> None:
        print("Стандартная конфигурация автомобиля (Standart): ")
        self.builder.setEngine()
        self.builder.setFuel()
        self.builder.setATransmission()
        self.builder.setConditioner()

    def build_advanced(self) -> None:
        print("Расширенная конфигурация автомобиля (Advanced): ")
        self.builder.setEngine()
        self.builder.setFuel()
        self.builder.setATransmission()
        self.builder.setConditioner()
        self.builder.setOnBoardComp()
        self.builder.setFogLights()


if __name__ == "__main__":
    director = Director()
    builder = ConcreteBuilderCar()
    director.builder = builder

    # print("Базовая конфигурация автомобиля (Basic): ")
    director.build_basic()
    builder.product.list_parts()

    print("\n")
```

```
    # print("Стандартная конфигурация автомобиля (Standart): ")
    director.build_standart()
    builder.product.list_parts()

    print("\n")\

    # print("Расширенная конфигурация автомобиля (Advanced): ")
    director.build_advanced()
    builder.product.list_parts()

    print("\n")

    builder = ConcreteBuilderTruck()
    director.builder = builder

    print("Собранная конфигурация автомобиля (Custom): ")
    builder.setEngine()
    builder.setFuel()
    builder.setMTransmission()
    builder.setConditioner()
    builder.setFogLights()
    builder.product.list_parts()
```

test_stroitel.py

```
import unittest
from stroitel import *

class TestStringMethods(unittest.TestCase):

    def setUp(self):
        self.director=Director()

    def test_setEngine(self):
        builder = ConcreteBuilderCar()
        director = Director()
        director.builder = builder
        builder.setEngine()
        self.assertEqual(builder.product.list_parts(), "Двигатель легкового автом
обиля")
        print("\n")
        builder = ConcreteBuilderTruck()
        director.builder = builder
        builder.setEngine()
        self.assertEqual(builder.product.list_parts(), "Двигатель грузового автом
обиля")
        print("\n")

    def test_build_config(self):
        director = Director()
```

```
        builder = ConcreteBuilderCar()
        director.builder = builder

        director.build_standart()
        self.assertEqual(builder.product.list_parts(), "Двигатель легкового автом
обиля, Бензин, АКП, Кондиционер")
        print("\n")

        director.build_basic()
        self.assertEqual(builder.product.list_parts(), "Двигатель легкового автом
обиля, Бензин, МКП")
        print("\n")

        director.build_advanced()
        self.assertEqual(builder.product.list_parts(), "Двигатель легкового автом
обиля, Бензин, АКП, Кондиционер, Бортовой компьютер, Противотуманные фары")
        print("\n")

    def test_setMTransmission(self):
        director = Director()
        builder = ConcreteBuilderCar()
        director.builder = builder
        self.assertTrue(builder.setMTransmission())


if __name__ == '__main__':
    unittest.main()
```

Структурный паттерн проектирования «адаптер»:

adapter.py

```
import codecs

class Target:
    def request(self,path) -> str:
        f = open(path)
        text=f.read()
        f.close()
        return f"Содержимое файла: "+text


class Adaptee:
    def specific_request(self, path) -> str:
        f = open(path, 'r')
        text=f.read()
        f.close()
        return text


class Adapter(Target, Adaptee):
```

```
    def request(self, path) -> str:
        f=codecs.open(path, encoding='utf8')
        text=f.read()
        f.close()
        return f"Содержимое файла: "+text


def client_code(path, target: "Target") -> None:
    print(target.request(path), end="")
    return target.request(path)


if __name__ == "__main__":
    path = 'C:\\2\\rip\\lab4\\testfiles\\file.txt'
    # path = 'C:\\2\\file.txt'
    print("\n")
    print("Попытка открытия файла без необходимости смены кодировки")
    target = Target()
    client_code(path, target)
    print("\n\n")

    adaptee = Adaptee()
    print("Попытка открытия файла без смены кодировки")
    print(f"Содержимое файла: {adaptee.specific_request(path)}", end="\n\n")

    print("Попытка открытия файла с кодировкой UTF-8")
    adapter = Adapter()
    client_code(path, adapter)
```

test_adapter.py

```
import unittest
from adapter import *
import os

class TestStringMethods(unittest.TestCase):

    def test_adapter(self):
        str1="Текстовый Файл --- "
        for index in 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя':
            with self.subTest(index=index):
                s=str(index)
                testpath="C:\\2\\rip\\lab4\\testfiles\\file"+s+".txt"
                str1 = str1 + s
                with open(testpath, 'wb') as f:
                    f.write(str1.encode('utf-8'))
                f.close()
                print("\n")

                adaptee = Adaptee()
```

```
                print("Попытка открытия файла без смены кодировки")
                print(f"Содержимое файла: {adaptee.specific_request(testpath)}",
end="\n")
                f.close()

                print("Попытка открытия файла с кодировкой UTF-8")
                adapter = Adapter()
                str2='111'
                if s=='ж':
                    self.assertEqual(client_code(testpath, adapter), "Содержимое
файла: " + str2)
                else:
                    self.assertEqual(client_code(testpath, adapter), "Содержимое
файла: " + str1)
                # self.assertEqual(client_code(testpath, adapter), "Содержимое фа
йла: " + str1)
                os.remove(testpath)

if __name__ == '__main__':
    unittest.main()
```

Поведенческий паттерн проектирования «стратегия»:

strategy.py

```
from __future__ import annotations
from abc import ABC, abstractmethod
from typing import List
import json
from tabulate import tabulate

class Context():

    def __init__(self, strategy: Strategy) -> None:
        self._strategy = strategy

    @property
    def strategy(self) -> Strategy:
        return self._strategy

    @strategy.setter
    def strategy(self, strategy: Strategy) -> None:
        self._strategy = strategy

    def do_some_business_logic(self,path) -> None:
        with open(path, encoding='utf-8') as f:
            data = json.load(f)
        result = self._strategy.do_algorithm(data)
        best=[]
        for i in range(0,5):
```

```
            best.append(result[i])
        print ("\n        Лучший отель:")
        print ("Название:           "+result[0]['name'])
        print ("Стоимость за день:  "+str(result[0]['costperday']))
        print ("Звезды:             "+str(result[0]['star']))
        print ("Цена за звезду:     "+str(result[0]['starcost']))
        print ("\n             Топ лучших отелей:")
        print (tabulate(best, headers='keys', tablefmt="fancy_grid"))
        return result[0]['name']



class Strategy(ABC):
    @abstractmethod
    def do_algorithm(self, data):
        pass


class ConcreteStrategyA(Strategy):
    def do_algorithm(self, data):
        return sorted(data, key=lambda k: k['star'], reverse = True)


class ConcreteStrategyB(Strategy):
    def do_algorithm(self, data):
        return sorted(data, key=lambda k: k['costperday'])


class ConcreteStrategyC(Strategy):
    def do_algorithm(self, data):
        return sorted(data, key=lambda k: k['starcost'])


if __name__ == "__main__":

    path = 'data_light.json'
    context = Context(ConcreteStrategyA())
    print("\n\nКлиент: Лучший отель по звездам")
    context.do_some_business_logic(path)
    print()

    print("\n\nКлиент: Лучший отель по цене")
    context.strategy = ConcreteStrategyB()
    context.do_some_business_logic(path)

    print("\n\nКлиент: Лучший отель по цене за звезду")
    context.strategy = ConcreteStrategyC()
    context.do_some_business_logic(path)
```

step.py

```
from strategy import *

# -*- coding: utf-8 -*-

from radish import given, when, then

@given('"I have the path "{path}"')
def have_path(step, path):
    step.context.path = path

@when('I am making a request')
def do_request(step):
    context.strategy = ConcreteStrategyB()
    step.context.result = context.do_some_business_logic(step.context.path)

@then('I expect the result to be "{result}"')
def expect_result(step, result):
    assert step.context.result == result
```

test_strategy.feature

Feature: My first feature file using radish

In order to test my awesome software

I need an awesome BDD tool like radish

to test my software.

Scenario: Test my strategy

Given I have the path "C:\2\rip\lab4\data_light.json"

When I am making a request

Then I expect the result to be "Отель4"

data_light.json

```
[
    {
        "name": "Отель1",
        "costperday": 1000,
        "star": 4,
        "starcost": 250
    },
    {
```

```
        "name": "Отель2",
        "costperday": 1000,
        "star": 5,
        "starcost": 200
    },
    {
        "name": "Отель3",
        "costperday": 1200,
        "star": 4,
        "starcost": 300
    },
    {
        "name": "Отель4",
        "costperday": 300,
        "star": 2,
        "starcost": 150
    },
    {
        "name": "Отель5",
        "costperday": 2000,
        "star": 5,
        "starcost": 400
    },
    {
        "name": "Отель6",
        "costperday": 1600,
        "star": 4,
        "starcost": 400
    },
    {
        "name": "Отель7",
        "costperday": 900,
        "star": 3,
        "starcost": 300
    },
    {
        "name": "Отель8",
        "costperday": 750,
        "star": 3,
        "starcost": 250
    }
]
```

3. Результаты

stroitel.py

```
c:\2\rip\lab4>stroitel.py
Базовая конфигурация автомобиля (Basic):
Комплектация легкового автомобиля: Двигатель легкового автомобиля, Бензин, МКП

Стандартная конфигурация автомобиля (Standart):
Комплектация легкового автомобиля: Двигатель легкового автомобиля, Бензин, АКП,
Кондиционер

Расширенная конфигурация автомобиля (Advanced):
Комплектация легкового автомобиля: Двигатель легкового автомобиля, Бензин, АКП,
Кондиционер, Бортовой компьютер, Противотуманные фары

Собранная конфигурация автомобиля (Custom):
Комплектация грузового автомобиля: Двигатель грузового автомобиля, Дизель, МКП,
Кондиционер, Противотуманные фары
```

test_stroitel.py

```
c:\2\rip\lab4>test_stroitel.py
Стандартная конфигурация автомобиля (Standart):
Комплектация легкового автомобиля: Двигатель легкового автомобиля, Бензин, АКП,
Кондиционер

Базовая конфигурация автомобиля (Basic):
Комплектация легкового автомобиля: Двигатель легкового автомобиля, Бензин, МКП

Расширенная конфигурация автомобиля (Advanced):
Комплектация легкового автомобиля: Двигатель легкового автомобиля, Бензин, АКП,
Кондиционер, Бортовой компьютер, Противотуманные фары

.Комплектация легкового автомобиля: Двигатель легкового автомобиля

Комплектация грузового автомобиля: Двигатель грузового автомобиля

..
----------------------------------------------------------------------
Ran 3 tests in 0.001s

OK
```

adapter.py

```
c:\2\rip\lab4>adapter.py


Попытка открытия файла без необходимости смены кодировки
Содержимое файла: Hello, World! (РџСЂРёРІРµС,, РњРёСЂ!)



Попытка открытия файла без смены кодировки
Содержимое файла: Hello, World! (РџСЂРёРІРµС,, РњРёСЂ!)


Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Hello, World! (Привет, Мир!)
```

test_adapter.py

```
c:\2\rip\lab4>test_adapter.py


Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- а

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- аб

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абв

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІРіi
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвг

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІРіiСГ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгд

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІРіiСГРµ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгде

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІРіiСГРµС‘
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеё

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІРіiСГРµС‘Р¶
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёж

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІРіiСГРµС‘Р¶Р·
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжз

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІРіiСГРµС‘Р¶Р·Рё
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзи

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,PsPIC‹Р№ РѕР°РЅР»Р» --- Р°Р±РІРіiСГРµС‘Р¶Р·РёР№
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзий
```

```
Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийк

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийкл

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»Рj
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклм

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»РjPS

Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмн

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»РjPS
Ps
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмно

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»РjPS
PsPї
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмноп

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»РjPS
PsPїСЂ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопр

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»РjPS
PsPїСЂСѓ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрс

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»РjPS
PsPїСЂСѓС,
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрст

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»РjPS
PsPїСЂСѓС,Сґ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрсту

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- РѕР±СЂР°Р¶РµРЄР»РjPS
PsPїСЂСѓС,СґС„
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуф
```

```
Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфх

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхц

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†С‡
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцч

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†С‡С€
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцчш

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†С‡С€С‰
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцчщ

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†С‡С€С‰СЉ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцчщъ

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†С‡С€С‰СЉС‹
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцчщъы

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†С‡С€С‰СЉС‹Сњ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцчщъыь

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†С‡С€С‰СЉС‹СњРЄ
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцчщъыьэ

Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚РѕРІ» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅ
РѕРїСЂРѕСЃ„С…С†С‡С€С‰СЉС‹СњРЄР®
Попытка открытия файла с кодировкой UTF-8
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцчщъыьэю
```

```
Попытка открытия файла без смены кодировки
Содержимое файла: РўРµРєСЃС‚,РѕРІС‹Р№ РґР°Р№Р» --- Р°Р±РІРіРґРµС‘Р¶Р·РёР№РєР»РјРЅРѕРїСЂС
РѕРїСЂРѕСЃ,СѓС„„С…С‡С€С€С‰СЊС‹СЊСЌСЋСЏ
Попытка открытия файла с кодировкой UTF-8

======================================================================
FAIL: test_adapter (__main__.TestStringMethods) (index='ж')
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\2\rip\lab4\test_adapter.py", line 28, in test_adapter
    self.assertEqual(client_code(testpath, adapter), "Содержимое файла: " + str2
)
AssertionError: 'Содержимое файла: Текстовый Файл --- абвгдеёж' != 'Содержимое ф
айла: 111'
- Содержимое файла: Текстовый Файл --- абвгдеёж
+ Содержимое файла: 111


----------------------------------------------------------------------
Ran 1 test in 0.075s

FAILED (failures=1)
Содержимое файла: Текстовый Файл --- абвгдеёжзийклмнопрстуфхцчшщъыьэюя
```

strategy.py

```
c:\2\rip\lab4>strategy.py


Клиент: Лучший отель по звездам

        Лучший отель:
Название:           Отель2
Стоимость за день:  1000
Звезды:             5
Цена за звезду:     200

             Топ лучших отелей:
```

name	costperday	star	starcost
Отель2	1000	5	200
Отель5	2000	5	400
Отель1	1000	4	250
Отель3	1200	4	300
Отель6	1600	4	400

Клиент: Лучший отель по цене

Лучший отель:
Название: Отель4
Стоимость за день: 300
Звезды: 2
Цена за звезду: 150

Топ лучших отелей:

name	costperday	star	starcost
Отель4	300	2	150
Отель8	750	3	250
Отель7	900	3	300
Отель1	1000	4	250
Отель2	1000	5	200

Клиент: Лучший отель по цене за звезду

Лучший отель:
Название: Отель4
Стоимость за день: 300
Звезды: 2
Цена за звезду: 150

Топ лучших отелей:

name	costperday	star	starcost
Отель4	300	2	150
Отель2	1000	5	200
Отель1	1000	4	250
Отель8	750	3	250
Отель3	1200	4	300