

Московский государственный технический университет им. Н.Э. Баумана

Факультет “Радиотехнический”
Кафедра ИУ5 “Системы обработки информации и управления”

Отчет по проделанной лабораторной работе номер 4
по курсу

Базовые компоненты интернет технологий

Вариант 14

Подготовил:

Студент группы РТ5-31Б

Мамаев Т.Э.

Проверил:

Доцент кафедры ИУ5

Гапанюк Ю.Е.

23 Декабря 2021г

Описание задания

1. Необходимо для произвольной предметной области реализовать от одного до трех шаблонов проектирования: один порождающий, один структурный и один поведенческий. В качестве справочника шаблонов можно использовать следующий каталог. 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 Моск-объектов.

Текст программы

```
#builder.py
from __future__ import annotations
from abc import ABC, abstractmethod
from typing import Any
```

```
class Builder(ABC):
```

```
    @property
    @abstractmethod
```

```

def car(self) -> None:
    pass

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

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

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

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

```

```

class CarBuilder(Builder):
    def __init__(self) -> None:
        self.reset()

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

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

    def wheels(self) -> None:
        self._product.add("Колеса")

    def engine(self) -> None:
        self._product.add("Двигатель")

    def body(self) -> None:
        self._product.add("Кузов")
    def turbo(self) -> None:
        self._product.add("Турбина")
class Car():
    def __init__(self) -> None:
        self.parts = []

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

    def list_parts(self) -> None:

```

```
        print(f"Части автомобиля: {' '.join(self.parts)}",
end="")
```

```
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 TurboCar(self) -> None:
        self.builder.wheels()
        self.builder.engine()
        self.builder.body()
        self.builder.turbo()

    def AtmoCar(self) -> None:
        self.builder.wheels()
        self.builder.engine()
        self.builder.body()
```

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

    print("Турбовый автомобиль: ")
    director.TurboCar()
    builder.car.list_parts()
    print("\n")
    print("Автомобиль без турбины: ")
    director.AtmoCar()
    builder.car.list_parts()
```

```
#test_mock.py
import unittest
import sys, os
from unittest.main import main
from unittest.mock import patch, Mock
```

```
import Builder
sys.path.append(os.getcwd())
from Builder import *
```

```
class Test_builder(unittest.TestCase):
```

```

@patch.object(Director, 'TurboCar')
def test_turbocar(self, mock_turbocar):
    mock_turbocar.return_value = None

    director = Director()
    builder = CarBuilder()
    director.builder = builder

    self.assertEqual(director.TurboCar(), None)
if __name__ == "__main__":
    unittest.main()

#test_tdd.py
import unittest
from Builder import *

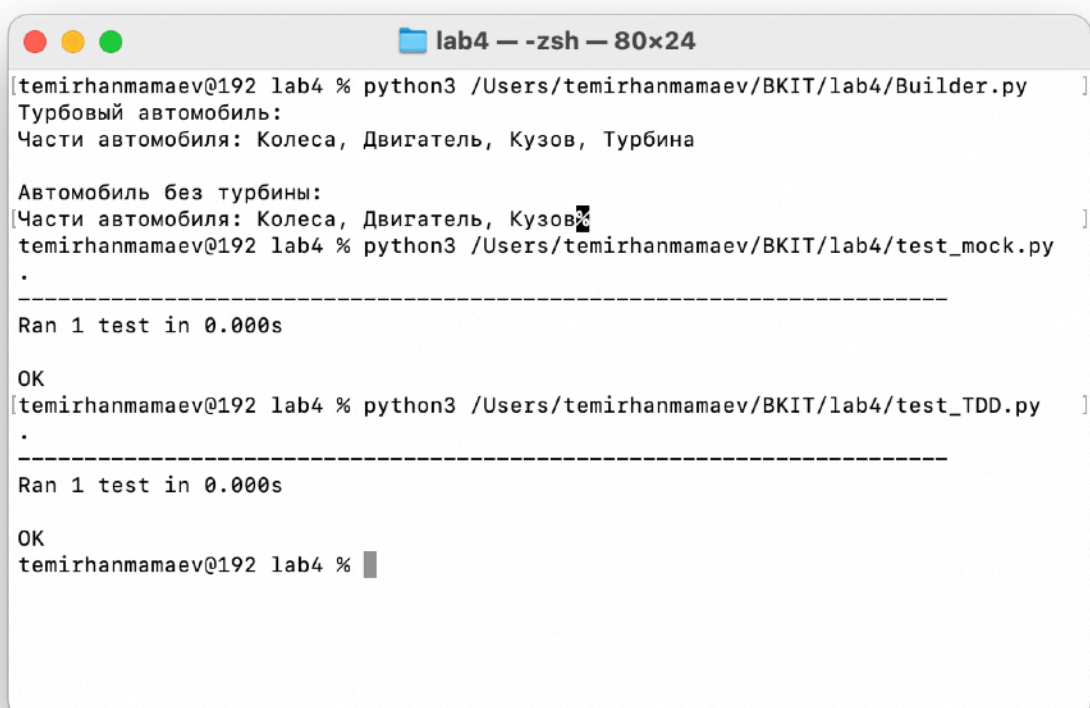
class TestBuilder(unittest.TestCase):
    def Test_builder(self):
        builder = ()
        self.assertEqual(builder.create(), None)

    def test_wheels(self):
        builder = CarBuilder()
        self.assertEqual(builder.wheels(), None)

if __name__ == "__main__":
    unittest.main()

```

Результат выполнения программы



```

lab4 — zsh — 80x24
[temirhanmamaev@192 lab4 % python3 /Users/temirhanmamaev/BKIT/lab4/Builder.py ]
Турбовый автомобиль:
Части автомобиля: Колеса, Двигатель, Кузов, Турбина

Автомобиль без турбины:
[Части автомобиля: Колеса, Двигатель, Кузов%
temirhanmamaev@192 lab4 % python3 /Users/temirhanmamaev/BKIT/lab4/test_mock.py ]
.
-----
Ran 1 test in 0.000s

OK
[temirhanmamaev@192 lab4 % python3 /Users/temirhanmamaev/BKIT/lab4/test_TDD.py ]
.
-----
Ran 1 test in 0.000s

OK
temirhanmamaev@192 lab4 % █

```