

**Московский государственный технический
университет им. Н.Э. Баумана**

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

Курс «Парадигмы и конструкции языков программирования»
Отчет по лабораторной работе №4
«Введение в модульное тестирование и паттерны проектирования»

Выполнил:

студент группы ИУ5-31Б

Маркин Денис

Проверил:

преподаватель каф. ИУ5

Нардид Анатолий
Николаевич

Москва, 2024 г.

Постановка задачи

1. Необходимо для произвольной предметной области реализовать от одного до трех шаблонов проектирования: один порождающий, один структурный и один поведенческий. Для сдачи лабораторной работы в минимальном варианте достаточно реализовать один паттерн.
2. В модульных тестах необходимо применить следующие технологии:
 - TDD - фреймворк.
 - BDD - фреймворк.
 - Создание Mock-объектов.

Текст Программы

Behavioral.py

```
from abc import ABC, abstractmethod
```

```
#State Шаблон состояния
```

```
class State(ABC):
```

```
    @abstractmethod
```

```
    def handle(self):
```

```
        pass
```

```
class Outside(State):
```

```
    def handle(self):
```

```
        return "Студент не в ВУЗе"
```

```
class Inside(State):
```

```
    def handle(self):
```

```
        return "Студент учится"
```

```
class Walk(State):
```

```
def handle(self):  
    return "Студент в пути"
```

```
class Window(State):  
    def handle(self):  
        return "Студент в ВУЗе, но сейчас у него окно"
```

```
class Student:  
    def __init__(self):  
        self.state = Outside()  
  
    def set_state(self, state: State):  
        self.state = state  
  
    def where(self):  
        return self.state.handle()
```

```
me = Student()  
print(me.where())  
me.set_state(Inside())  
print(me.where())  
me.set_state(Window())  
print(me.where())  
me.set_state(Outside())  
print(me.where())  
me.set_state(Walk())  
print(me.where())
```

test_Behavioral.py

```
import unittest
```

```
# Тест пройден Сначала тесты, потом разработка
```

```
from Behavioral import Student, Outside, Inside, Window, Walk
```

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

```
    def setUp(self):
```

```
        self.student = Student()
```

```
    def test_initial_state(self):
```

```
        self.assertEqual(self.student.where(), "Студент не в ВУЗе")
```

```
    def test_state_inside(self):
```

```
        self.student.set_state(Inside())
```

```
        self.assertEqual(self.student.where(), "Студент учится")
```

```
    def test_state_window(self):
```

```
        self.student.set_state(Window())
```

```
        self.assertEqual(self.student.where(), "Студент в ВУЗе, но сейчас у  
него окно")
```

```
    def test_state_outside(self):
```

```
        self.student.set_state(Outside())
```

```
        self.assertEqual(self.student.where(), "Студент не в ВУЗе")
```

```
    def test_state_walk(self):
```

```
        self.student.set_state(Walk())
```

```
self.assertEqual(self.student.where(), "Студент в пути")
```

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

```
● Студент не в ВУЗе  
  Студент учится  
  Студент в ВУЗе, но сейчас у него окно  
  Студент не в ВУЗе  
  Студент в пути
```

Behavioral.py

```
● Студент не в ВУЗе  
  Студент учится  
  Студент в ВУЗе, но сейчас у него окно  
  Студент не в ВУЗе  
  Студент в пути
```

```
.....
```

```
-----  
Ran 5 tests in 0.001s
```

OK

■