**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](https://github.com/ugapanyuk/courses_content/blob/main/pres/pcpl/test_pattern.pdf)»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Маркин Денис |  | 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()

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описаниеBehavioral.py

Изображение выглядит как текст, снимок экрана, Шрифт, инструмент

Автоматически созданное описание