Московский государственный технический университет им. Н.Э. Баумана

Факультет ИУ
Кафедра ИУ5

Курс «Основы информатики»

Отчет по лабораторной работе №_5-6_ «Шаблоны проектирования и модульное тестирование в Python»

Выполнил:
студент группы ИУ5-33Б:

Номоконов В.А
Подпись и дата:

Проверил:
преподаватель каф.

____

_____
Подпись и дата:

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. В качестве справочника шаблонов можно использовать следующий каталог. 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

lab5-6\design_patterns\Adapter.py

```
class ExternalTestService:
    def run_quiz(self, quiz_data):
        print(f"Running external quiz with questions: {quiz_data}")

class LocalTestSystem:
    def start_test(self, questions):
        print(f"Starting test with questions: {questions}")

class TestAdapter:
    def __init__(self, external_service):
        self.external_service = external_service

    def start_test(self, questions):
        quiz_data = {"questions": questions}
        self.external_service.run_quiz(quiz_data)

# Использование
local_test = LocalTestSystem()
local_test.start_test(["Question 1", "Question 2"])

external_service = ExternalTestService()
adapter = TestAdapter(external_service)
adapter.start_test(["Question A", "Question B"])
```

lab5-6\design_patterns\Observer.py

```
class CourseNotifier:
    def __init__(self):
        self._observers = []

    def subscribe(self, observer):
        self._observers.append(observer)

    def unsubscribe(self, observer):
        self._observers.remove(observer)

    def notify(self, course):
        for observer in self._observers:
```

```
            observer.update(course)

    class Student:
        def __init__(self, name):
            self.name = name

        def update(self, course):
            print(f"{self.name}, новый курс доступен: {course}")

    # Использование
    notifier = CourseNotifier()

    student1 = Student("Алексей")
    student2 = Student("Мария")

    notifier.subscribe(student1)
    notifier.subscribe(student2)

    notifier.notify("Python для начинающих")
    # Алексей, новый курс доступен: Python для начинающих
    # Мария, новый курс доступен: Python для начинающих
```

lab5-6\design_patterns\Singleton.py

```
    class SettingsMeta(type):
        _instances = {}

        def __call__(cls, *args, **kwargs):
            if cls not in cls._instances:
                cls._instances[cls] = super().__call__(*args, **kwargs)
            return cls._instances[cls]

    class Settings(metaclass=SettingsMeta):
        pass

    class PlatformSettings(Settings):
        def __init__(self):
            self.theme = "Light"
            self.language = "English"

        def update_settings(self, theme: str = None, language: str = None) ->
None:
            if theme:
                self.theme = theme
            if language:
                self.language = language

        def display(self) -> dict:
            return {"theme": self.theme, "language": self.language}

    # Использование
    settings1 = PlatformSettings()
    settings2 = PlatformSettings()

    settings1.update_settings(theme="Dark", language="Spanish")
```

```
print(settings2.display())  # {'theme': 'Dark', 'language': 'Spanish'}
print(settings1 is settings2)  # True
```

lab5-6\tests\features\steps\manage_courses(BDD).py

```
from behave import given, when, then
from course_manager import CourseManager

@given("I have an empty course list")
def step_given_empty_course_list(context):
    context.manager = CourseManager()

@when('I add a course titled "{title}" in "{category}" category')
def step_when_add_course(context, title, category):
    context.manager.add_course(title, category)

@then("I should see the course in the list")
def step_then_see_course_in_list(context):
    courses = context.manager.list_courses()
    assert len(courses) == 1
    assert courses[0]["title"] == "Python Basics"
    assert courses[0]["category"] == "Programming"

@given('I have a course titled "{title}" in "{category}" category')
def step_given_existing_course(context, title, category):
    context.manager = CourseManager()
    context.manager.add_course(title, category)

@when('I add a course titled "{title}" in "{category}" category')
def step_when_add_duplicate_course(context, title, category):
    context.manager.add_course(title, category)
```

lab5-6\tests\features\duplicate_courses.feature

```
Feature: Duplicate courses

Scenario: Prevent adding duplicate courses
  Given I have a course titled "Python Basics" in "Programming" category
  When I add a course titled "Python Basics" in "Programming" category
  Then I should not see a duplicate coursedef test_1():
    return 1
```

lab5-6\tests\features\manage_courses.feature

```
Feature: Course management

  Scenario: Add a course to the system
    Given I have an empty course list
    When I add a course titled "Python Basics" in "Programming" category
    Then I should see the course in the list
```

lab5-6\tests\course_manager.py

```
class Course:
  def __init__(self, title, category):
    self.title = title
    self.category = category

class CourseManager:
  def __init__(self):
    self.courses = []

  def add_course(self, title, category):
    course = Course(title, category)
    self.courses.append(course)
    return course

  def list_courses(self):
    return [{"title": c.title, "category": c.category} for c in self.courses]
```

lab5-6\tests\test_course_manager(TDD).py

```
import pytest
from course_manager import CourseManager

@pytest.fixture
def course_manager():
    """Фикстура для инициализации CourseManager с тестовыми данными."""
    manager = CourseManager()
    manager.add_course("Python Basics", "Programming")
    manager.add_course("Data Science", "Data Analysis")
    return manager

def test_add_course(course_manager):
    """Тест добавления нового курса."""
    new_course = course_manager.add_course("Machine Learning", "AI")
    assert new_course.title == "Machine Learning"
    assert new_course.category == "AI"
    assert len(course_manager.list_courses()) == 3

def test_list_courses(course_manager):
    """Тест получения списка курсов."""
    courses = course_manager.list_courses()
    assert len(courses) == 2
    assert courses[0]["title"] == "Python Basics"
    assert courses[1]["category"] == "Data Analysis"

def test_no_duplicate_courses(course_manager):
    """Тест, чтобы убедиться, что дублирующиеся курсы не добавляются."""
    course_manager.add_course("Python Basics", "Programming")
    courses = course_manager.list_courses()
    assert len(courses) == 2  # Список остается неизменным
```

```
def test_empty_course_list():
    """Тест, что при пустом менеджере список курсов пуст."""
    manager = CourseManager()
    courses = manager.list_courses()
    assert courses == []
```

lab5-6\tests\test_notifications(Mock).py

```
from unittest.mock import MagicMock
from course_manager import CourseManager

class Notifier:
    def notify(self, student, course):
        print(f"Notifying {student} about new course: {course}")

def test_notification():
    notifier = Notifier()
    notifier.notify = MagicMock()

    course_manager = CourseManager()
    course_manager.add_course("Machine Learning", "AI")

    notifier.notify("Alex", "Machine Learning")
    notifier.notify.assert_called_once_with("Alex", "Machine Learning")
```

Анализ результатов

```
PS C:\Users\exxor\PyLabsWW> pytest
============================================= test session starts =============================================
platform win32 -- Python 3.11.10, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\Users\exxor\PyLabsWW
collected 5 items

labs\lab5-6\tests\test_course_manager(TDD).py ..F.                                                    [ 80%]
labs\lab5-6\tests\test_notifications(Mock).py .                                                       [100%]

================================================== FAILURES ===================================================
_____________________________________________ test_no_duplicate_courses _____________________________________________

course_manager = <course_manager.CourseManager object at 0x0000025826FB6ED0>

    def test_no_duplicate_courses(course_manager):
        """Тест, чтобы убедиться, что дублирующиеся курсы не добавляются."""
        course_manager.add_course("Python Basics", "Programming")
        courses = course_manager.list_courses()
>       assert len(courses) == 2  # Список остается неизменным
E       AssertionError: assert 3 == 2
E        +  where 3 = len([{'category': 'Programming', 'title': 'Python Basics'}, {'category': 'Data Analysis', 'title': 'Data S
cience'}, {'category': 'Programming', 'title': 'Python Basics'}])

labs\lab5-6\tests\test_course_manager(TDD).py:30: AssertionError
=========================================== short test summary info ===========================================
FAILED labs/lab5-6/tests/test_course_manager(TDD).py::test_no_duplicate_courses - AssertionError: assert 3 == 2
========================================= 1 failed, 4 passed in 0.26s =========================================
```

```
PS C:\Users\exxor\PyLabsWW> & c:/Users/exxor/PyLabsWW/.conda/python.exe c:/Users/exxor/PyLabsWW/labs/lab5-6/design_patterns/Adap
ter.py
Starting test with questions: ['Question 1', 'Question 2']
Running external quiz with questions: {'questions': ['Question A', 'Question B']}
PS C:\Users\exxor\PyLabsWW>
```

```
PS C:\Users\exxor\PyLabsWW> & c:/Users/exxor/PyLabsWW/.conda/python.exe c:/Users/exxor/PyLabsWW/labs/lab5-6/design_patterns/Obs
rver.py
Алексей, новый курс доступен: Python для начинающих
Мария, новый курс доступен: Python для начинающих
PS C:\Users\exxor\PyLabsWW>
```

```
PS C:\Users\exxor\PyLabsWW> & c:/Users/exxor/PyLabsWW/.conda/python.exe c:/Users/exxor/PyLabsWW/labs/lab5-6/design_patterns/Sing
leton.py
{'theme': 'Dark', 'language': 'Spanish'}
True
PS C:\Users\exxor\PyLabsWW>
```