Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

**Департамент анализа данных и машинного обучения**

**Отчёт**

по дисциплине «Управление качеством программных систем»

**Лабораторная работа №4**

Выполнила:

студентка учебной группы ПИ21-1 Агапенкова Д.А.

Научный руководитель:

*Клочков Евгений Юрьевич*

  Москва 2024

**Цель работы:**

Протестировать программный продукт с помощью набора тестов.

**1. Описание эквивалентных классов**

Эквивалентный класс - это набор тестовых данных, которые имеют одинаковое влияние на тестируемую функциональность.

В данном случае мы выделили следующие эквивалентные классы:

Корректные данные: данные, которые соответствуют всем требованиям к формату и содержанию.

Некорректные данные: данные, которые не соответствуют хотя бы одному требованию к формату и содержанию.

1. **Список используемых тестов:**

Тесты можно увидеть здесь:

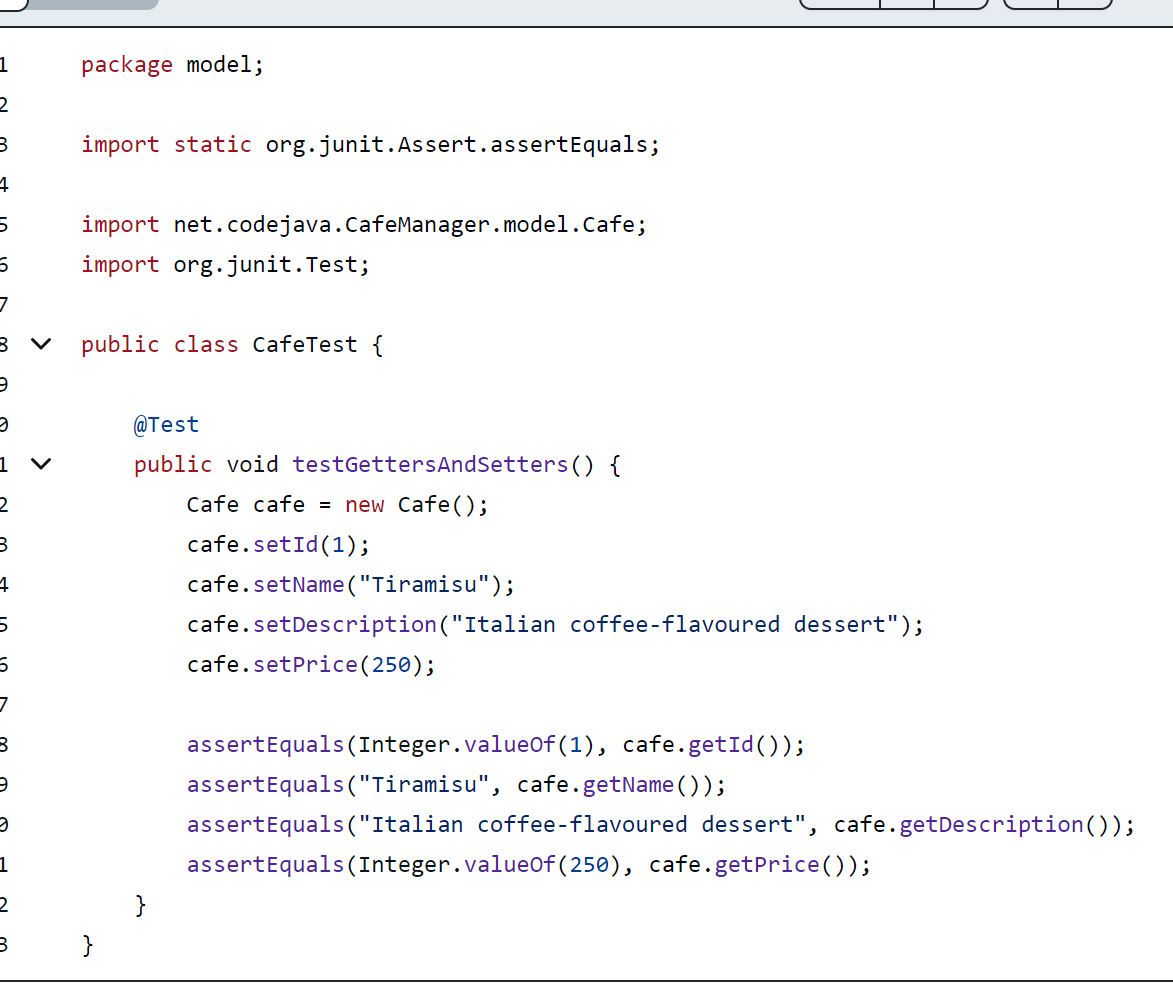
**Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание**

Отдельно тестируются сервисы и репозитории. Для каждого – соответствующие тесты.

1. Тесты для моделей завтрака, обеда, ужина, кофе, доставки. Примеры тестов (представлены соответственно не все):Изображение выглядит как текст, снимок экрана, число, Шрифт

   Автоматически созданное описание

****

****

1. **Тесты для сервисов завтрака, обеда, ужина, кофе, доставки.**

**Пример:**

**Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание**

Для каждого метода каждого класса существует тест для проверки корректности его работы. Верно работает сортировка, поиск по строке товара в таблице, добавление товара, удаление товара и т.д.

**Описание эквивалентных классов:**

Ниже описаны эквивалентные классы для некоторых тестов. Остальные можно выделить аналогично, т.к. данные схожи.

1. Тестирование моделей на примере завтраков.

Тестирование функции создания завтрака.

Эквивалентный класс: Создано корректно.

Тестовые данные:

1

"Яйца с беконом"

799

Сравниваем элемент с:

1

"Яйца с беконом"

799

Эквивалентный класс: Создано некорректно.

Тестовые данные:

1

"Яйца с беконом"

799

Сравниваем элемент с:

1

"Яйца с беконом"

899

2. Тестирование сервисов. На примере сервиса для завтраков.

Тестирование функции поиска по айди.

Эквивалентный класс: Корректный результат поиска.

Тестовые данные:

1

"Яйца с беконом"

799

Поиск по айди 1

Эквивалентный класс: Некорректный результат поиска.

Тестовые данные:

1

"Яйца с беконом"

799

Поиск по айди 2

**Тестирование:**

Код тестируемой программы и тестов находится по ссылке: <https://github.com/DashaAga/spring-boot-server>

Тесты были пройдены успешно.

**Выводы и ответы на контрольные вопросы:**

В целом, достаточно небольшого количества тестов, чтобы проверять корректность работы кода. Необходимо проверить каждый класс, сервис на работу каждого метода и корректной обработки строк: ввода чисел вместо букв и наоборот, ввода пустого поля, недостаточная длина строки (например, для номеров телефона).

1. Опишите методику выделения эквивалентных классов. Методика выделения эквивалентных классов заключается в группировке входных данных или состояний программы в классы, которые предполагается эквивалентными, чтобы уменьшить количество тестовых случаев.

2. В чем цель тестирования граничных значений? Цель тестирования граничных значений - проверить поведение программы на границах допустимых значений входных данных, так как ошибки часто возникают именно на этих границах.

3. Что такое методика черного ящика? Методика черного ящика - это тестирование программы без знания внутренней реализации, только на основе её функциональных требований и спецификаций.

4. В чем разница между методикой черного, белого и серого ящиков? Методика черного ящика сосредотачивается на функциональности приложения, белый ящик проверяет внутреннюю структуру и код, а серый ящик - комбинацию двух предыдущих подходов.

5. Что представляет собой тест-дизайн? Тест-дизайн представляет собой процесс планирования, создания и организации тестовых случаев и процедур, основанных на требованиях к программному обеспечению.