Лабораторная работа №8
Инспекция кода модулей проекта

1 Цель работы

1.1 Изучить техники рефакторинга программного кода.

1.2 Изучить процесс инспекции программного кода.

2 Литература

2.1 Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). – URL: https://znanium.com/catalog/product/1794453 . – Режим доступа: по подписке. – Текст : электронный. – п.5.9.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

В файле OrderManagementApp.cs представлен исходный код приложения для рефакторинга. Нужно создать проект, использующий БД (в коде указано подключение к SQLite, можно использовать другую СУБД) и EF Core. После проверки работоспособности проекта провести инспецию программного кода и выполнить его рефакторинг по заданию.

5.1 Переименование переменной

В классе OrderService метод CalculateFinalPrice использует переменную t. Нужно присвоить более осмысленное название, чтобы значение было понятно другим разработчикам.

5.2 Замена магического числа

В классе OrderService метод CalculateFinalPrice использует "магические числа". Нужно присвоить им более осмысленные названия, чтобы значение было понятно другим разработчикам.

5.3 Упрощение условного выражения

В методе CalculateFinalPrice класса OrderService нужно сделать условное выражение для расчета скидки более компактным с помощью тернарного оператора.

5.4 Инкапсуляция поля

В классе Customer поле Name является публичным, что позволяет любому коду изменить его напрямую. Нужно инкапсулировать это поле через свойство и добавить проверку, чтобы значение имени не было пустым или null.

5.5 Перемещение метода

В классе Order имеется ссылка на объект Customer. Метод PrintOrderDetails в OrderService выводит адрес электронной почты клиента. Нужно переместить метод, который выводит информацию о клиенте, в класс Customer, чтобы логика находилась ближе к данным.

5.6 Извлечение метода

В классе OrderService метод PrintOrderDetails делает несколько вещей: выводит идентификатор заказа, итоговую сумму, статус экспресс-доставки и электронную почту клиента. Нужно разделить эти задачи, выделив каждую часть в отдельные методы.

5.7 Замена вложенных условий на guard clauses

В классе CustomerService метод PrintCustomerInfo проверяет customer на null и выводит данные, если он не равен null. Нужно изменить этот код, чтобы использовать guard clauses и сразу выйти из метода, если customer равен null.

5.8 Замена длинного метода

В классе OrderService метод CalculateFinalPrice делает слишком много работы по расчету итоговой стоимости заказа. Разделить этот метод на несколько частей, выделив в отдельный объект-калькулятор, который будет выполнять расчет и учитывать различные параметры.

6 Порядок выполнения работы

6.1 Используя БД MySQL или MSSQL, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего выполняется инспекция программного кода?

8.2 Какие основные критерии следует учитывать при проведении инспекции программного кода?

8.3 Как правильно организовать процесс рефакторинга кода без нарушения существующей функциональности?

8.4 Какие инструменты статического анализа кода можно использовать для выявления «запахов кода»?

8.5 Как оценить эффективность проведенного рефакторинга?

8.6 Какие метрики можно использовать для измерения улучшений проведенного рефакторинга ?