**Отчет по самостоятельной работе №8**

**по дисциплине МДК 01.02 “Инструментальные средства разработки программного обеспечения”.**

Выполнил: студент

группы 319

Меркушин Евгений Романович

Дата 17.11.2024

**Цель работы**

Цель данной работы – научиться проводить инспекцию кода, выявлять ошибки и улучшать качество модулей программного проекта.

**Основная структура задания**

# Задание 1: Анализ кода модулей

1. **Проведение анализа кода на наличие ошибок**:
   * Прочитайте код модуля внимательно, обращая внимание на синтаксис, логику и структуру.
   * Ищите очевидные ошибки, такие как пропущенные операторы, неверные сравнения, отсутствие проверок на null и т.п.
   * Также обращайте внимание на возможное неправильное поведение программы в определенных условиях.
2. **Исправление выявленных ошибок**:
   * Для каждой найденной ошибки напишите комментарий, объясняющий проблему.
   * Исправьте ошибки, следя за тем, чтобы не нарушить логику работы программы.
   * Если возможно, добавьте тесты для проверки исправленного кода.

# Задание 2: Оптимизация кода

1. **Оптимизация кода, устранение дублирования и неэффективных конструкций**:
   * Проверьте код на наличие дублирующихся фрагментов. Если найдете, подумайте о вынесении общего кода в отдельную функцию или метод.
   * Оцените эффективность используемых алгоритмов. Возможно, некоторые задачи можно решить быстрее и эффективнее.
   * Обратите внимание на использование коллекций и структур данных. Иногда замена одной коллекции на другую может значительно ускорить выполнение программы.
2. **Повторная инспекция**:
   * Еще раз внимательно прочитайте код, чтобы убедиться, что все изменения сделаны корректно и не привели к новым ошибкам.
   * Добавьте комментарии, поясняющие сделанные изменения.

# Задание 3: Использование инструментов статического анализа

1. **Инспекция кода с помощью инструментов статического анализа**:
   * Установите и настройте инструмент статического анализа, например, SonarQube.
   * Запустите анализ кода через инструмент и изучите полученные отчеты.
   * Найдите все ошибки и предупреждения, указанные инструментом.
2. **Исправление ошибок и предупреждений**:
   * Решите все проблемы, отмеченные инструментом. Некоторые из них могут потребовать тщательного анализа и переписывания кода.
   * Перепроверьте код после исправлений, чтобы удостовериться, что новые ошибки не возникли.

# Задание 4: Документирование кода

1. **Проверка соответствия стандартам документирования**:
   * Ознакомьтесь со стандартами документирования, принятыми в вашей команде или организации.
   * Просмотрите весь код модуля и убедитесь, что каждая функция, класс и важный фрагмент имеют соответствующие комментарии.
   * Особое внимание уделите комментариям к сложным участкам кода, где логика может быть неочевидна.
2. **Добавление недостающей документации**:
   * Напишите комментарии там, где они отсутствуют или недостаточны.
   * Используйте стандарты оформления комментариев, принятые в вашем проекте (например, Javadoc для Java, docstrings для Python).

# Задание 5: Тестирование после инспекции кода

1. **Тестирование модуля после исправления ошибок**:
   * Создайте или обновите существующие тесты для модуля.
   * Выполните тесты, чтобы проверить, что модуль работает корректно после всех изменений.
2. **Убедитесь, что тесты проходят корректно**:
   * Если тесты не проходят, найдите причину и исправьте ошибки.
   * Повторите тестирование до тех пор, пока все тесты не будут проходить успешно.

**Вывод**

Эти задания направлены на улучшение качества кода и его устойчивости к ошибкам. Регулярная инспекция, оптимизация и документирование кода помогают поддерживать высокий уровень разработки и облегчают дальнейшее сопровождение проекта.

Начало формы