

Практическая работа №4

Инспектирование программного кода

1 Цель работы

1.1 Изучить методы инспектирования кода программы и научиться применять их на практике для повышения качества программного обеспечения.

2 Литература

2.1 Игнатьев, А. В. Тестирование программного обеспечения : учебное пособие для вузов / А. В. Игнатьев. — 4-е изд., стер. — Санкт-Петербург : Лань, 2025. — 56 с. — ISBN 978-5-507-50858-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/481331> — Режим доступа: для авториз. пользователей.

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

5.1 Разработка кода для инспектирования

5.1.1 Создать новый публичный репозиторий GitHub с файлом readme.md, затем добавить в репозиторий новую ветку. Все следующие действия выполнять внутри созданной ветки.

5.1.2 Создать новый проект библиотеки классов. Разработать в проекте метод для сортировки и фильтрации уведомлений в списке со следующей сигнатурой:

```
IEnumerable<Notification> FilterAndSort(
    IEnumerable<Notification> notifications,
    NotificationFilterOptions options
) { }
```

Структура класса Notification следующая:

```
class Notification
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string? Content { get; set; }
    public NotificationType Type { get; set; } // Info,
    Warning, Error, System
    public DateTime CreatedAt { get; set; }
    public int Priority { get; set; } // 0 (низкий) до 5
    (высокий)
```

```
    public bool IsRead { get; set; }  
}
```

Класс `NotificationFilterOptions` должен содержать свойства, определяющие сортировку и фильтрацию данных по определенным критериям:

```
class NotificationFilterOptions  
{  
    public bool? IsRead { get; set; }  
    public NotificationType[]? Types { get; set; } // список нужных типов  
    public string? SearchText { get; set; }  
    public int? MinPriority { get; set; }  
    public SortNotificationBy? SortBy { get; set; } // по Date, Priority, Title  
    public bool Descending { get; set; }  
}
```

При разработке старайтесь придерживаться принципам качества кода и SOLID.

5.1.3 Отправить изменения в GitHub и создать новый pull-request.

5.2 Выполнение инспектирования

5.2.1 Откройте публичный репозиторий проекта другого студента и изучите его pull-request и начните review.

5.2.2 Оцените выбранные разработчиком архитектуру метода и стратегию подхода к фильтрации, сортировке.

5.2.3 Оцените наличие или отсутствие дефектов в разрабатываемом коде, таких как неполная/неверная реализация заданных требований, отсутствие логических проверок на null и т.д.

5.2.4 Оцените соответствие кода метода принципам SOLID и тестируемость программного кода.

5.3 Сделайте вывод исходя из оценок, укажите результат инспектирования в репозитории.

6 Порядок выполнения работы

6.1 Повторить теоретический материал п. 3.1;

6.2 Выполнить задание п. 5.1-5.3;

6.3 Ответить на контрольные вопросы п. 8;

6.4 Заполнить отчет п. 7.

7 Содержание отчета

7.1 Титульный лист;

7.2 Цель работы;

7.3 Ответы на контрольные вопросы п. 6.3;

7.4 Вывод по проделанной работе.

8 Контрольные вопросы

- 8.1 Какие основные техники инспектирования кода существуют?
- 8.2 Что такое принципы SOLID?
- 8.3 Какие ошибки можно предотвратить при помощи инспектирования кода?