

Упражнения: Документиране и коментиране на кода

1. Коментари в програмния код на .NET Framework
Изберете два типа в .NET Framework reference source в github-хранилището: (<http://referencesource.microsoft.com/>).
Разгледайте всички коментари по кода, както и документацията. Лесен ли е за четене и разбиране? Има ли безполезни / безсмислени коментари? Кодът самоописателен ли е? Самоописателен код не означава код, в който няма коментари. В големите компании документирането на public интерфейса се изисква. Интерфейсите дават възможност на други разработчици да разширят и използват настоящия код. Понякога цялостната цел на един интерфейс не може да се разясни само чрез имената на членовете в него. Точно тук много помага документацията.
2. Документирайте изпитна задача
Отворете <https://github.com/desko-petkova/RS2025/tree/main/Code/06-Refactoring/StudentJsonTxt>
 1. Добавете коментари където трябва, за да направите кода по-лесен за четене и поддръжка.
 2. Добавете цялостна XML документация: хедър на файла, всички класове, методи и т.н.
 3. Инсталирайте StyleCop от NuGet Package Manager -> StyleCop.Analyzers и се уверете, че кодът не нарушава никоя от конвенциите.
3. Документиране на интерфейси Дадено ви е C# решение, което съдържа няколко интерфейса и класа, изграждащи проста игра „Змия“. Не променяйте нищо в кода. Задачата ви е да документирате интерфейсите IDrawable, IMovable и IKeyboardHandler. Документацията на C# код е много подобна на коментари в кода. Важната разлика е, че тя се използва от компилатора да генерира подсказки за документираните класове/интерфейси/методи.
Съвет: писането на /// генерира основните xml тагове, необходими за документацията.
4. Разширения на низове Даден ви е файл с програмен код от Visual Studio, наречен StringExtensions.cs. Създайте XML документация за файла, всички класове и техните членове. Добавете подходящи коментари в кода на местата, където това би подобрило четливостта и разбирането му.