




Как ти звучи да имам един тестов проект и в него да създам отделни тестови класове.



Създаването на един тестов проект, в който добавяте отделни тестови класове за всяка част от вашето приложение, е добра практика и често срещан подход при разработката на софтуер. Този начин на структуриране на тестовете ви позволява да поддържате организация и прегледност, като в същото време имате едно място за всички тестове.

Един тестов проект би могъл да включва следната структура:

markdown

 Copy code

- **MyProject.Tests** (тестовият проект)
 - **UnitTests** (директория за юнит тестовете)
 - **Class1Tests.cs** (тестов клас за **Class1**)
 - **Class2Tests.cs** (тестов клас за **Class2**)
 - **IntegrationTests** (директория за интеграционните тестове)
 - **IntegrationTest1.cs**
 - **IntegrationTest2.cs**
 - **FunctionalTests** (директория за функционалните тестове)
 - **FunctionalTest1.cs**
 - **FunctionalTest2.cs**

Този подход е често използван и се счита за добра практика при създаването на тестове за софтуерни приложения.



Как е най-добре да си кръстя тестовия проект, т.е. - `MyProject.Tests` (тестовият проект), ако основния ми проект се казва примерно `Skeleton`?



Когато кръстите тестовия проект за основния си проект (например "`Skeleton`"), е важно да изберете име, което е информативно и лесно за разбиране, когато гледате структурата на решението си. Възможни са няколко начина за наименование на тестовия проект:



1. **`Skeleton.Tests`**: Това е доста обичайната конвенция и ясно показва, че това е тестовият проект за "`Skeleton`" проекта. Преимуществото на този начин на именуване е ясното съответствие между основния проект и тестовия проект.
2. **`Skeleton.UnitTests`**: Ако планирате да разделяте тестовете си на юнит тестове и интеграционни тестове, тогава можете да използвате подпроекти като "`Skeleton.UnitTests`" за юнит тестове и "`Skeleton.IntegrationTests`" за интеграционни тестове. Това оставя пространство за организация и показва целта на тестовия проект.
3. **`Skeleton.TestSuite`**: Този вариант може да бъде подходящ, ако планирате да имате много тестови проекти, свързани с основния ви проект, и искате да ги групирате в общо `TestSuite` решение. Например, ако имате и други свързани проекти към "`Skeleton`," можете да ги организирате в `TestSuite` заедно с тестовия проект.

Изборът на име за тестовия проект зависи от вашите предпочитания и структурата на вашето решение. Главното е да бъде информативно и да помага на разработчиците да разберат неговата цел, когато разглеждат проекта.



Как е добре да кръстя тестов клас, който тества клас с име `Axe`?



Когато кръщавате тестов клас, който тества клас с име "Axe," е важно да използвате информативно име, което показва какво се тества и какви сценарии се проверяват. Ето няколко примерни начина за именуване на тестовия клас:

1. **AxeTests**: Този начин на именуване е обичайният подход и показва, че този клас съдържа тестове за класа "Axe." Това е ясно и лесно за разбиране.
2. **AxeUnitTest**: Ако планирате да разделяте тестовете си на юнит тестове и интеграционни тестове, можете да добавите "Unit" към името, за да посочите, че това са юнит тестове за "Axe."
3. **AxeFunctionalityTests**: Ако искате да подчертаете, че този клас тества различни аспекти на функционалността на "Axe," можете да използвате "FunctionalityTests" или подобно име.
4. **AxeTestSuite**: Ако имате няколко тестови класове, свързани с "Axe" и искате да ги групирате заедно, можете да използвате "TestSuite" или подобно име.

Изборът на име за тестовия клас трябва да бъде базиран на принципите за четимост и яснота, така че всеки, който разглежда вашите тестове, да може бързо да разбере какво се тества.



Мога ли да използвам точка в името на класа, примерно `Test.Axe`?



Не е препоръчително да използвате точка (".") в името на класа, включително в името на тестов клас. В много съвременни програмни езици и инструменти, точката се използва за обозначаване на `namespaces` (пространства от имена) и обекти в програмния код, и използването ѝ в имената на класове може да доведе до проблеми и недоразумения.

