



Ето табличка с най-често срещаните изключения и техните описания:

Изключение	Описание
<code>ArgumentException</code>	Некоректни аргументи, предадени на метод.
<code>ArgumentNullException</code>	Аргумент, който не трябва да бъде <code>null</code> , е получен като <code>null</code> .
<code>InvalidOperationException</code>	Нелегална операция в даден контекст.
<code>NullReferenceException</code>	Опит за достъп до обект, който е <code>null</code> .
<code>IndexOutOfRangeException</code>	Опит за достъп до невалиден индекс в масив или колекция.

Описание кога се случват:

- 1. `ArgumentException`** Това изключение се хвърля, когато методът получи некоректен аргумент, който не попада в обхвата на очакваните стойности. Например, ако методът приема индекс, който трябва да бъде положителен, а се подаде отрицателен, може да се хвърли `ArgumentException`. Често се използва, когато стойността на параметъра е извън допустимия диапазон, но не е конкретно свързана с `null` стойности.
- 2. `ArgumentNullException`** Това изключение се хвърля, когато аргументът, който не трябва да бъде `null`, е подаден като `null`. Обикновено това се случва, когато методът изисква референтен тип (например обект или низ), и предаването на `null` би довело до грешки при работа с този аргумент. Например, ако метод очаква валиден обект и получи `null`, той ще хвърли `ArgumentNullException`.
- 3. `InvalidOperationException`** Това изключение се хвърля, когато текущата операция не е допустима в конкретния контекст. Например, опит за премахване на елемент от празна колекция или опит за промяна на неизменяем обект може да доведе до `InvalidOperationException`. То се използва за ситуации, когато операцията сама по себе си е валидна, но не може да бъде изпълнена при текущото състояние на обекта.
- 4. `NullReferenceException`** Това изключение се хвърля, когато кодът се опитва да използва обект, който е `null`. Например, ако имаш обект, който не е инициализиран (т.е., неговата стойност е `null`), и се опиташ да достъпиш негови членове (методи, полета, свойства), ще бъде хвърлено `NullReferenceException`. Това е често срещана грешка при работа с обекти, които не са коректно създадени или инициализирани.
- 5. `IndexOutOfRangeException`** Това изключение се хвърля, когато се прави опит за достъп до елемент в масив или колекция с индекс, който е извън границите на допустимите индекси. Например, ако опиташ да достъпиш елемент с индекс 10 в масив, който има само 5 елемента, ще бъде хвърлено `IndexOutOfRangeException`.

Демо как да прихванем изключение в тест

```
public void Axe_Attack_WithZeroDurability_ShouldThrowException()
{
    // Arrange
    Axe axe = new Axe(10, 0); //създавам изхабена брадва
    Dummy dummy = new Dummy(100, 100);

    // Act
    try
    {
        axe.Attack(dummy); //ако има изключение автоматично
        //се прехвърля изпълнението в първият catch блок!

        //не би се изпълнила ако някъде от горе се хвърли изключение
        Assert.Fail("Метода не хвърли изключение");
    }
    catch (IndexOutOfRangeException ex)
    {
        // Assert type of exception
        Assert.That(ex, Is.TypeOf<InvalidOperationException>());
        //InvalidOperationException

        // Assert the message
        Assert.That(ex.Message, Is.EqualTo("Axe is broken"));
    }
}
```

- Изключенията носят в себе си string Свойство Message.
- Вторият Assert проверява какъв точно надпис се съдържа в него. Разбира се не е задължително да проверяваме надписа, ако не ни интересува точността му