

Компонентно тестване: утвърдени практики

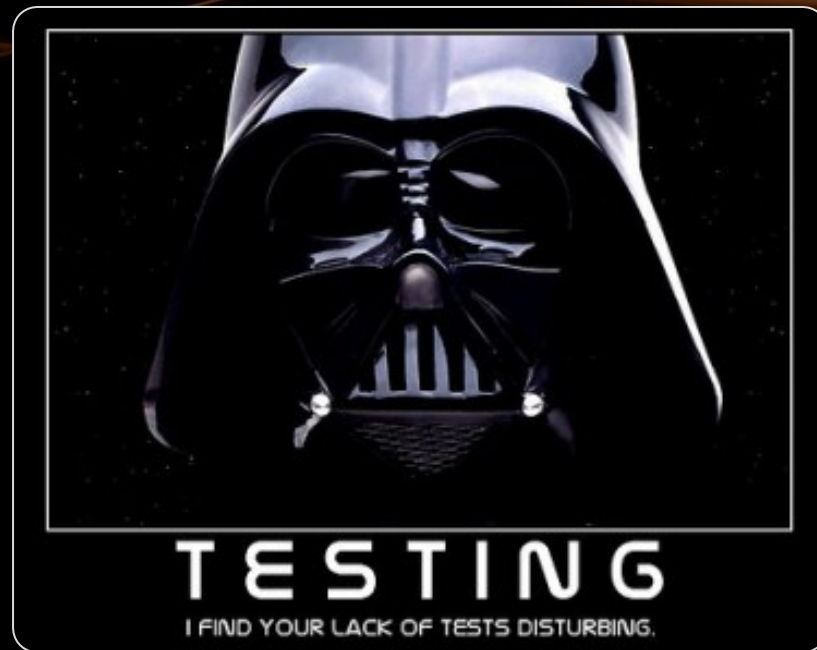


Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>





Добри практики

Как да пишем добри тестове?

Твърдения

- Условно твърдение

```
Assert.IsTrue(bool condition, string message);
```

- Сравнително твърдение

```
Assert.AreEqual(expected value, actual value);
```

- Твърдение за проверка на изключение

```
Assert.Throws(  
Type expectedExceptionType, TestDelegate code);
```

Твърдения (2)

- Низово твърдение

```
StringAssert.Contains  
    (string expected, string actual);
```

- Твърдение за колекция

```
CollectionAssert.Contains  
    (IEnumerable expected, object actual);
```

- Файлово твърдение

```
FileAssert.AreEqual  
    (FileInfo expected, FileInfo actual);
```


IsTrue() c/y AreEqual()

■ IsTrue(condition)

```
Assert.IsTrue(axe.DurabilityPoints == 12);
```

Дава само True/False
информация

✖ Test Failed - AxeLosesDurablyAfterAttack
Message: Expected: True
But was: False

■ AreEqual(expected, actual)

```
Assert.AreEqual(12, axe.DurabilityPoints);
```

По-добро описание, когато
очакваме стойност

✖ Test Failed - AxeLosesDurablyAfterAttack
Message: Expected: 12
But was: 9

Съобщения при твърдения

- Твърденията могат да **показват съобщения**
 - Помагат с **диагностиката**

```
Assert.AreEqual(12, axe.DurabilityPoints,  
    "Axe Durability doesn't change after attack");
```

Помага за намирането
на проблема

✖ Test Failed - AxeLosesDurabilityAfterAttack
Message: Axe Durability doesn't change after attack
Expected: 12
But was: 9

Магически числа

- Избягвайте употребата на магически числа

```
private const int axeDurability = 10;  
private const int axeAttack = 10;
```

```
[Test]  
public void AxeLosesDurabilityAfterAttack()  
{  
    Axe axe = new Axe(axeDurability, axeAttack);  
  
    Assert.AreEqual(axeDurability, axe.DurabilityPoints, "...");  
}
```

Не се повтаряйте

```
private BankAccount account;
```

```
[SetUp]
```

Изпълнява се
преди всеки тест

```
public void TestInit()
```

```
{
```

```
    this.account = new BankAccount();
```

```
}
```

```
[TearDown]
```

Изпълнява се
след всеки тест

```
public void TestCleanUp() {}
```


Именуване на методите в тестовете

- Имената на тестовете трябва:
 - Да ползват **подходяща терминлогия**
 - Да бъдат **описателни** и **четими**

```
IncrementNumber() {}  
Test1() {}  
TestTransfer() {}
```



```
DepositAddsMoneyToBalance() {}  
DepositNegativeShouldNotAddMoney() {}  
TransferSubtractsFromSourceAddsToDestAccount() {}
```



Задача: Рефакториране на тестове

- Рефакторирайте тестовете за **Axe** и **Dummy** класовете
- Уверете се, че:
 - **Имената** на тест методите са **описателни**
 - Използвайте **подходящи твърдения** (assert equals c/y assert true)
 - Използвайте **съобщения за твърденията**
 - Няма **магически числа**
 - Няма **повторение на кода** (Не се повтаряйте)

Решение: Рефакториране на тестове

```
private const int AxeAttack = 2;
private const int AxeDurability = 2;
private const int DummyHealth = 20;
private const int DummyXP = 20;
private Axe axe;
private Dummy dummy;

[SetUp]
public void TestInit()
{
    this.axe = new Axe(AxeAttack, AxeDurability);
    this.dummy = new Dummy(DummyHealth, DummyXP);
}
```

Решение: Рефакториране на тестове (2)

```
[Test]
public void AxeLosesDurabilityAfterAttack()
{
    axe.Attack(dummy);
    Assert.AreEqual(1, axe.DurabilityPoints,
        "Axe Durability doesn't change after attack");
}

[Test]
public void BrokenAxeCantAttack()
{
    axe.Attack(dummy);
    axe.Attack(dummy);
    var ex = Assert.Throws<InvalidOperationException>(
        () => axe.Attack(dummy));
    Assert.That(ex.Message, Is.EqualTo("Axe is broken.")).
}
```


Компонентно тестване: утвърдени практики



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

