

Testes de Unidade

Usando JUnit



Primeiro Exemplo (1/2)

```
import junit.framework.TestCase;

/**
 * Testes de unidade para a classe {@link Pessoa}
 */
public class TestPessoa extends TestCase {

    /**
     * Um teste de unidade para verificar se o nome está
     * formatado corretamente
     */
    public void testObterNomeCompleto() {
        Pessoa p = new Pessoa("Fulano", "Tal");
        assertEquals("Fulano Tal", p.getNomeCompleto());
    }
}
```

Primeiro Exemplo (2/2)

```
/**
 * Um teste de unidade para verificar se os nulos são
 * tratados corretamente
 */
public void testNomeEhNulo() {
    Pessoa p = new Pessoa(null, "Tal");
    assertEquals("? Tal", p.getNomeCompleto());

    // Este código só é executado se passar pelo
    // assertEquals anterior.
    p = new Pessoa("Fulano", null);
    assertEquals("Fulano ?", p.getNomeCompleto());
}
}
```

Executando JUnit

❑ Teste com falhas

```
java -cp junit.jar;. junit.textui.TestRunner TestPessoa
```

```
.F.F
```

```
Time: 0.02
```

```
There were 2 failures:
```

```
1)
```

```
testObterNomeCompleto(TestPessoa)junit.framework.AssertionFailedError: expected:<Fulano Tal> but was:<FulanoTal>  
    at TestPessoa.testObterNomeCompleto(TestPessoa.java:14)
```

```
2) testNomeEhNulo(TestPessoa)junit.framework.
```

```
AssertionFailedError: expected:<? Tal> but was:<?Tal>  
    at TestPessoa.testNomeEhNulo(TestPessoa.java:22)
```

```
FAILURES!!!
```

```
Tests run: 2, Failures: 2, Errors: 0
```

Executando JUnit

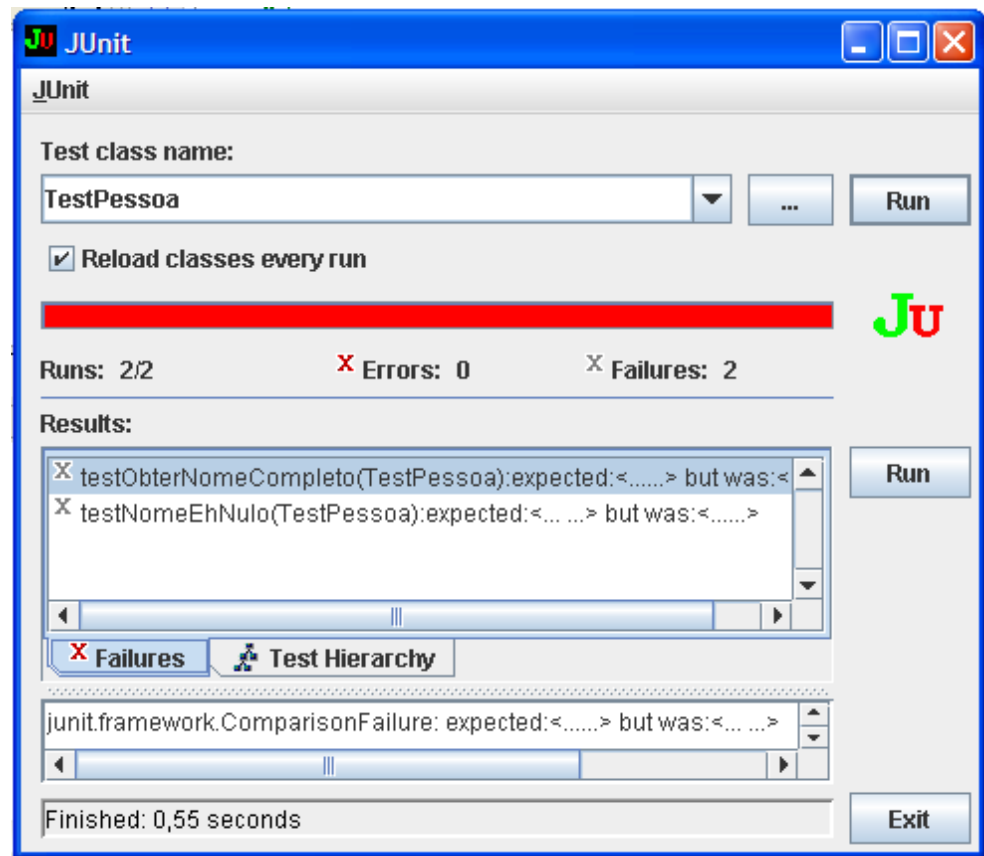
- Depois de corrigir os erros da classe Pessoa

```
java -cp junit.jar;. junit.textui.TestRunner TestPessoa
```

```
..  
Time: 0.01  
OK (2 tests)
```

Usando o Ambiente Gráfico

```
java -cp junit.jar;. junit.swingui.TestRunner TestPessoa
```



Resumo dos Métodos Assert

Método	Descrição	Teste passa se
<code>assertEquals(a,b)</code>	Compara dois valores	<code>a.equals(b)</code>
<code>assertFalse(a)</code>	Avalia uma expressão booleana	<code>a == false</code>
<code>assertTrue(a)</code>		<code>a == true</code>
<code>assertNotNull(a)</code>	Compara uma variável com nulo	<code>a != null</code>
<code>assertNull(a)</code>		<code>a == null</code>
<code>assertNotSame(a,b)</code>	Compara dois objetos	<code>a == b</code>
<code>assertSame(a,b)</code>		<code>a != b</code>
<code>fail()</code>	Causa uma falha no teste atual	

Usando os Métodos Assert

□ Sintaxe

- `assertEquals(valorEsperado, valorTestado);`
- `assertEquals("Mensagem indicando o que deveria ocorrer", valorEsperado, valorTestado);`

□ Exemplo

```
Pessoa p = new Pessoa("Fulano", "Tal");  
assertEquals(  
    "Deveria haver um espaço entre os nomes",  
    "Fulano Tal", p.getNomeCompleto());
```


Exemplo: Jogo da Velha

```
public class velha {  
    public velha(){}  
    public char getGanhador(){}  
    public boolean getFim(){}  
    public void setMarcacao(int posicao){}  
    public void voltar(){}  
    public char[] getMarcacao(){}  
    public char getJogadorVez(){}  
    public void iniciar(){}  
}
```

Testando o Jogo da Velha (1/3)

```
import junit.framework.TestCase;
/**
 * Teste de Unidade para a classe {@link Velha}.
 */
public class TesteVelha extends TestCase {
    /**
     * Teste de unidade para verificar se o jogador da vez está
     * correto após uma jogada.
     */
    public void testPassarVez(){
        Velha jogo = new Velha();
        jogo.setMarcacao(0);
        char jogadorAntes = jogo.getJogadorVez();
        jogo.setMarcacao(3);
        char jogadorDepois = jogo.getJogadorVez();
        assertEquals("O jogador da vez deve mudar após cada jogada",
            jogadorAntes, jogadorDepois);
    }
}
```

Testando o Jogo da Velha (2/3)

```
/**
 * Teste de unidade para verificar se o jogo não chegou ao fim
 */
public void testNaoFim(){
    String msg = "O jogo só deve chegar ao fim quando alguém " +
        "ganhar ou quando todas as posições " +
        "estiverem preenchidas";
    Velha jogo = new Velha();
    assertFalse(msg, jogo.getFim());
    jogo.setMarcacao(0);
    assertFalse(msg, jogo.getFim());
    jogo.setMarcacao(1);
    assertFalse(msg, jogo.getFim());
    jogo.setMarcacao(2);
    assertFalse(msg, jogo.getFim());
    jogo.setMarcacao(3);
    assertFalse(msg, jogo.getFim());
}
```

Testando o Jogo da Velha (3/3)

```
/**
 * Teste de unidade para verificar se é possível marcar uma
 * posição fora da faixa
 */
public void testMarcacaoInvalida(){
    jogo.setMarcacao(-1);
    jogo.setMarcacao(10);
}
```

Granularidade dos Testes

- ❑ Cada teste deve verificar um pedaço específico da funcionalidade
- ❑ Não combine testes não relacionados em um único método `testXXX()`
- ❑ Se o primeiro teste falhar os seguintes não serão executados

Métodos setUp() e tearDown()

- Agrupam código de iniciação e de finalização usados por todos os casos de teste
- Seqüência de execução do JUnit:
 - Cria uma instância da classe de teste para cada método de teste. (Exemplo: 5 testes, 5 instâncias).
 - Para cada instância:
 - Chama o método setUp();
 - Chama o método de teste;
 - Chama o método tearDown();

Usando setUp()

```
import junit.framework.TestCase;
public class TesteVelha extends TestCase {
    private Velha jogo;
    public void setUp() {
        jogo = new Velha();
        jogo.setMarcacao(0); jogo.setMarcacao(1); jogo.setMarcacao(2);
    }

    public void testPassarVez(){
        char jogadorAntes = jogo.getJogadorVez();
        jogo.setMarcacao(3);
        char jogadorDepois = jogo.getJogadorVez();
        assertNotSame("O jogador da vez deve mudar após cada jogada",
            jogadorAntes, jogadorDepois);
    }
    public void testNaoFim(){
        String msg = "O jogo só deve chegar ao fim quando alguém ..."
        assertFalse(msg, jogo.getFim());
    }
}
```