

Trabalho - JUnit

Integrantes:

Gabriel Alejandro Figueiro Galindo

Eric Jardim

Nando Augusto Veloso

Yghor Ribas Gomes

Descrição:

O JUnit é um framework open-source de testes unitários para a linguagem de programação Java. Ele faz parte da família de estruturas de teste de unidade conhecidas coletivamente como xUnit, tendo um papel de grande importância na história do desenvolvimento orientado a testes.

Quem criou/desenvolveu:

O JUnit foi principalmente desenvolvido por Kent Beck, Erich Gamma, David Saff e Kris Vasudevan.

Principais funcionalidades:

O JUnit fornece uma estrutura para organizar testes em classes e métodos, permitindo a escrita de testes de unidade de maneira estruturada.

Ele utiliza anotações para marcar métodos como testes. Entre as anotações mais comuns, pode ser notado a anotação `@Test`, a anotação `@Before` e a anotação `@After`.

O JUnit utiliza asserções, como `assertEquals`, `assertTrue` e `assertFalse`, para verificar se as condições esperadas são verdadeiras. Se uma asserção falhar, o teste é considerado um fracasso.

Quem utiliza (grandes empresas, projetos, etc, se houver):

O JUnit é amplamente utilizado por uma variedade de empresas, desde pequenas startups até grandes corporações. Ele se tornou um padrão na comunidade de desenvolvimento de software, especialmente na plataforma Java.

O JUnit também é utilizado em projetos de código aberto, projetos de desenvolvimento de jogos, projetos de IoT e projetos de software embarcado.

Principais vantagens:

O framework do JUnit facilita a criação e manutenção do código para a automação de testes com apresentação dos resultados, permitindo a execução de testes repetidamente sem a necessidade de intervenção manual.

A maioria das IDEs populares oferece suporte integrado para o JUnit, tornando mais fácil executar, depurar e gerenciar testes diretamente do ambiente de desenvolvimento.

O JUnit também fornece um feedback imediato sobre o sucesso ou a falha dos testes, além de ser amplamente adotado na comunidade Java e em muitas outras linguagens

de programação, sendo o padrão para testes de unidade na indústria.

Principais desvantagens:

O JUnit é projetado principalmente para testes de unidade, o que significa que ele não é a ferramenta mais adequada para testes de integração que envolvem várias partes do sistema. Outra desvantagem é que testar códigos que envolvem múltiplas threads e concorrência pode ser difícil com o JUnit, uma vez que ele não fornece recursos avançados para testes multithread.

Por último, como o JUnit tem como o foco testes de unidade, eles cobrem principalmente a lógica interna de métodos e classes. Eles podem não cobrir aspectos mais amplos, como a interface do usuário ou integração com outros sistemas.

Indicações de Uso:

As principais indicações de uso incluem: O teste unitário, o TDD, Testes de Regressão e Testes de Integração Parcial.

Categorização:

O JUnit é uma ferramenta destinada à realização de testes de unidade, que são um nível de teste.

Instalação:

Existem muitas maneiras de utilizar o JUNIT 5, este passo a passo demonstra como utilizá-lo no Visual Studio Code.

Requisitos: Ter o Java instalado em sua máquina. Basta utilizar o comando “java -version” para verificar a versão instalada do Java.

Utilizando o extension pack for java:

Passo 1: Na aba extensões do VsCode busque por “Extension Pack for Java” e instale-a.

Passo 2: Entre na aba “Testing” do seu VsCode e clique em Enable Java Tests. Selecione JUnit Jupiter, uma pasta de nome lib será criada.

Sem o extension pack for java:

Passo 1: Para realizar o download acesse o site oficial do framework (<https://junit.org/junit5/>) e faça o download do arquivo JAR.

Passo 2: Copie o arquivo JAR que você baixou para o diretório do seu projeto. Crie uma pasta chamada lib e coloque o arquivo JAR lá.

Passo 3: Configure o seu IDE para incluir o arquivo JAR do JUnit no classpath do seu projeto. No VsCode você pode precisar editar o arquivo settings.json.

Tudo pronto para utilizar o JUnit no seu projeto no Vscode, basta criar uma classe de teste e importar as classes e anotações necessárias do JUnit.

Exemplos de Uso:

Alguns dos exemplos de uso do Junit são:

1- Testar igualdade:

```
@Test
public void testSoma() {
    Calculadora calc = new Calculadora();
    int resultado = calc.soma(10, 20);
    assertEquals(30, resultado);
}
```

2- Testar valores nulos:

```
@Test
public void testBuscarUsuarioInexistente() {
    Sistema sistema = new Sistema();
    Usuario usuario = sistema.buscarUsuario("eric");
    assertNull(usuario);
}
```

```
}
```

3- Testar valores não nulos:

```
@Test  
public void testCriarObjeto() {  
    Object obj = new Object();  
    assertNotNull(obj);  
}
```

4- Testando condições verdadeiras/falsas:

```
@Test  
public void testPar() {  
    Numero num = new Numero();  
    assertTrue(num.ePar(4));  
    assertFalse(num.ePar(5));  
}
```