# Testes de unidade com JUnit

FURB
UNIVERSIDADE DE BLUMENAU

# Bibliografia


- MASSOL, V.; HUSTED, T. **JUnit em Ação**. Nova York: Manning Publications, 2003.

- HIGHTOWER, R.; LESIECKI, N. **Java Tools for Extreme Programming:** Mastering Open Source Tools, Including Ant, JUnit, and Cactus. New York: Wiley Computer Publishing, 2002.

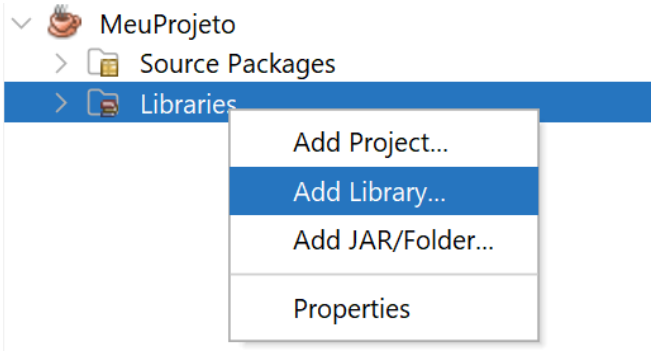
- OBJECT MENTOR. **JUnit.org - Resources for Test Driven Development**. Disponivel em: <http://www.junit.org/>. Acesso em: 15 dez. 2011.

# Sobre o JUnit

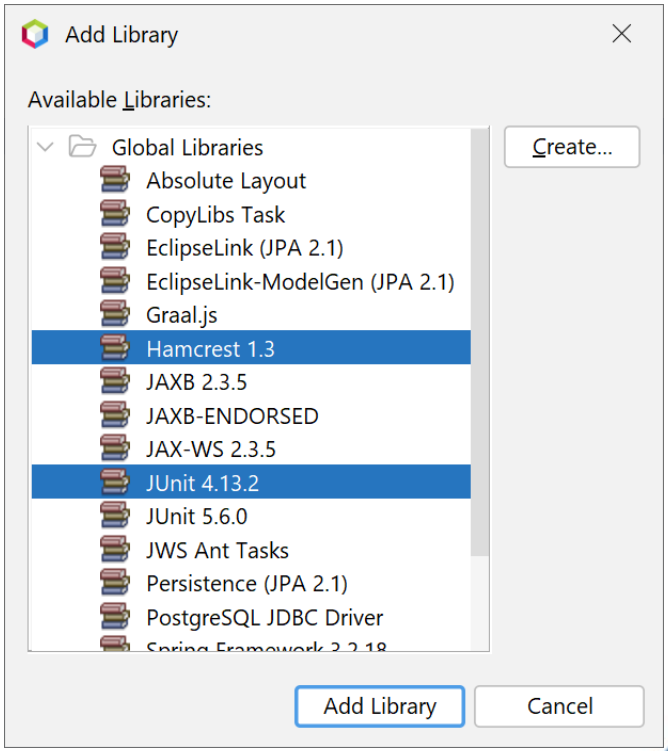
- JUnit é um framework para execução de teste de unidade em Java
- Criado em 1997 por Erich Gamma e Kent Beck
- Hospedado no Source Forge (depreciado) e GitHub
- Possui três implementações (versão 3, versão 4 e versão 5)
- Distribuído sob a licença Eclipse Public License
  - Versão 4 usa a versão 1.0 da licença, já a versão 5 utiliza a versão 2.0 da licença;

# Como testar com Junit no NetBeans

- Para habilitar a execução de testes, clicar com o botão direito em “Libraries” e selecionar “Add Library”




- Selecionar as bibliotecas:
  - Hamcrest 1.3 e
  - JUnit 4.13.2



# Como testar com JUnit no Netbeans

- Clicar em Arquivo > Novo arquivo
- Em “Categoria”, selecionar “Unit Tests”
- Em “File Types”, selecionar “Test for Existing Class”
- Selecionar a classe a ser testada
- Desmarcar todas as caixas de seleção
- Clicar em Finalizar

# Testes com JUnit

- Como vamos implementar testes para a camada de negócio, haverá uma classe “paralela” (de teste) para cada classe a ser testada
- Em geral, o nome desta classe é igual ao nome da classe a ser testada, porém com sufixo “Test”

- Para cada “caso de teste”, criar um método e introduzir a anotação `@Test`
  - O método deve ser público e não pode ter parâmetros
  - O método não pode retornar dados (void)
  - Dentro deste método, deve-se utilizar um comando *assert*, para validar uma situação

# Como testar com JUnit

- Exemplo:

```
 1 import static org.junit.Assert.*;
 2 import org.junit.*;
 3
 4 public class CalculadoraTest {
 5
 6     @Test
 7     public void test1_SomarNumeros() {
 8         Calculadora calc = new Calculadora();
 9         int resultado = calc.somar(10,75);
10         assertEquals(85, resultado);
11     }
12
13 }
```

Valor esperado

Valor obtido

"Espero que o somar() resulte em 85"

# Métodos *Assert*

- Os métodos *assert* são utilizados para verificar se uma condição é verdadeira. Caso a verificação não seja bem sucedida, é lançada uma exceção

| Método | Descrição |
|---|---|
| `assertEquals(long esperado, long real)` | Verifica se os dois números inteiros são iguais |
| `assertEquals(double esperado, double real, double limite)` | Verifica se dois números decimais são iguais, sem atingir o limite de diferença |
| `assertEquals(objetoEsperado, objetoReal)` | Verifica se dois objetos são iguais |
| `assertTrue(boolean condição)` | Verifica se a condição é verdadeira |
| `assertFalse(boolean condição)` | Verifica se a condição é falsa |
| `assertNotNull(objeto)` | Verifica se a variável referencia um objeto |
| `assertNull(objeto)` | Verifica se a variável não referencia um objeto |

# Casos de teste com o mesmo contexto

- Quando dois ou mais casos de teste compartilham o mesmo contexto, é possível definir o contexto num método que contém a anotação *@Before*

```
@Before
public void inicializarContexto() {
    calc = new Calculadora();
}
```

- Igualmente, é possível definir um método para ser executado após cada caso de teste, com a anotação *@After*

```
@After
public void finalizarContexto() {
    arquivo.close();
}
```

# Exceções

- A anotação @Test permite utilizar o parâmetro “expected” para implementar casos de teste que esperam que ocorra uma determinada exceção

```
1 @Test(expected=IllegalArgumentException.class)
2 public void test013() throws Exception {
3     imp.setSalario(-100);
4 }
```

# Tempo limite de execução

- Para limitar o tempo máximo de execução de um teste, pode ser utilizado o parâmetro *timeout*

```
1 @Test(timeout = 1000)
2 public void testLoadUserCfg() {
3     db.setUser("teste");
4     db.loadUserConfig();
5 }
```