

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

## Prática

### Testes Unitários.

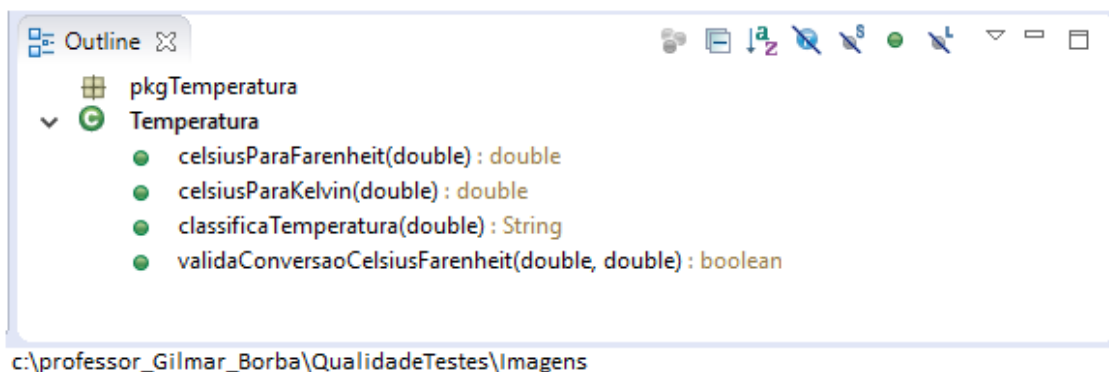
#### Objetivos:

- Praticar a partir de um exemplo simples os **Testes Unitários** para métodos específicos de uma classe.
- Entender o processo de criação de **Testes Unitários automatizados**.
- Conhecer as diretivas: @test, @after, @before.
- Conhecer e trabalhar com os métodos: assertEquals(), assertTrue() ... assertFalse(), @assertNull(), @assertNotNull(), @assertSame(), @assertNotSame(), setUp() e tearDown().

#### Considerações sobre a atividade:

A classe possui métodos de conversão de temperatura e classificação de temperatura.

Veja os métodos da classe:



A CLASSE **TEMPERATURA** SERÁ A CLASSE A SER TESTADA

#### Orientações:

(A) Crie um novo projeto JAVA: **TemperaturaTesteUnitario**

(B) Crie um pacote (com um nome a seu critério)

(C) Crie a classe **Temperatura**, copie cole a classe. Ajuste o nome do Pacote.

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

(D) Criar a Classe de Teste, siga os passos:

- 1 - Selecionar o pacote
- 2 – Acionar o botão direito do mouse (menu)
- 3 – Selecionar a opção **New**
- 4 – Selecionar a opção **JUnit Test Case**

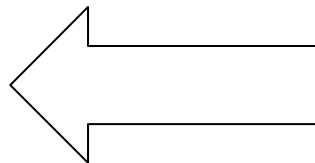
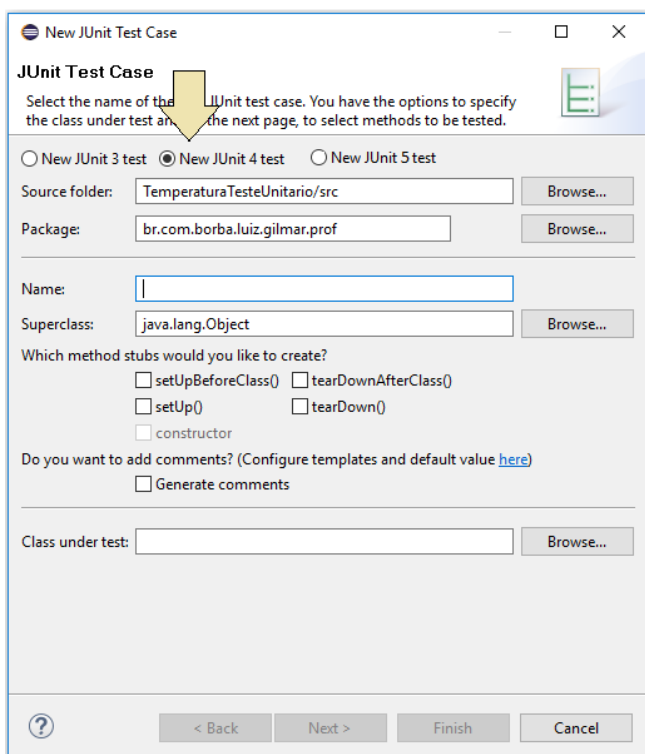
(E) Uma vez na caixa de diálogo **New JUnit Test Case** inserir as seguintes informações:

- 1 – Name: **TemperaturaTESTE** (nome da classe de teste)
- 2 – Deixar selecionado somente o método setUp() na opção:  
Which method stubs would you like to create?
- 3 – Na opção “Class Under test”, acionar o botão [Browse] informe **Temperatura**, para localizar e definir a classe **Temperatura** do pacote do nosso projeto.
- 4 – Após localizar a classe “**Temperatura**” do nosso pacote acione o botão [OK]
- 5 - Ao retornar a tela anterior (**New JUnit Test Case**) Acione o botão [NEXT]

## ATENÇÃO:

Trabalharemos com a versão 4 do Framework Junit!

NÃO ESQUECER DE DE HABILITAR A OPÇÃO **New Junit 4 Text**. Mesmo procedimento para as demais atividades.

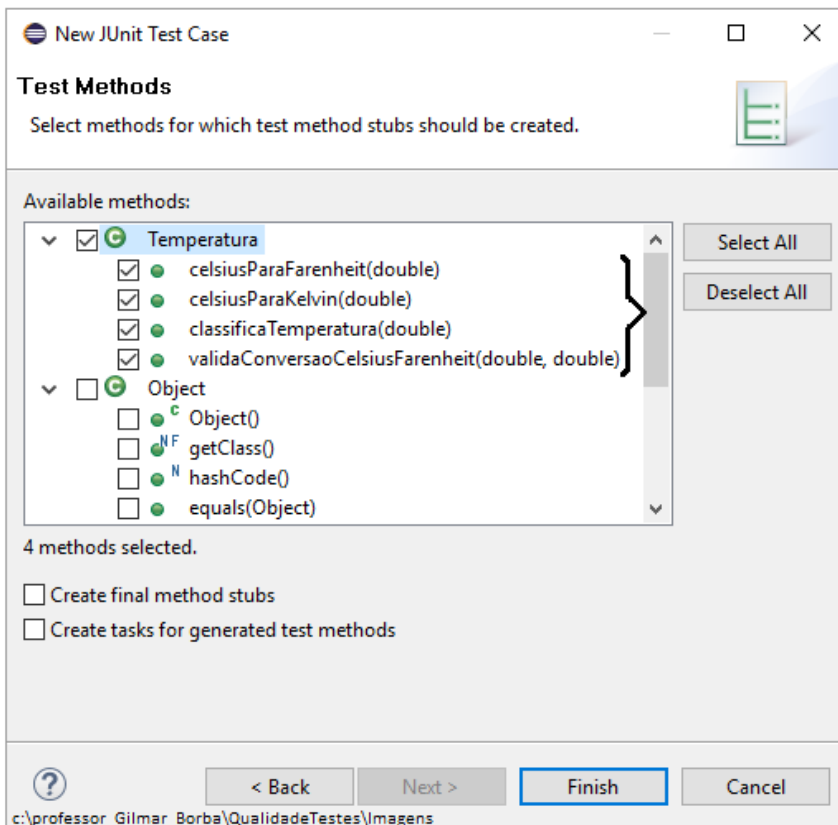


(F) Escolha os métodos que deverão ser testados:

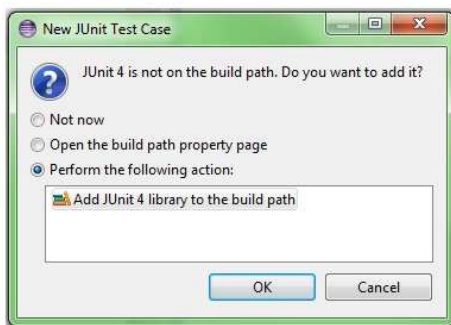
Selecionar todos os **Métodos** / Acione o botão **[FINISH]**

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba



(G) Na primeira vez que for criada uma classe de teste será questionado sobre o caminho (build Path) de procura do JUnit 4 . Deixe a opção: ***“Perform the following action”*** selecionada e aciona o botão [OK]



Nesse momento foi gerada a classe de teste

(H) Criar o tipo objTemp e o objeto de mesmo nome instanciado no método setUp().

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

```
1 package pkgTemperatura;  
2  
3 import static org.junit.Assert.*;  
4  
5  
6  
7  
8 public class TemperaturaTeste {  
9  
10     Temperatura objTemp;  
11     @Before  
12     public void setUp() throws Exception {  
13         objTemp = new Temperatura();  
14     }
```

c:\professor\_Gilmar\_Borba\QualidadeTestes\Imagens

**(I) Alterar o nome dos métodos de teste para** (veja imagem):

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

```
1 package pkgTemperatura;
2 import static org.junit.Assert.*;
3 import org.junit.Before;
4 import org.junit.Test;
5 public class TemperaturaTeste {
6
7     Temperatura objTemp;
8     @Before
9     public void setUp() throws Exception {
10         objTemp = new Temperatura();
11     }
12
13     @Test
14     public void deveriaConverterCelsiusParaFahrenheit() {
15         fail("Not yet implemented");
16     }
17
18     @Test
19     public void deveriaConverterCelsiusParaKelvin() {
20         fail("Not yet implemented");
21     }
22
23     @Test
24     public void deveriaClassificarTemperatura() {
25         fail("Not yet implemented");
26     }
27
28     @Test
29     public void deveriaValidarConversaoCelsiusFahrenheit() {
30         fail("Not yet implemented");
31     }
32 }
33
```

c:\professor\_Gilmar\_Borba\QualidadeTestes\Imagens

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

**(J) Implementar os seguintes testes** (*use assertEquals()*):

Método de teste: deveriaConverterCelsiusParaFahrenheit()

Texto	Valor esperado (oF)	Parâmetro (oC)
CASO 1:	23	-5
CASO 2:	32	0
CASO 3:	64,40	18

**(K) Implementar os seguintes testes** (*use assertEquals()*):

Método de teste: deveriaConverterCelsiusParaKelvin()

Texto	Valor esperado (oF)	Parâmetro (oC)
CASO 4:	268.150	-5
CASO 5:	273.150	0
CASO 6:	291.150	18

**(L) Implementar os seguintes testes** (*use assertTrue()*):

Método de teste: deveriaClassificarTemperatura()

Texto	Valor esperado (retorno String)	Parâmetro (oC)
CASO 7:	NEGATIVA	-5
CASO 8:	ZERO	0
CASO 9:	POSITIVA	18

**(M) Implementar o seguinte teste** (*use assertTrue()*):

Método de teste: deveriaValidarConversaoCelsiusFahrenheit ()

Texto	Valor esperado (retorno boolean)	Parâmetro1 (oC)	Parâmetro1 (oF)
CASO 10:	True	-5	23