

 03
Rodando o JUnit

Vamos nos preparar para criar testes de unidade. Para isso, no projeto **fj15-empresa**, temos uma nova necessidade de separar os gastos que precisam ser aprovados pela gerência.

Crie a classe `FiltradorDeGastosEspeciais` com o seguinte método:

```
public List<Gasto> filtraGastosGrandes(List<Gasto> gastos) {  
    List<Gasto> gastosAcimaDoLimite = new ArrayList<Gasto>();  
    for (Gasto gasto : gastos) {  
        if(gasto.getValor() > this.limite)  
            gastosAcimaDoLimite.add(gasto);  
    }  
    return gastosAcimaDoLimite;  
}
```

Use o `ctrl + shift + O` para fazer os imports necessários, `ctrl + I` para criar o atributo `limite` e `ctrl + 3` constructor para criar o construtor que recebe o limite.

Agora, com o cursor nessa classe, faça `ctrl + N` JUnit Test Case e crie, aos moldes do explicado no capítulo, um teste de unidade que verifica que, dada uma lista de gastos, só devolve os gastos acima do limite. Não esqueça de colocá-lo na **source folder de testes!**. Além disso, ao tentar criar um *JUnit Test Case*, o eclipse o ajudará a importar a biblioteca do JUnit 4! Leia as janelas e selecione a opção que automaticamente já adiciona ao build-path.

```
public class FiltradorDeGastosTest {  
    @Test  
    public void deveRetornarGastosAcimaDoLimite() {  
        // Criação dos gastos  
        Gasto gasto1 = new Gasto("Aluguel", 1000.0);  
        Gasto gasto2 = new Gasto("Gasolina", 500.0);  
        Gasto gasto3 = new Gasto("Almoço", 200.0);  
        List<Gasto> gastos = Arrays.asList(gasto1, gasto2, gasto3);  
        // Instanciação do filtrador  
        FiltradorDeGastosEspeciais filtrador = new FiltradorDeGastosEspeciais();  
        // Executa o filtro  
        List<Gasto> gastosFiltrados = filtrador.filtraGastosGrandes(gastos);  
        // Verifica o resultado  
        assertEquals(1, gastosFiltrados.size());  
        assertEquals("Aluguel", gastosFiltrados.get(0).getDescricao());  
    }  
}
```

Preencha o método de teste com a criação de 3 gastos, um deles acima do limite estipulado, um abaixo e um igual.