



04

## Deserialização

Vamos agora a deserialização. Configure o xml base que utilizaremos para deserializar nossa compra e configure seu xstream:

```
@Test
public void deveGerarUmaCompraComOsProdutosAdequados() {
    String xmlDeOrigem = "<compra>\n"+
        "  <id>15</id>\n"+
        "  <produtos>\n"+
        "    <produto codigo=\"1587\">\n"+
        "      <nome>geladeira</nome>\n"+
        "      <preco>1000.0</preco>\n"+
        "      <descricao>geladeira duas portas</descricao>\n"+
        "    </produto>\n"+
        "    <produto codigo=\"1588\">\n"+
        "      <nome>ferro de passar</nome>\n"+
        "      <preco>100.0</preco>\n"+
        "      <descricao>ferro com vaporizador</descricao>\n"+
        "    </produto>\n"+
        "  </produtos>\n"+
        "</compra>";
    XStream xstream = new XStream();
    // configure seu xstream aqui
}
```

Tente deserializar:

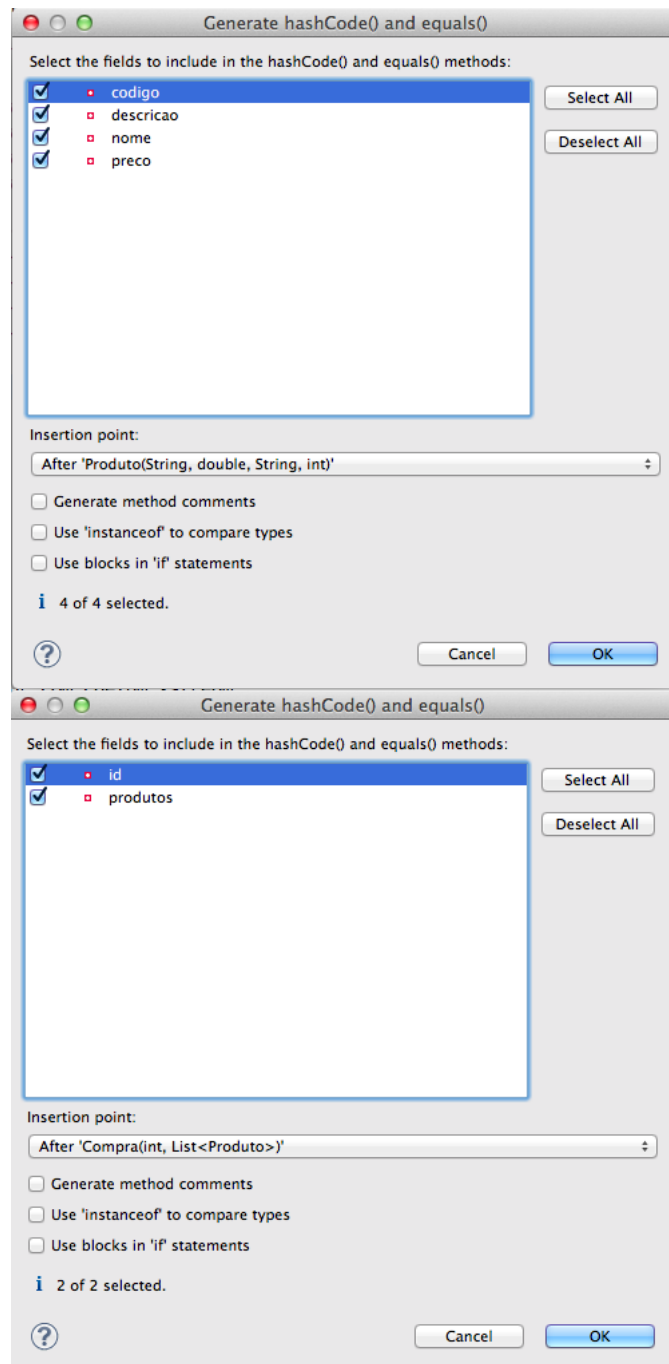
```
Compra compraResultado = (Compra) xstream.fromXML(xmlDeOrigem);
```

E confira que a compraResultado é igual a compra que esperamos:

```
Produto geladeira = new Produto("geladeira", 1000,
    "geladeira duas portas", 1587);
Produto ferro = new Produto("ferro de passar", 100,
    "ferro com vaporizador", 1588);
List<Produto> produtos = new ArrayList<Produto>();
produtos.add(geladeira);
produtos.add(ferro);

Compra compraEsperada = new Compra(15, produtos);
assertEquals(compraEsperada, compraResultado);
```

Faltou implementarmos o equals, claro. Na classe Produto, CTRL+3, generate hashCode and equals, OK. Faça o mesmo na classe Compra:



Rode os testes. Compartilhe aqui o seu método de teste.

## Responda

INSERIR CÓDIGO	FORMATAÇÃO
<div></div>	

