

1. Nosso banco precisa tributar dinheiro de alguns bens que nossos clientes possuem. Para isso, vamos criar uma interface no nosso projeto **Banco** já existente:

```
interface Tributavel {  
    double calculaTributos();  
}
```

Tributavel.java

Lemos essa interface da seguinte maneira: "todos que quiserem ser tributável precisam saber calcular tributos, devolvendo um double".

2. Alguns bens são tributáveis e outros não. **ContaPoupanca** não é tributável, já para **ContaCorrente** você precisa pagar 1% da conta como tributo. Crie a classe **SeguroDeVida** também tributável.
3. Crie um **GerenciadorDeImpostoDeRenda**, que recebe todos os tributáveis (através do método **adicionar**) de uma pessoa e soma seus valores e inclua nele um método para devolver seu total.
4. Crie a classe **Banco**, que possui um método **adicionar** para adicionar contas. Faça **Conta** implementar a interface **Comparable** e implemente o método **ordenar** para ordenar as contas com base no **saldo**.
5. Na classe **Conta**, modifique o método **deposita(double x)**: Ele deve lançar uma **exception** chamada **IllegalArgumentException**, que já faz parte da biblioteca do Java, sempre que o valor passado como argumento for inválido (por exemplo, quando for negativo).
6. Crie uma classe **TestaDeposita** com o método **main**. Crie uma **ContaPoupanca** e tente depositar valores inválidos.
7. Adicione o **try/catch** para tratar o erro na classe **TestaDeposita** ao imprimir "Você tentou depositar um valor inválido" caso seja passado um valor inválido.
8. Ao lançar a **IllegalArgumentException**, passe via construtor uma mensagem a ser exibida. Lembre que a **String** recebida como parâmetro é acessível depois via o método **getMessage()** herdado por todas as **Exceptions**.
9. Crie sua própria **Exception**, **ValorInvalidoException**. Para isso, você precisa criar uma classe com esse nome que seja filha de **RuntimeException**. Faça o método **deposita** lançar essa exceção.
10. Coloque um construtor na classe **ValorInvalidoException** que receba valor inválido que ele tentou passar (isto é, ele vai receber um double valor). Reutilize o construtor da superclasse para definir a mensagem em função do valor.

11. Declare a classe `ValorInvalidoException` como filha direta de `Exception` em vez de `RuntimeException`. Ela passa a ser `checked`. O que isso resulta?

Divirta-se!