

## Mãos na massa: Criando a exceção

No projeto `bytebank-herdado-conta` vamos refatorar o código da classe `Conta` e criar uma nova exceção.

6) Primeiramente crie a nossa exceção `SaldoInsuficienteException` :

```
public class SaldoInsuficienteException extends Exception{ //checked

    public SaldoInsuficienteException(String msg) {
        super(msg);
    }
}
```

2) Agora abra a classe `Conta` . Procure o método `saca(..)` e troque o tipo do retorno de `boolean` para `void` . Remove os `return` s e lança a exceção nova. Nosso método `saca(..)` agora vai ter a seguinte cara:

```
public void saca(double valor) throws SaldoInsuficienteException{

    if(this.saldo < valor) {
        throw new SaldoInsuficienteException("Saldo: " + this.saldo + ", Valor: " + valor);
    }

    this.saldo -= valor;
}
```

Repare que invertemos a lógica para podermos lançar a exceção antes.

3) Veja que agora nosso método `transfere()` também precisa ser alterado, já que agora ele não terá mais um retorno do tipo `boolean` e que o método `saca()` não retorna `void` , além da exceção na assinatura do método.

Altere o método para ficar como:

```
public void transfere(double valor, Conta destino) throws SaldoInsuficienteException{
    this.saca(valor);
    destino.deposita(valor);
}
```

4) Por conta disso, precisaremos alterar nosso método `saca()` de nossa classe `ContaCorrente` :

```
@Override
public void saca(double valor) throws SaldoInsuficienteException {
    double valorASacar = valor + 0.2;
    super.saca(valorASacar);
}
```

5) Agora altere a classe `TesteContas` para funcionar com nossas exceções. Para tal adicione um "throws" na assinatura do método `main` :

```
public class TesteContas {  
  
    public static void main(String[] args) throws SaldoInsuficienteException{  
  
        ContaCorrente cc = new ContaCorrente(111, 111);  
        cc.deposita(100.0);  
  
        ContaPoupanca cp = new ContaPoupanca(222, 222);  
        cp.deposita(200.0);  
  
        cc.transfere(110.0, cp);  
  
        System.out.println("CC: " + cc.getSaldo());  
        System.out.println("CP: " + cp.getSaldo());  
    }  
}
```

Depois tente transferir um valor inválido e execute o código:

6) Por fim, crie uma nova classe `TesteSaca` para testar o método `saca`. Use um `try-catch` para capturar a exceção:

```
public class TesteSaca {  
  
    public static void main(String[] args) {  
        Conta conta = new ContaCorrente(123, 321);  
  
        conta.deposita(200.0);  
  
        try {  
            conta.saca(210.0);  
        } catch (SaldoInsuficienteException ex) {  
            System.out.println("Exception: " + ex.getMessage());  
            ex.printStackTrace();  
        }  
  
        System.out.println(conta.getSaldo());  
    }  
}
```

Execute a classe `TesteSaca` !