

## Exercícios - Tratamento de exceções

- 1) Pesquise na API Java os tipos de Exceção `IllegalArgumentException` e `NumberFormatException`. O que causa essas exceções? (escreva como comentário em seu código). Implemente um programa em Java que faça o tratamento dessas exceções

- 2) Implemente a classe “ContaCorrente”:

Atributos (todos float):

- “limite” que armazena a quantidade atual disponível do limite da conta que o usuário possui;
- “saldo” que é o valor que realmente é pertencente ao usuário;
- “valorLimite” que consiste no valor máximo que o banco lhe fornece como valor de limite.

Métodos:

- `public void sacar(float valor)`
- `depositar(float valor)`
- `setValorLimite(float valor)`.

Na construção dos três métodos faça com que eles lancem exceções relacionados aos argumentos, por exemplo, sacar, depositar ou setar um valor negativo para esses eventos. Lance também uma exceção no caso de tentar sacar um valor maior que o possível.

Nos métodos lancem a exceção “`IllegalArgumentException()`” (Java) com comando “`throw`”, passe o motivo da exceção (uma `String`) em seu construtor.

Implemente a classe `TesteContaCorrente`, onde se deve instanciar um objeto da classe “`ContaCorrente`”, usar os três métodos construídos acima e tratar as exceções propostas com os comandos “`try`” e “`catch`”, no `catch` imprimir na tela o método “`getMessage()`” da exceção declarada no `catch`. Cause condições apropriadas para a execução das exceções.

- 3) Implemente a Classe `Cliente`:

- a. Identifique os atributos e métodos dessa classe (incluir getters/setters).  
Atributo obrigatório: CPF
- b. Os métodos `set` devem lançar exceções do tipo `DadoInvalidoException` quando o dado passado como parâmetro não for válido.

Implemente a classe `CadastroCliente`:

- c. Deve possuir um conjunto de `Cliente` (utilizar `array` de tamanho 10)
- d. Deve conter os métodos `inserir` (que adiciona um novo cliente ao conjunto de `Cliente`) e o `buscar` (que busca um cliente no conjunto)
- e. O método `inserir` deve lançar a `RepositorioException` quando não puder mais inserir clientes no array e a exceção `ClienteJaExistenteException` quando um cliente com o mesmo CPF já estiver cadastrado.
- f. O método `buscar` deve lançar a exceção `ClienteInexistenteException` quando o cliente que deseja buscar não estiver cadastrado.