# Conceitos da linguagem Java

**POO** - Programação Orientada a Objetos. Este tipo de programação trata os elementos da linguagem de maneira semelhante aos objetos reais.

**Packages** - É semelhante ao conceito de biblioteca de funções, sendo que um package (pacote) é um conjunto de classes, que ficam num diretório com o mesmo nome do pacote. O package default é o java.lang.\*; que é adicionado a todo arquivo java mesmo que o programador não o referencie. O .\* diz ao java para incluir todas as classes do pacote.

**Classes** - É um conjunto de objetos com características cmuns. Uma classe é como um modelo para a criação de objetos, que tem as mesmas características da classe à qual pertence.

1. **public** **class** Pessoa
2. {
3. **String** nome;
4. **Date** nascimento;
5. **double** dinheiroNaCarteira;
7. **public** Pessoa(String nome, Date nasc)
8. {
9. **this**.nome  = nome;
10. **this**.nascimento = nasc;
11. }
13. **public** **void** gastar(**double** valor)
14. {
15. dinheiroNaCarteira -= valor;
16. }
18. **public** **void** receber(**double** valor)
19. {
20. dinheiroNaCarteira += valor;
21. }
22. }

**Objetos** - É um elemento de uma classe. Temos uma classe "gatos" que é formada pelos objetos "gato". Todos os objetos tem em comum o fato de serem gatos (mesma espécie), mas podem ter características diferentes entre si. Objetos tem variáveis e métodos como também classes.

**Instância** - Uma instância de uma classe é um novo objeto criado dessa classe, com o operador new. Instanciar uma classe é criar um novo objeto do mesmo tipo dessa classe. Uma classe somente poderá ser utilizada após ser instanciada.

**Métodos** - Os métodos representam os estados e ações dos objetos e classes.

1. **public** **class** Pessoa
2. {
3. **String** nome;
4. **Date** nascimento;
5. **double** dinheiroNaCarteira;
7. **public** Pessoa(String nome, Date nasc)
8. {
9. **this**.nome  = nome;
10. **this**.nascimento = nasc;
11. }
13. **public** **void** gastar(**double** valor)
14. {
15. dinheiroNaCarteira -= valor;
16. }
18. **public** **void** receber(**double** valor)
19. {
20. dinheiroNaCarteira += valor;
21. }
22. }

**Variáveis** - As variáveis e constantes representam as características dos objetos.

**Packages** - Classes - Objetos - Métodos e variáveis.

**Interface** - É a declaração de um conjunto de constantes e métodos sem qq implementação, usando a palavra implements. É o conjunto de requisições que um objeto pode atender.

**Superclasse** - Todas as classes são criadas tendo outra como base. A classe que gerou a outra é chamada de superclasse. E fazemos referência à supercasse de uma classe usando a palavra extends. A classe gerada é chamada de subclasse. Toda classe tem uma subclasse. Quando não declaramos explicitamente a superclasse a super default é a Object.

**Construtores** - São usados para inicializar objetos. É o método que tem o mesmo nome da sua classe. Não pode ter um tipo de retorno e é chamado pelo operador new.

**Polimorfismo** - É a capacidade de um método executar a ação adequada dependendo do tipo de objeto.

**Threads** - São fluxos de execução paralelos, executando cada um tarefas diferentes. Em máquinas com dois ou mais processadores a execução pode ser simultânea e em máquinas com um único a execução será simulada, dividindo o tempo de processamento.

**Herança** - Quando a subclasse herda as características da superclasse. Denota especialização.

**Encapsulamento** - É o empacotamento (encapsulamento) de variáveis e métodos, ocultando a implementação do usuário. Representa reutilização, segurança e facilidade de manutenção.

**Coleta automática de Lixo** - Após processar o fecha-chaves do método main a linguagem libera a memória dos objetos até então em uso.

**Dicas sobre Classes**

- Sempre que possível use variáveis private   
- Nunca deixe de inicializar variáveis   
- Dividir classes grandes   
- Quando temos um programa composto por vários arquivos e o principal (aquele que contém o métod main) chama classes existentes nos demais, basta compilar o arquivo principal que os demais serão compilados.   
- A palavra-chave this representa o objeto atual.

Até logo!

Leia mais em: [Conceitos da linguagem Java](http://www.devmedia.com.br/conceitos-da-linguagem-java/5341#ixzz3sgz6m64o) <http://www.devmedia.com.br/conceitos-da-linguagem-java/5341#ixzz3sgz6m64o>