Java

Java Development Kit

JDK é um conjunto de ferramentas que permite o desenvolvimento de software pela linguagem **java**. O mesmo deve ser instalado e depois configurado na variável de ambiente para funcionar.

Comentarios

Existem 3 tipos de comentários

//

comentário de 1 linha

/\*

\*

\*

\*/

Comentário de varias linhas

/\*\*

\* @author

\* @since

\*/

Comentário de documentação

O comentário de documentação possue algumas definições que podem ser adicionada a ele como no exemplo acima **@author, @Since**

Variáveis e Constante

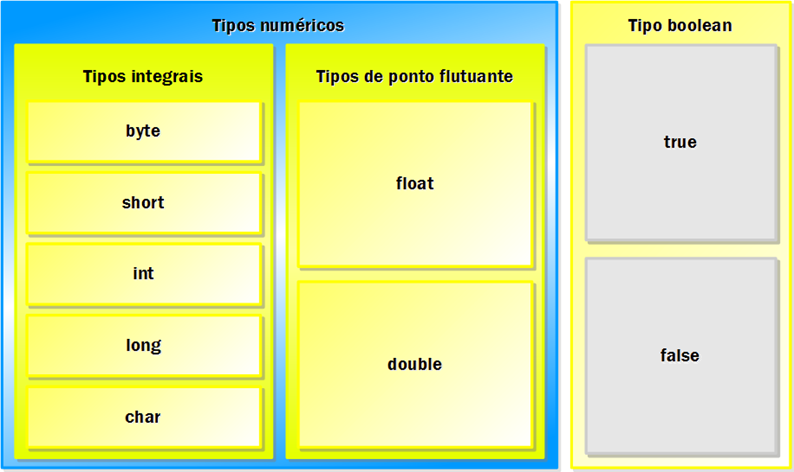
**Variáveis**: São propriedades definidas em um objeto que podem ter seu valor alterado durante o uso do sistema

**Constante:** Ao contrario da variável a constante não pode ser modificado trata-se de um valor fixo dentro do objeto

Tipos primitivos em Java

Em Java, nós temos **8 tipos primitivos**:

* byte;
* short;
* int;
* long;
* boolean;
* char;
* float; e
* double.



Import

Utilizamos o import para importar uma classe ou classes do mesmo ou de outro pacote para utilizarmos suas funções

String

É uma classe que permite trabalhar com textos, ela possui inúmeros métodos que permite deixar o texto com letra maiúscula, minúscula, concatenar entre outras disponíveis na classe

Objeto vs Tipo Primitivo

A diferença entre eles e que o tipo primitivo recebe apenas o valor determinado a ele como exemplo citarei o **int,** esse tipo primitivo recebe apenas números inteiros e não possui métodos ou atributos. Já o objeto pode possuir atributos, métodos e construtor

Wrappers

São Objetos desenvolvidos para trabalharem recebendo valores de tipo primitivo mas assim como a Classe **String** eles possuem métodos pre determinados que podem ser utilizados para agilizar o desenvolvimento do programador. Cada tipo primitivo possui se Wrapper.

Em Java, nós temos **8 tipos primitivos e seus wrappers**:

* byte (Byte);
* short (Short);
* int (Integer);
* long (Long);
* boolean (Boolean);
* char (Character);
* float; (Float)
* double (Double)

Como foi dito os Wrappers possuem inúmeros métodos entre eles estão métodos de conversão de tipo primitivos entre outros. Geralmente a ide abre os métodos disponíveis para uso então basta navegar entre eles e utilizar o que necessita

Java Persistence API (JPA)

Vamos aprender como a utilizar o framework **hibernate** e **jpa** que são utilizados para agilizar a persistência com o banco de dados. Nessa primeira parte vamos entender o padrão utilizado pelo **jpa** e como ele funciona

**DataMapping:** gerenciamento de dados, mapeamento de dados é o processo de criação de mapeamentos de elementos de dados entre dois modelos de dados distintos. O mapeamento de dados é usado como primeiro passo para uma ampla variedade de tarefas de integração de dados

**ORM (Object Relational Mapper): é uma técnica de mapeamento objeto relacional que permite fazer uma relação dos objetos com os dados que os mesmos representam**. Ultimamente tem sido muito utilizada e vem crescendo bastante nos últimos anos.

Nessa parte do aprendizado aprenderemos a usar o padrão **data Mapping** e relacionar objetos a tabelas utilizando o **orm**