Java I (Anderson Menezes)

[anderson.menezes@rj.senac.br](mailto:anderson.menezes@rj.senac.br)

## Orientação a Objetos e Características do Java 21/02/18

**• Paradigma** *// um exemplo que serve como modelo; padrão*

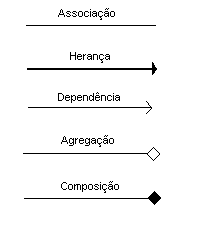
**• Objeto** *// representação de uma entidade do mundo real em uma solução computacional*

• **Classe** *// é onde modelamos os objetos; especificação que define um objeto*

Características **(atributos)** *// propriedades do objeto, também conhecidas como variáveis*

Comportamento **(métodos)** *// ações que os objetos podem exercer*

• **Relacionamentos** *// fornecem um caminho para comunicação entre os objetos*

- Tipos de relacionamento:

• **Mensagens** *// representam a comunicação entre os objetos. Ex: Chamadas de um método de um objeto por outro objeto*

• **Polimorfismo** *// capacidade do objeto de assumir diferentes formas*

• **Encapsulamento** *// técnica utilizada para esconder uma ideia, ou seja, não expor detalhes internos para o usuário*

• **Reuso** *// reutilização de um software ou código, sem necessidade de uma “nova criação”*

• **Abstração** *// é utilizada para a definição de entidades do mundo real (objetos e/ou classes)*

*Características do Java:*

**Multiplataforma** *// pode ser executado em mais de uma plataforma*

**Linguagem Interpretada** *// o código fonte é executado por um interpretador, que em seguida é executado pelo sistema operacional ou processador*

**Fortemente tipada** *// toda e qualquer variável em Java precisa ser definida*

## Tipos de Dados e Operadores 28/02/18

86 bits 166 bits 326 bits 64 bits 326 bits 64 bits 166 bits 1 bit

**Tipos primitivos:** **byte, short, int, long**, **float, double**, char, boolean)

inteiros reais

(ponto flutuante)

**Tipos complexos:**

- Criados pelas classes que escrevemos

- Não se limita ao armazenamento de apenas um valor por vez

- A existência de métodos

• Sempre especificamos o tipo de uma variável (declaração):

*tipo nomeVariavel;*

Exemplos:

**int a;**

**float b;**

**char c;**

**a = 15;** // atribuição de valores

**• Operadores**

**- Aritméticos:** + , - , \* , / , %

**- Relacionais:** < , > , <= , >= , != , ==

**- Negação:** !

**- Lógicos:** && , ||

## Projeto (Modelo) 28/02/18

**PrimeiroProjeto**

src (source)

**telas**

**TelaPrincipal.java**

**CadastroCliente.java**

bancoDados

• Convenção de Nomenclaturas:

* **Nome de projeto** = indiferente quanto **maiúsculo ou minúsculo**
* **Nome de pacote** = sempre inicia com letra **minúscula** *// demais palavras, letra maiúscula*
* **Nome de classe** = sempre inicia com letra **maiúscula**
* Variáveis e métodos = sempre iniciam com letra **minúscula** *// demais palavras, letra maiúscula*

## Laços e Condicionais 07/03/18

• Controles de Fluxo

- Condicionais: if, swich

- Laços de repetição: for, while, do

EM PROGRAMAÇÃO:

if (condição) {

lógica1;

} else {

lógica2;

}

while (condição) {

lógica;

}

do {

lógica;

} while (condição);

## Objetos e Classes 21/03/18

**CLASSE** = ”Descrição do objeto”

**OBJETO** = Instância criada a partir da classe

Em Java:

1. Pessoa p; // Referenciando um objeto
2. p = **new** Pessoa(); // Construindo um novo objeto (instanciando)
3. Pessoa p = **new** Pessoa(); // referenciação e instanciação na mesma linha

## Métodos de entrada e saída 21/03/18

1. System.***out***.println();

* println() = método
* out = atributo
* System = classe

1. Scanner entrada; // 1. Declarar uma referência p/ o objeto
2. entrada = **new** Scanner (System.***in***);

// 2. Instanciar um objeto chamando o construtor da classe “Scanner ()”

// 3. É necessário que a classe Scanner pertença ao mesmo pacote que está sendo utilizado a referência e o construtor da Classe Scanner.

**import** java.util.Scanner;

1. String nome = entrada.nextLine (); // 3. Atribuindo o valor da entrada à variável “nome”

## Interface Gráfica 09/05/18

* AWT (Abstract Window Toolkit)
* Swing
  + Look and Feel
  + Pug-ins
* Padrões
  + Design (Projeto) – Design Patterns
  + Arquiteturais
* Composite

## Biblioteca

• Usuário

JRE (Java Runtine Environment)

• Desenvolvedor

JDK (Java Development Kit)

IDE (Integrated Development Environment) *// Eclipse, NetBeans, etc.*