

PROGRAMAÇÃO ORIENTADA A OBJETOS

Prof. Milton Palmeira Santana



Introdução ao JAVA

- » JAVA é mais que uma linguagem de programação, é uma plataforma.
- » O JAVA foi criado no início dos anos 90 pela Sun Microsystems com o objetivo de ser uma linguagem independente de plataforma de execução.

The Oracle logo consists of the word "ORACLE" in a white, sans-serif font, set against a solid red rectangular background.

ORACLE®

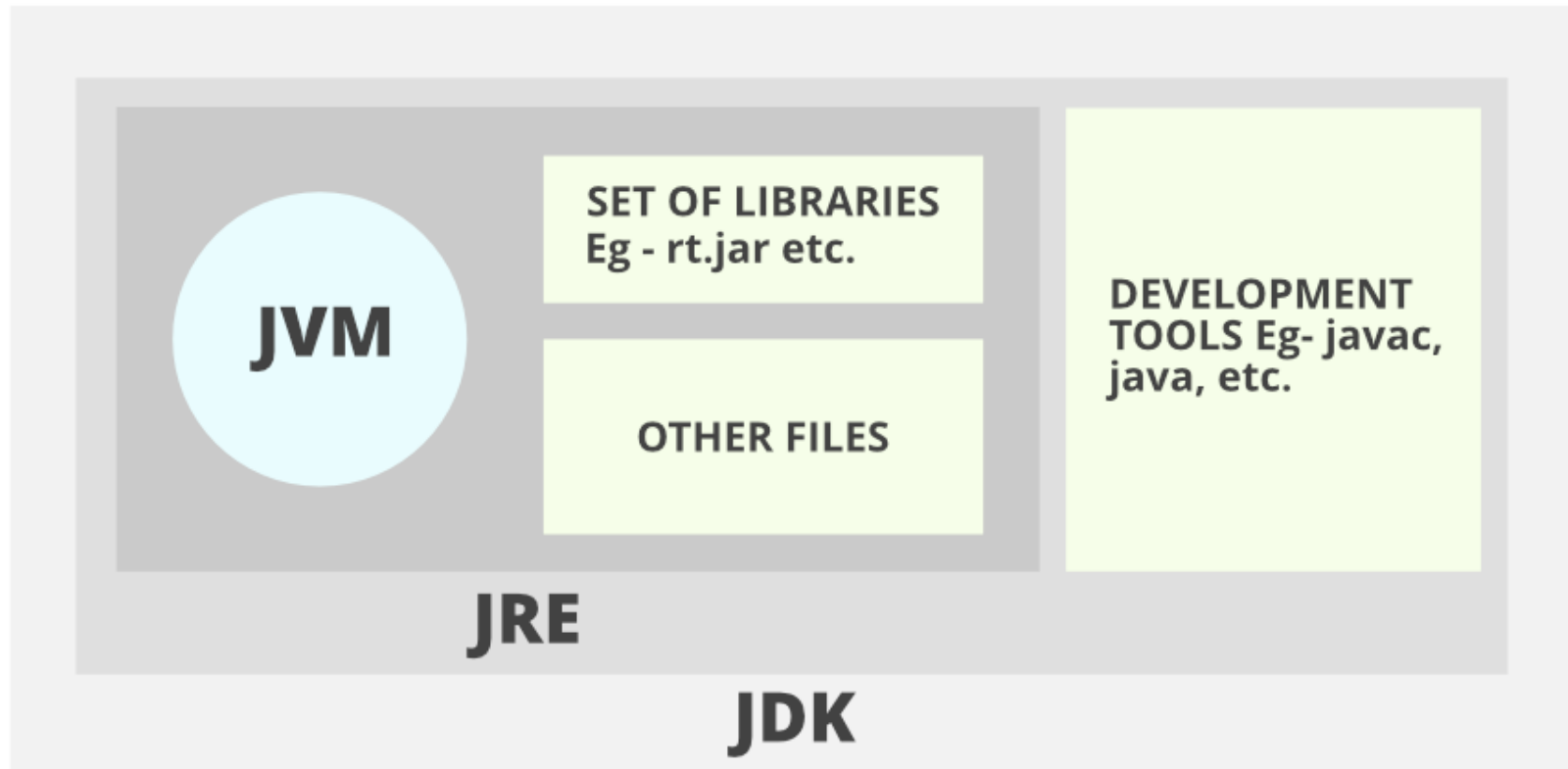
Introdução ao JAVA

- » ByteCode
- » JVM – Java Virtual Machine



Introdução ao JAVA

- » JRE – Java Runtime Environment
- » JDK – Java Development Kit



Introdução ao JAVA

- » JSE – Standard Edition
- » JEE – Enterprise Edition
- » JME – Micro Edition

Introdução ao JAVA

» IDE para JAVA



Apache

NetBeans IDE



eclipse

Introdução ao JAVA

- » Netbeans foi desenvolvido por um grupo de estudantes (chamado Xelfi em alusão ao Delphi) em 1996.
- » Em 1999 o projeto havia evoluído para uma IDE proprietária.
- » Nessa época, a Sun Microsystems havia desistido de seu projeto de IDE para Java e procurado por novas iniciativas. A escolha foi o NetBeans.
- » Por alguns meses a Sun modificou o nome do projeto e o manteve proprietário, até que em junho de 2000 a Sun disponibilizou o código fonte do NetBeans tornando-o uma plataforma OpenSource;

Introdução ao JAVA

- » Criando o primeiro programa
- » Crie um novo projeto no netbeans e chame de “Hello World”.
- » Adicione outra mensagem.
- » Faça com que apareça 2 frases, uma abaixo da outra utilizando apenas 1 `System.out.println`.

Introdução ao JAVA

- » O que pode dar errado?
 - Falta de ;
 - Erro de digitação
 - Falta do código main
 - “esquecer” algum código como o public por exemplo
 - Não adicionar uma classe para iniciar o programa

Introdução ao paradigma de orientação a objetos

- » Paradigma significa modelo, padrão.
- » Programa de computador
- » O paradigma de programação orientada a objetos considera que os dados a serem processados e os mecanismos de processamento desses dados devem ser considerados em conjunto.
- » Quando representamos elementos reais de forma simplificada e padronizada, estamos criando um modelo para esses elementos. Modelos são representações simplificadas de objetos, pessoas, itens, tarefas, processos, conceitos, ideias etc.

Introdução ao paradigma de orientação a objetos

- » Seria adequado, então, associarmos modelos a elementos que vemos no mundo real?

Histórico do paradigma de orientação a objetos

- » O criador da expressão programação orientada a objetos (POO) foi Alan Kay, o mesmo que criou a linguagem Smalltalk.
- » No entanto, mesmo antes de o termo ter sido criado, ideias conceituais sobre orientação a objetos já estavam sendo aplicadas na linguagem SIMULA 67.
- » Estudando à época, vislumbrou um computador pessoal que pudesse fornecer aplicações orientadas a gráficos e intuiu que uma linguagem como a SIMULA poderia oferecer bons recursos para leigos criarem tais aplicações.

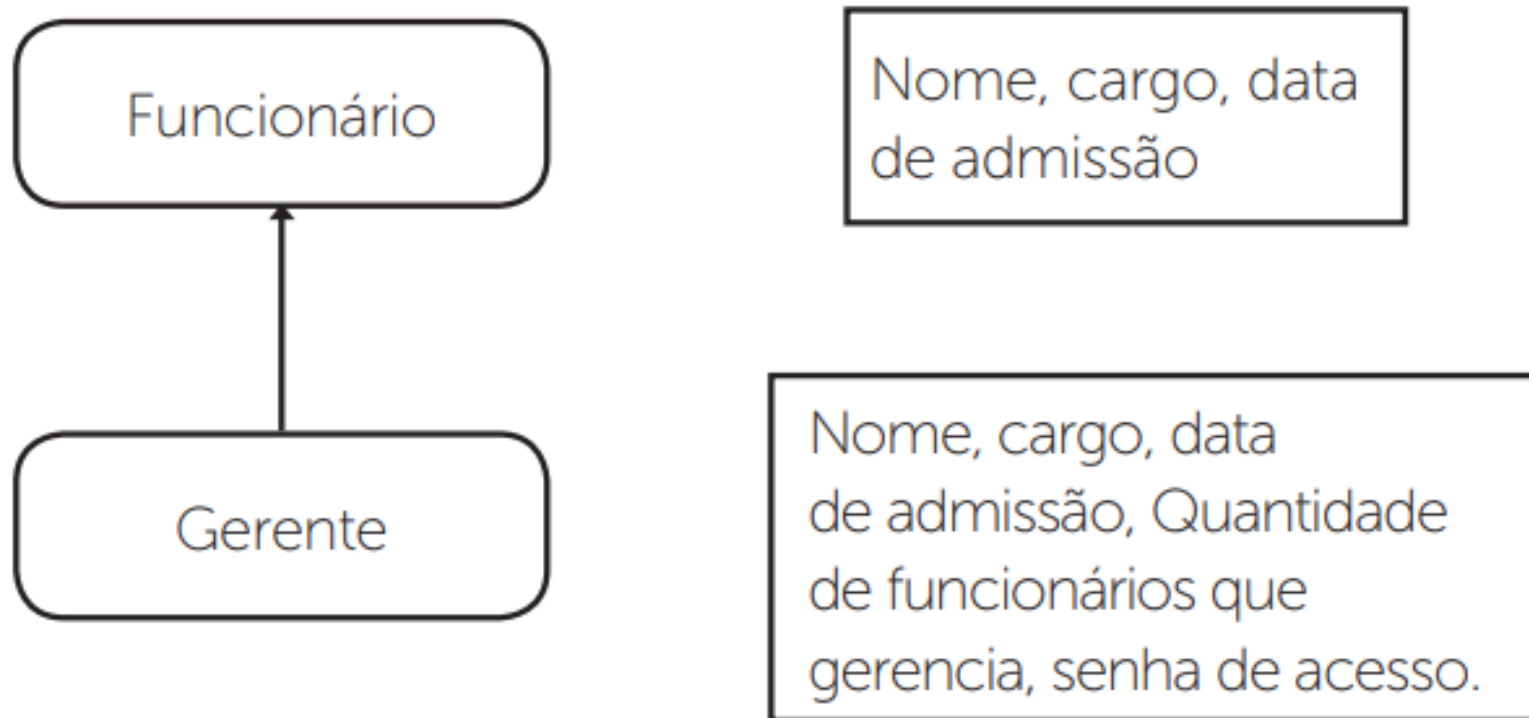
Histórico do paradigma de orientação a objetos

- » Kay então resolveu “vender sua visão” à Xerox e no início dos anos 1970; sua equipe criou o primeiro computador pessoal, o Dynabook. A linguagem Smalltalk, que era orientada a objetos e também orientada a gráficos, foi desenvolvida para programar o Dynabook.
- » A ideia da programação orientada a objetos ganhou impulso na década de 1970 e, no começo da década de 1980, Bjarne Stroustrup integrou a orientação a objeto na linguagem C, o que resultou no C++, tida como a primeira linguagem OO usada em massa.
- » No início dos anos 1990, um grupo da Sun, liderado por James Gosling, desenvolveu uma versão mais simples do C++, que foi batizada de Java.

Características e princípios do paradigma de orientação a objetos

- » Machado (2015) entende que o paradigma da orientação a objeto é fundamentado por quatro características:
- » **Abstração:** a abstração está relacionada à definição precisa de um objeto. Essa definição inclui sua identificação (nome), suas características (ou propriedades) e o conjunto de ações que ele desempenha. Ex: Cachorro.
- » **Herança:** por meio dessa característica do paradigma OO, um objeto filho herdará características e comportamentos do objeto pai. Quando estiver criando classes, você vai perceber que essa possibilidade permite o reaproveitamento de código e torna o trabalho mais racional e otimizado.

Características e princípios do paradigma de orientação a objetos



Características e princípios do paradigma de orientação a objetos

- » **Encapsulamento:** nesse nosso contexto, o termo encapsulamento está relacionado à proteção ou ocultação dos dados do objeto.

Variáveis

- » São espaços reservados (alocados) na memória RAM do computador, que devem possuir um nome e um tipo de dado, sendo que seu conteúdo pode sofrer alteração durante a execução do programa.
- » Exemplos: o Nome de um cliente, as Notas de um aluno etc.

Variáveis

- » O nome de uma variável:
 - não pode ser iniciado por número.
 - não pode conter espaços ou qualquer caracter especial, exceto o underline “_”.
 - não pode ser igual ao de qualquer palavra reservada da linguagem de programação.
 - Procure utilizar nomes curtos, e que possuam algum significado associado ao identificador.
 - **Atenção:** Não é regra, mas procure sempre declarar uma variável com sua primeira letra minúscula. Será abordado mais a frente o assunto

Tipos de dados

- » Dados, são as informações em sua forma mais bruta. Sendo que estas informações podem ser classificadas em alguns grupos distintos:
 - **Numéricos:** Conjunto formado pelos números (inteiros, reais, etc).
 - **Alfanuméricos:** Conjunto formado por letras e números (palavras).
 - **Lógicos:** Representam apenas dois valores/estados. Verdadeiro ou Falso.

Tipos de dados

byte

- Inteiro de 8 bits. Pode assumir valores de -128 a 127

int

- Inteiro de 32 bits

long

- Inteiro de 64 bits

short

- Inteiro de 16 bits

double

- Ponto flutuante de 64 bits

float

- Ponto flutuante de 32 bits.

boolean

- Tipo de dados booleano. Pode ser apenas true ou false.

char

- Caractere unicode de 16 bits.

string

- Texto em Unicode

Classe Scanner

- » A classe Scanner no Java tem a finalidade de facilitar a entrada de dados no modo Console;
- » Com ela podem ser convertidos textos para tipos primitivos;
- » Para utilizar a classe Scanner precisamos fazer primeiro a seguinte importação:
 - `import java.util.Scanner;`

Classe Scanner

- » Para realizar a leitura de um texto deve-se criar um objeto do tipo Scanner que passa como argumento o objeto **System.in**;
- » Nota: Os conceitos apresentados serão vistos com mais detalhes na aula de Orientação a Objetos.
 - `Scanner scan = new Scanner(System.in);`

Classe Scanner

- » Agora para realizar a leitura dos dados utilizamos o objeto scan criado e utilizar alguns métodos como abaixo:
- » Para ler uma string
 - `scan.nextLine();`
- » Para ler um valor inteiro
 - `scan.nextInt();`
- » Para ler um valor do tipo double
 - `scan.nextDouble();`

Classe Scanner

- » Realizar a leitura de uma string, um número inteiro e um número double.

Exercício

- » Elabore um programa que solicite o nome do usuário. Mostre em seguida a seguinte mensagem:
 - “Seu nome é: nomeDigitado”
 - Após serem efetuados testes solicite também a idade do usuário e mostre na tela.
- » Faça um programa que receba 2 números e faça a soma entre eles.



Anhanguera