



CURSO

TECNICO EM INFORMATICA

DISCIPLINA

**LINGUAGEM DE
PROGRAMAÇÃO_II**

PROFESSOR: EDILSON MELO

Introdução ao Java

- ✓ A linguagem de programação Java foi criada em 1991 por James Gosling, ela iniciou-se como parte do projeto Green da Sun Microsystems. Inicialmente a linguagem iria chamar-se Oak (Carvalho) em referência a árvore que era visível pela janela de James Gosling.
- ✓ A mudança de nome ocorreu pois já existia uma linguagem de programação com este nome, então a linguagem foi rebatizada para Java;

Introdução ao Java

- ✓ Linguagem de programação JAVA possui um ambiente de desenvolvimento composto:
- ✓ Pelo compilador, interpretador, gerador de documentação etc.;
- ✓ Possui um ambiente de execução que pode ser praticamente qualquer máquina que possua Java Runtime Environment (JRE) instalado.

Introdução ao Java

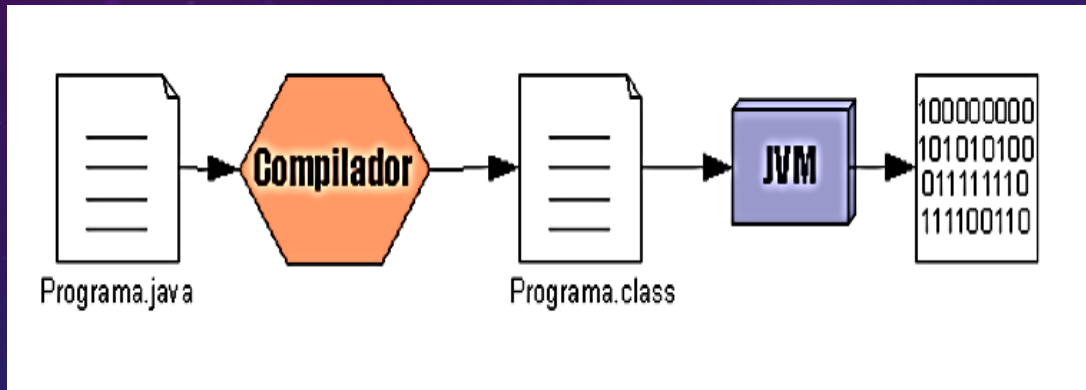
➤ Máquina Virtual Java:

- ✓ JVM máquina virtual java (JVM) é uma máquina imaginária que emula uma aplicação em uma máquina real;
- ✓ É a JVM que permite a portabilidade do código Java, isto ocorre porque todo código Java é compilada para um formato intermediário, bytecode, este formato é então interpretado pela JVM;
- ✓ Existem diversas JVMs cada uma delas destinada a um tipo de sistema operacional (Windows, Linux, Mac e etc.), desta forma sendo o código da aplicação Java, bytecode, um código interpretado pela JVM, podemos desenvolver uma aplicação sem nos preocuparmos onde ela será executada pois sabemos que existindo a JVM instalada nosso código será executável.

Introdução ao Java

Fases de um programa Java

As fases pelo qual passam um programa Java relacionam-se da seguinte forma:



1. Criação do código fonte (Programa.java);
2. Compilação do código fonte e geração do bytecode (Programa.class);
3. Interpretação do bytecode pela máquina virtual;
4. Conversão do bytecode em linguagem de máquina.

Introdução ao Java

➤ Hotspot

- ✓ Hotspot é a máquina virtual Java, ela provê algumas funcionalidades muito importantes;
- ✓ A JVM vem se aprimorando e, em muitas situações, as aplicações Java tem desempenho similar as aplicações que são previamente compiladas;
- ✓ Este desempenho vem melhorando muito devido a otimização que a máquina virtual consegue fazer a medida que o código é executado.

Introdução ao Java

➤ JRE e JDK

- ✓ JRE: O Java Runtime Environment contém tudo aquilo que um usuário com um precisa para executar uma aplicação Java (JVM e bibliotecas), como o próprio nome diz é o “Ambiente de execução Java”;
- ✓ JDK: O Java Development Kit é composto pelo JRE e um conjunto de ferramentas úteis ao desenvolvedor Java.

Introdução ao Java

➤ Java - Definições

- ✓ **Classe:** É a estrutura que, quando construída, produzirá um objeto, dizemos “todo objeto é instância de alguma classe”;
- ✓ **Objeto:** Em tempo de execução, quando a JVM encontra a palavra reservada new é criada uma instância da classe apropriada;
- ✓ **Estado:** É definido pelo conjunto de atributos de uma classe, isto é, cada instância da classe possuirá um estado independente dos demais objetos;
- ✓ **Comportamento:** São os métodos da classe, comportamento é aquilo que uma classe faz (algoritmos), muitas vezes, um determinado comportamento(método) muda o estado do objeto, isto é, após a execução do método um ou mais atributos mudaram de valor;

Introdução ao Java

➤ Java - Nomenclatura

- ✓ Existem três aspectos importantes, em relação a nomenclatura, que devemos considerar quando estamos programando Java.
- ✓ **Identificadores válidos:** Definem as regras para que o compilador identifique o nome como válido.
 - I. Devem iniciar com uma letra, cifrão (\$) ou sublinhado/underscore (_);
 - II. Após o primeiro caracter podem ter qualquer combinação de letras, caracteres e números;
 - III. Não possuem limite de tamanho;
 - IV. Não podem ser palavras reservadas;
 - V. Identificadores são case sensitive isto é, “Nome” e “nome” são identificadores diferentes.

Exemplos:

Identificadores válidos	Identificadores inválidos
_codigo	5ident
\$turma	-idade
\$\$_5A	%valor

Introdução ao Java

✓ **Convenção de nomenclatura da SUN:** São recomendações da SUN para nomenclatura de classes, métodos e variáveis.

- I. Classes e interfaces: A primeira letra deve ser maiúscula e, caso o nome seja formado por mais de uma palavra, as demais palavras devem ter sua primeira letra maiúscula também (camel Case);
- II. Métodos: A primeira letra deve ser minúscula e após devemos aplicar o camel Case;
- III. Variáveis: Da mesma forma que métodos;
- IV. Constantes: Todas as letras do nome devem ser maiúsculas e caso seja formada por mais de uma palavra separada por underscore.

Exemplos:

Classes	Métodos	Variáveis	Constantes
Carro	desligar	motor	COMBUSTIVEL
CursoJavaIniciante	iniciarModulo	quantidadeModulos	NOME_CURSO
Hotel	reservarSuiteMaster	nomeReservaSuite	TAXA_SERVICO

Introdução ao Java

- ✓ **Convenção JavaBeans:** Requisitos para que os nomes atendam a especificação para JavaBeans.
 - I. Se o atributo não for um booleano o método getter (utilizado para obter a variável) deve iniciar por “get”;
 - II. Se o atributo for um booleano o método getter pode iniciar por “get” ou “is”;
 - III. O método setter (atribuição) da propriedade sempre deve iniciar por “set”;
 - IV. O restante do nome deve ser o nome do atributo concatenado ao prefixo(is, get ou set) em letra maiúscula;
 - V. O método setter e getter sempre devem ser públicos.

Exemplos:

Setters – atributo nome	Getters – atributo valor
public void setNome(argumento)	public boolean getValor()
	public boolean isValor()