



Programação Orientada a Objeto com a linguagem



Prof. Edson Melo de Souza, Me.

prof.edson.melo@gmail.com

www.edsonmelo.com.br





Conteúdo Programático



- Introdução
- Conceitos Gerais
- Tipos de Dados, Valores Literais e Variáveis
- Operadores
- Comandos
- Orientação a Objetos
- Classes
- Métodos
- Construtores
- Acesso a Banco de Dados
- Modelo MVC





Conceitos Gerais

- Criada em 1991 por **James Gosling**
- Utilização em Aparelhos Eletrônicos
- Java é tanto uma **linguagem** quanto uma **plataforma**
- É Gratuita





Linguagens de Programação



- Linguagens de Máquina
 - Comandam ações do computador através de instruções sequenciadas por **0 e 1**
 - Cada tipo de CPU possui uma em particular
- Linguagem Assembly
 - Semelhante as linguagens de máquina, porém utilizam nomes simbólicos . Ex.: N EQU 4 (*atribui o valor 4 a N*)
- Linguagem de Alto Nível
 - Estão mais próximas das linguagens humanas, são mais fáceis de aprender e de se utilizar que as outras
 - Ex.: C, Pascal, C#, Java, etc.





Compilação

É o processo que traduz o **código fonte** (*escrito em uma linguagem de alto nível*) para o **código executável** (*linguagem de máquina*)





Compilador

Programa que tem a função de compilar um código fonte

Tarefas

1. Verificar as regras sintáticas da linguagem, informar erros
2. Gerar o código executável se tudo estiver correto





Erros de um Programa

- **Erros de Compilação:** São erros que ocorrem durante a compilação de um programa. Se o compilador não encontrar nenhum erro, significa apenas que ele está sintaticamente correto
- **Erros de Execução:** Erros que ocorrem durante a execução do programa e que não são detectados na hora da compilação





ORACLE





Java: Conceitos Gerais

- Orientada a **Objeto**, simples e familiar
- Robusta e segura
- Arquitetura neutra e **portável**
- Alto desempenho
- **Interpretada**, multiprocessada e dinâmica



Java: Conceitos Gerais

- Linguagem tanto compilada quando interpretada
 - Compilado para *Java **bytecode***
 - Interpretado pela plataforma da *Java Virtual Machine (JVM)*



A máquina Virtual Java

- A **JVM** (Java Virtual Machine) é responsável por interpretar o Java Bytecode e traduzi-lo em operações que deverão ser executadas pelo sistema operacional
- Diferentes sistemas podem executar a mesma operação de modos diferentes





O Java Runtime Environment



- É o ambiente de execução dos programas em Java
- Para rodar programas em Java é necessário ter o JRE instalado no computador



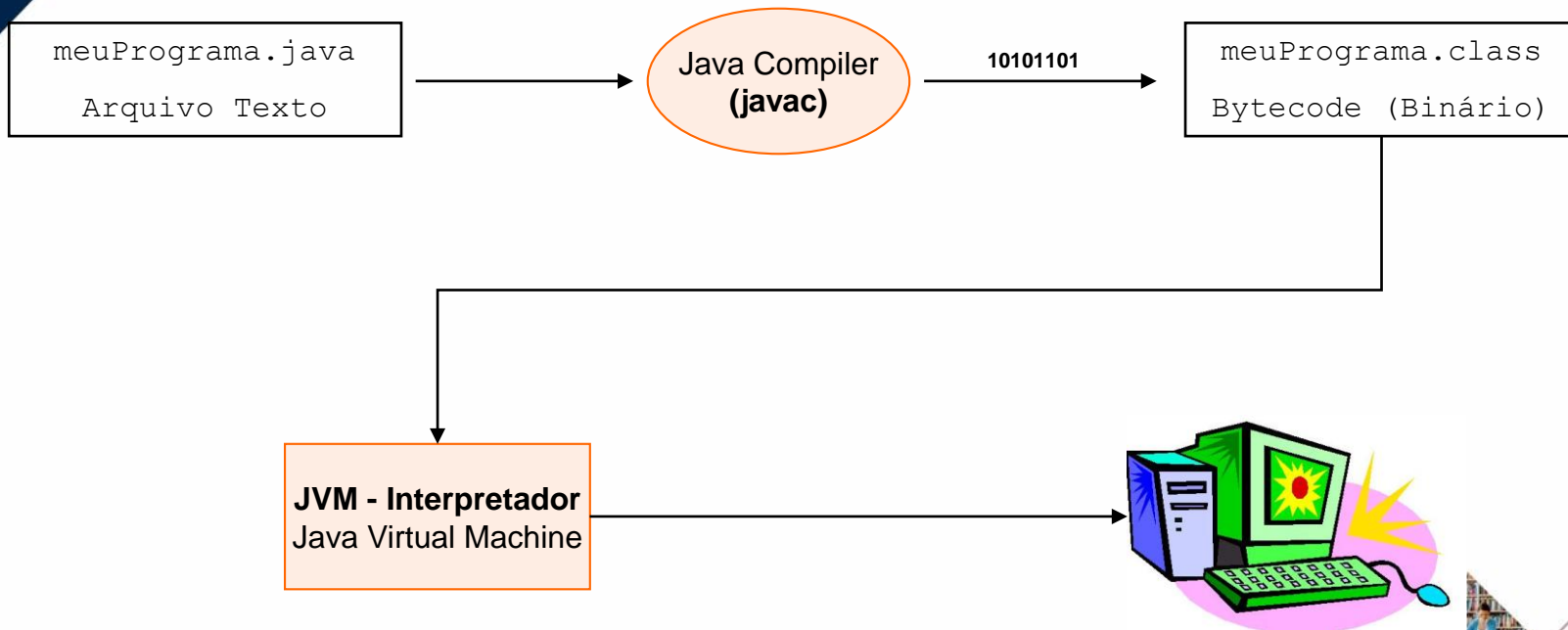


A Java Development Kit

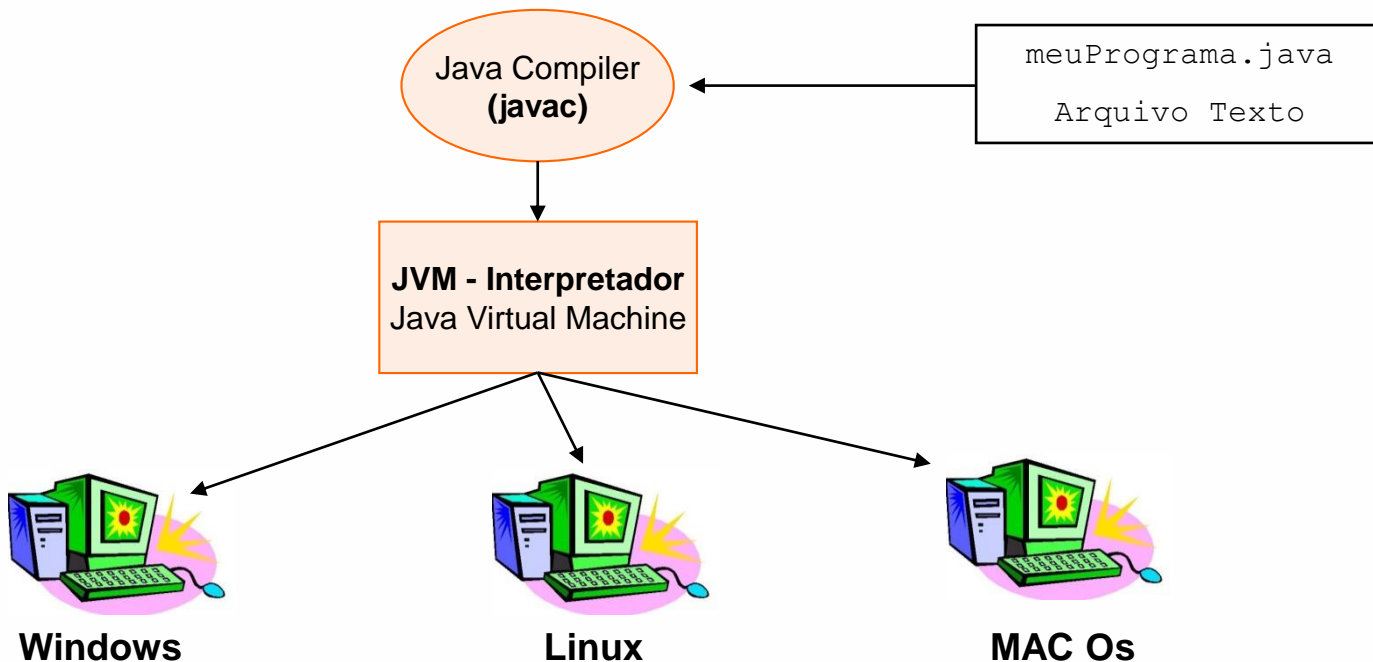
- É a plataforma de desenvolvimento Java SE (Standard Edition), também conhecida como J2SE
 - Inclui o compilador Java chamado de **javac**
 - O intérprete do **bytecode**
 - Classes já programadas para facilitar o desenvolvimento
 - É gratuito



Funcionamento



Portabilidade





Programando em Java

Ambiente Integrado de Desenvolvimento
Integrated Development Environment – IDE

Programas que auxiliam no desenvolvimento de programas, fornecendo ferramentas para acelerar o processo de construção.





IDE's

- Um IDE possui diversas ferramentas:
 - Editor de texto para escrever os programas
 - Funcionalidades de compilação
 - Janela para visualizar resultados
- Exemplos de IDE para Java
 - Eclipse (*mais utilizada no mundo*)
 - Netbeans





Estrutura de um Programa Java

**Um programa Java é um conjunto de
uma ou mais classes**

Requisitos para o funcionamento de um programa

1. Incluir uma classe com o mesmo nome do arquivo (*.java) onde o programa é guardado
2. Esta classe **pública** deve incluir um subprograma *main()* – é onde se iniciará a execução do programa





Exemplo de um Programa Java



```
public class OlaMundo
{
    public static void main(String args[])
    {
        system.out.println("Olá Mundo!");
    }
}
```





Exemplo de um Programa Java



Os programas incluem normalmente três tipos de componentes:

- **Declarações:** permitem reservar memória para o armazenamento das estruturas de dados envolvidas
- **Instruções:** indicam ao computador o que deve fazer
 - São separadas por vírgula
- **Comentários:** são úteis para os humanos, mas ignorados para o computador
 - `/* */` comentar mais de uma linha
 - `/** */` usado para documentação
 - `//` comentar até o final da linha





Sintaxe do Java

A sintaxe do Java, assim como em outras linguagens, inclui:

- Palavras reservadas
- Identificadores
- Símbolos
- Literais



Palavras Reservadas

abstract
boolean
break
byte
case
catch
char
class
const
continue

default
do
double
else
extends
final
finally
float
for
goto

if
implements
import
instanceof
int
interface
long
native
new
package

private
protect
public
return
short
static
strictfp
super
switch
synchronized

this
throw
throws
transient
try
void
volatile
while



Identificadores

Um identificador é definido pelo programador para nomear entidades que o programa manipula:
Variáveis, Constantes, Subprogramas e Classes

Identificador := { **letras**, **números**, **_**, **\$**}

- Não devem começar com números nem com “_”
- Não existe tamanho máximo (comprimento)
- Distinção entre **maiúsculas** e **minúsculas**





Convenções para Identificadores



Para uma melhor clareza na escrita de um programa é aconselhado seguir as seguintes convenções:

- Sempre que possível usar letras minúsculas
 - Ex.: `main()`
- Exceções: o `nome de uma classe` começa por maiúscula
 - Ex.: `Math`, `Poligono`, `System`
- Se o nome possui várias palavras, usar maiúscula para a primeira letra que segue a outra
 - Ex.: `imprimirMensagem`, `lerNome`
- Nome de constante sempre em maiúscula
 - Ex.: `MAX`, `TAXA_DE_JUROS`



Símbolos

- Operadores
 - Ex.: $+$, $-$, $*$, $=$, etc
 - O operador $=$ (igual) tem a função de atribuição e o operador $==$ (igual igual) tem a função de comparação entre valores, retornando um valor booleano verdadeiro se forem idênticos
- Sinais de pontuação
 - Ex.: $\{$, $\}$, $($, $)$, $;$, etc.



Literais

São dados explícitos que podem ser manipulados pelo programa:

- Números inteiros: -17 e 255
- Reais: 1.4142135, 1.562E-19 e 4.23f
- Caracteres: 'a', 'A', '\n'
- Cadeia de caracteres: "Olá Mundo", "Boa noite"





Variáveis

Uma variável representa uma localização em memória na qual podemos guardar um valor e recuperá-lo posteriormente.





Variáveis: Sintaxe e Declaração

`tipo_de_dado nome_da_variavel;`

`tipo_de_dado nome_da_variavel = valor inicial;`

`tipo_de_dado nome_da_variavel_1, nome_da_variavel_2;`

O **tipo** permite determinar o espaço em memória que será utilizado para armazenar aquele dado

O **nome** permite manipular a variável sem que seja necessário saber a localização na memória





Tipos de Dados Primitivos

Tipo	Memória Ocupada	Valores
Números Inteiros		
byte	8 bits	de -128 até 127
short	16 bits	de -32768 até 32767
int	32 bits	de -2147483648 até 2147483647
long	64 bits	de -9223372036854775808 até 9223372036854775807
Números Reais		
float	32 bits	de -3.4×10^{38} até 3.4×10^{38}
double	64 bits	de -1.7×10^{308} até 1.7×10^{308}
Representação de Caracteres		
char	16 bits	entre 0 e 65535
Valores Lógicos		
boolean	1 bit	true ou false





Exercícios

Quando a compilação de um programa Java é realizada, qual o **nome** e o **tipo** do arquivo gerado?

R: **Bytecode** e o seu tipo é **Binário**



Exercícios

Em que local um arquivo compilado em Java é executado?

R: Na **JVM** (Java Virtual Machine)

Exercícios

Qual a função do método **main** em um programa Java?

R: Realizar a execução do programa.

Bibliografia

**Programação Orientada a Objetos com
Java™ *sem mistérios*.** Evandro Carlos
Teruel, 2016. *Disponível na Central do Aluno em PDF*

GUI - Grupo de Usuários Java
www.guj.com.br