Java Básico

Prof. Me. Fabiano Fernandes

Conteúdo da aula

- Vamos relembrar?
  - Características de Java
  - Compilador vs. Interpretador.
  - Hello World
  - Tipos Primitivos
  - Operadores
  - Casts
  - Controle de fluxo
- Ok?

A história de Java

- Criada em 1995, pela Sun Microsystems
  - Green Project
- Incorporou vários conceitos de C/C++
- Idealizada por James Gosling
- Atrações principais:
  - Portabilidade
  - Fácil integração com a Web
- Primeiro grande passo
  - Netscape Navigator em Java!

Como estamos hoje?

- Programação Web
  - Applets
  - Server Side Programming – JSP
- Engenharia de Software
- Banco de Dados
- Multimídia
- Em geral, aplicações que precisem de alto grau de portabilidade.

Características

- Interpretada.
- Portável.
- Robusta.
- Extensível.
- Segura.
- Multi-tarefa.
- Baixa performance.
- Orientada a objetos.
- Case-sensitive.

Conceitos errados sobre Java

- Java = JavaScript.
- Java é totalmente portável.
- “Meu freio ABS roda em Java!!!”
- “Java é a melhor linguagem que existe!”
- “Java é trivial!”

Conceitos sobre Java

- Compilar:

– javac Nome_Classe.java

– javac –cp c:\diretorio_Classe Nome_Classe.java

- Executar:

– java Nome_Classe

Conceitos sobre Java

- JDK?
- JRE?
- JVM?

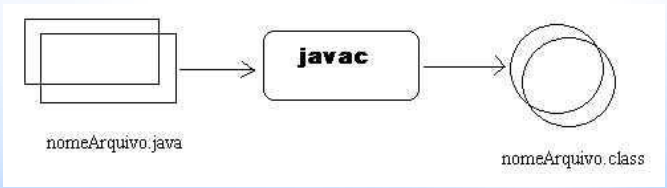
Primeiro Programa

- Hello World!!!
  - Abra o arquivo *HelloWorld.java* no editor de texto indicado.
  - Abra o *Java Runtime Environment* (JRE)

Link:
http://java.sun.com/docs/books/tutorial/getStarted/application/index.html

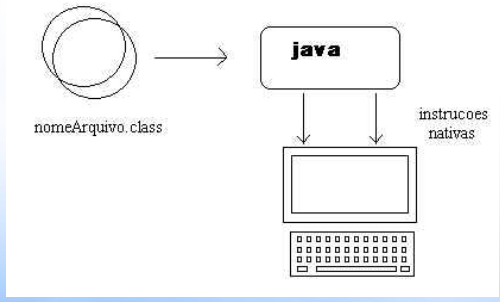
Compilando...

- Comando: *javac*
  - Sintaxe: *javac [NomeDaClasse].java*
- Exemplo: *javac HelloWorld.java*

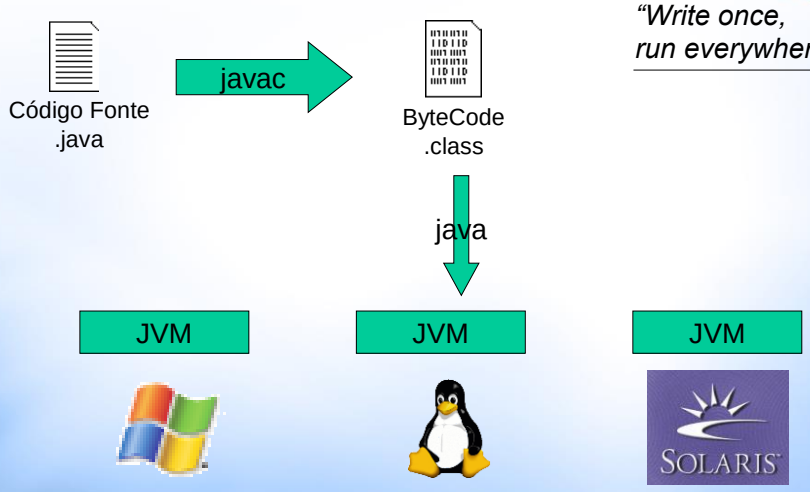


Executando...

- Comando: *java*
  - Sintaxe: *java [NomeDaClasse]*
- Exemplo: *java HelloWorld*

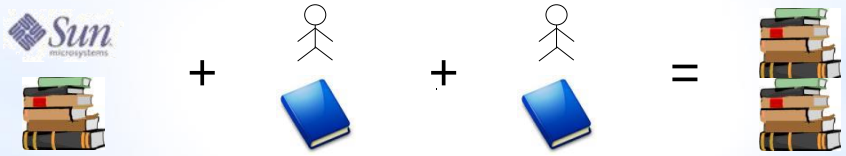


Portabilidade



"Write once, run everywhere!"

Extensibilidade



Conceitos Iniciais

- Java é Case sensitive
- Tudo em Java deve estar dentro de uma Classe
- Regras para o nome de uma Classe

  Exemplos: – PrimeiroExemplo; ClasseExemplo; CarroDeMao.
- Não pode utilizar uma palavra reservada do Java

Conceitos Iniciais

- Palavras Reservadas do Java:

byte - short - int - long - char - boolean - double - float - public - private - protected - static - abstract - final - strictfp - transient - synchronized - native - void - class - interface - implements - extends - if - else - do - default - switch - case - break - continue - assert - const - goto - throws - throw - new - catch - try - finally - return - this - package - import - instaceof - while - for - volatile - super

Comentários

- Aumentam a clareza do código.
- Facilitam a manutenção do programa.
- Aumentam o valor agregado do software.

Comentários

- Blocos de Comandos:
  - Delimitam um conjunto de comandos
  - Utiliza { e } – Exemplo:

```
public class Aula {
        public static void main(String[] args) {
                System.out.println("Codigo Exemplo");
        }
}
```

Conceitos Iniciais

- Java é fortemente tipada
- Tipos:
  - Primitivos
  - Referências para Objetos

Exemplos

- *HelloWorldComentado.java*

Tipos Primitivos

- Definidas pela linguagem
- Não precisam de construtor
- Podem ser:
  - Inteiro
  - Numéricos de Ponto Flutuante
  - Outros (char e boolean)

Tipos Primitivos

Tipo	Tamanho
boolean	{true, false}
byte	8-bit
short	16-bit
int	32-bit
long	64-bit
char	16-bit (unsigned)
float	32-bit
double	64-bit

Variáveis

- Declaração vs. Inicialização

Variáveis

Declaração
int a; boolean b; float x, y, z;

Inicialização
a = 10; *//supõe que a já estava declarada.* boolean b = false; *//declara e inicializa b*

Variáveis

- Toda variável deve ter um valor antes mesmo de seu valor ser usado.
- Possível erro de compilação!

Variáveis

- *Variaveis1.java*

Operadores Aritiméticos

+	op1 + op2	Soma op1 e op2; Concatenação.
-	op1 - op2	Subtrai op2 de op1
*	op1 * op2	Multiplica op1 por op2
/	op1 / op2	Divide op1 por op2
%	op1 % op2	Resto da divisão de op1 por op2.
>	op1 > op2	Retorna true se op1 é maior que op2
>=	op1 >= op2	Retorna true se op1 é maior que ou igual a op2
<	op1 < op2	Retorna true se op1 é menor que op2
<=	op1 <= op2	Retorna true se op1 é menor que ou igual a op2
==	op1 == op2	Retorna true se op1 e op2 são iguais.
!=	op1 != op2	Retorna true se op1 e op2 são diferentes

Operadores Lógicos

&&	op1 && op2	E; avalia condicionalmente op2.
||	op1 || op2	OU; avalia condicionalmente op2.
!	!op	Negação; *true*, se op é *false*.
&	op1 & op2	AND; sempre avalia ambas expressões.
|	op1 | op2	OU; sempre avalia ambas expressões.
^	op1 ^ op2	XOU; true, se op1 e op2 têm valores diferentes.

Operadores Unários

++	op++	Incrementa o valor de op em 1; retorna o valor de antes do incremento.
++	++op	Incrementa o valor de op em 1; retorna o valor de depois do incremento.
--	op--	Decrementa o valor de op em 1; retorna o valor de antes do incremento.
--	--op	Decrementa o valor de op em 1; retorna o valor de depois do incremento.

Escrita de dados no console

- Atributo out da classe System
  - Método print e println;
- Exemplos:
  - System.out.print(“Teste”);
  - System.out.println(“Teste”);
- Concatenar com valores de variáveis: “+”
  - System.out.println(“Numero: ”+n);

Leitura de dados no console

- Primeiro se constrói um Scanner:
  - Scanner in = new Scanner(System.in);
- Utiliza os métodos da Classe Scanner para ler a entrada:
  - Ler uma linha de entrada:
    - in.nextLine();
  - Ler um inteiro:
    - in.nextInt();
  - Ler um Ponto Flutuante:
    - in.nextDouble();

Exercício

- Problema:

Crie um programa que recebe uma nota (pela classe Scanner) e:

- Nota 7 ou maior: aprovado
- Entre 5 e 7: tem direito de fazer uma prova de recuperação
- Menor que 5: reprovado direto.

Exercício

- Problema:
- Escrever um programa que calcule a média final de um aluno.
- Observações:
  - Usuário deve digitar as notas do 1º e do 2º Bimestre;
  - A nota do 1º Bimestre tem peso 4
  - A nota do 2º Bimestre tem peso 6;