INTRODUÇÃO AO JAVA

Linguagem JAVA

- Lançada em 1995, por James Gosling da Sum Microsystem

- Linguagem Orientada a Objetos
- Deriva da linguagem C
- Não tem nada a ver com JavaScript

- É ao mesmo tempo um ambiente e uma linguagem de programação

Linguagem Java

Posição mais baixa (desde 2001):
2º Lugar em Março de 2015

May 2020	May 2019	Change	Programming Language	Ratings
1	2	↑	C	17.07%
2	1	↓	Java	16.28%
3	4	↑	Python	9.12%
4	3	↓	C++	6.13%
5	6	↑	C#	4.29%
6	5	↓	Visual Basic	4.18%
7	7		JavaScript	2.68%
8	9	↑	PHP	2.49%
9	8	↓	SQL	2.09%
10	21	⇈	R	1.85%

Fonte: https://tiobe.com/tiobe-index/

Porque usar Java?

- Java funciona em diferentes plataformas
- É uma das linguagens de programação mais populares do mundo
- É fácil de aprender e simples de usar
- É de código aberto e gratuito
- É seguro, rápido e poderoso
- Tem um enorme apoio da comunidade (dezenas de milhões de desenvolvedores)

Compilação

Em uma linguagem de programação como C e Pascal, temos a seguinte situação ao compilar um programa.

Código Fonte → **compila** → **Código Binário**

O código fonte é compilado para código de máquina específico de uma plataforma. Muitas vezes o código fonte é desenvolvido visando uma única plataforma. O executável (binário) resultante será executado pelo sistema operacional e, por esse motivo, deve saber conversar com o S.O em questão.

Temos um código executável para cada sistema operacional. É necessário compilar uma vez para Windows, outra para o Linux, e assim por diante...

Alguns programas utilizam, por exemplo, bibliotecas gráficas. Mas elas são diferentes no Windows, no Linux, o que levaria a necessidade de reprogramação do código para diferentes plataformas.

Compilação em Java

O código-fonte é escrito em arquivos **.java** e para serem compilados.

O **compilador** do Java verifica o código nas regras de sintaxe da linguagem e depois grava ***bytecode*** em arquivos .class.

Bytecode é um conjunto de instruções destinadas a executar em uma **Java Virtual Machine** (JVM).

Compilação em Java

- Se pegarmos o código fonte do JAVA e enviar direto para o computador, ele não vai entender. Assim, é necessário o programa compilador da linguagem, chamado de **JavaC (Java Compiler).**

- O compilador vai gerar um código em **binário**. Diferente das outras linguagens, não se trata de um código executável, mas de um **bytecode**.

- O bytecode vai ser interpretado somente pela JVM, gerando um código compreensível pelo computador. Só então, ele será executado.

Write Once, Run Everywhere

- WORA é a abreviação da expressão “Write Once, Run Anywhere” (Escreva uma Vez, Rode em Qualquer Lugar), recurso aplicável aos programas que possuem a capacidade de serem executados em qualquer sistema operacional ou máquina.

- A Sun Microsystem utiliza essa terminologia para a Linguagem Java e de acordo com esse conceito, o mesmo código deve ser executado em qualquer máquina e, portanto, o código-fonte precisa ser portátil.

- A linguagem Java permite executar o bytecode Java em qualquer computador, independentemente do sistema ou do hardware, usando a JVM (Java Virtual Machine). O bytecode gerado pelo compilador não é específico da plataforma e, portanto, através da JVM pode executar em uma ampla variedade de máquinas e sistemas.

Adaptado de: https://www.w3schools.in/java-questions-answers/write-once-run-anywhere-wora/

JRE
(Java Runtime Environment)

Ambiente de Execução Java. Contém a Java Virtual Machine (JVM).

JDK
(Java Development Kit)

Kit de Desenvolvimento Java. Contém a Linguagem propriamente dita (Java Lang) e o compilador (JavaC).

Vamos Começar...

Olá, Mundo!

Classe Pública "OlaMundo"

```
public class OlaMundo {

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

Método Principal

Comando para Imprimir na Tela (Saída de Dados)

Saída de Dados

Comandos para imprimir dados na tela:

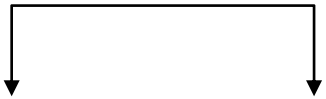
- System.out.println(“Digite seu nome: ”);

  Depois de imprimir uma informação na tela, o método println cria uma nova linha abaixo da atual e então posiciona o cursor na nova linha. Ou seja: o comando é executado e o cursor vai para a linha de baixo para a próxima execução.

- System.out.print(“Digite seu nome: ”);

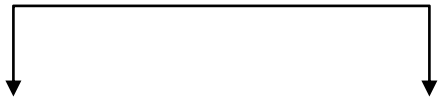
  Depois de imprimir uma informação na tela, diferentemente do método println, o método print não cria uma nova linha abaixo da atual, deixando o cursor na mesma linha onde informação foi impressa. Ou seja: o comando é executado e o cursor permanece na mesma linha para a próxima execução.

Saída de Dados



- System.out.printf(“Sua Nota é %.**2f** \n“, **nota**);

Printf significa print formatado. %f corresponde a variável nota. Para exibir 2 casas decimais na exibição da nota, coloca-se %.2f e modifica-se o número após o ponto para mais ou menos casas decimais.



System.out.print**f**("A Nota de **%s** é %.2f \n", **nome**, nota);

Assim como no exemplo anterior, %s corresponde a uma string.

Os nomes das variáveis, após as aspas, devem ser na ordem de exibição.

- System.out.**format**(“Sua Nota é %.**2f** \n“, **nota**);

Tambem é possível utilizar format com o mesmo resultado do System.out.printf ou System.out.print**f**.

Concatenação

Ato de unir strings pelo operador +, mas não o confunda com o operador de adição que utiliza o mesmo símbolo.

String palavra1 = “Professor";
String palavra2 = “Leandro";
System.out.println(palavra1 + palavra2);

Professor Leandro

Saída

String nome = “Rafael”
System.out.print(“Meu nome é“ + nome);

Meu nome é Rafael

Saída

double media = 7.5
System.out.print(“Sua média: “ + media);

Sua média é 7.5

Saída

Camel Case

- Denominação para a prática de escrever palavras compostas ou frases, onde cada palavra é iniciada com maiúsculas e unidas sem espaços.

- É um padrão largamente utilizado em diversas linguagens de programação, como Java.

- O Java tambem é **Case Sensitive**, fazendo diferenciação entre letras maiúsculas e minúsculas.

Ex: Palavra ≠ palavra

Camel Case

Primeira letra de todas as palavras em maiúsculo:
Classe ou Interface

Ex: MinhaClasse / Classe / ClasseJava

Primeira letra em minúsculo e as demais palavras com a primeira letra em maiúsculo: Atributo / Variável / Método.

Ex: meuAtributo

Ex: minhaVariavel

Se for uma única palavra, tudo fica em minúsculo.

Ex: metodo Ex: atributo

Camel Case

Somente letras minúsculas: Nome de um pacote. Por convenção os pacotes devem ser escritos no formato do endereço de um site ao contrário.

Ex: br.com.primeiroprograma

Ex: br.com.alunocurso_informatica

Todas as letras em maiúsculo: Nome de uma **CONSTANTE**

Ex: CONSTANTE

Ex: VALOR_DE_PI

Operadores Aritiméticos

Operador		Exemplo	
+	Adição	5 + 2	**n1 + n2**
-	Subtração	3 - 1	**n1 – n2**
*	Multiplicação	4 * 9	**n2 * n1**
/	Divisão	8 / 2	**n1 / n2**
%	Resto da Divisão (Módulo)	8 % 2	**n1 % n2**

Operadores de Incremento e Decremento

Operador		Exemplo	
++	Incremento	a = a + 1	**a++**
--	Decremento	b = b - 1	**b--**

Operadores de Atribuição

Operador		Exemplo	
+=	Somar e Atribuir	a = a + b	**a += b**
-=	Subtrair e Atribuir	a = a - b	**a -= b**
*=	Multiplicar e Atribuir	a = a * b	**a *= b**
/=	Dividir e Atribuir	a = a / b	**a /= b**
%=	Resto e Atribuir	a = a % b	**a %= b**

Operadores Relacionais

Operador		Exemplo	
<	Maior que	5 > 3	true
>	Menor que	4 < 7	true
>=	Maior ou Igual a	2 >= 3	false
<=	Menor ou Igual a	3 <= 1	false
==	Igual	3 == 2	false
!=	Diferente	2 != 1	true

Operadores Lógicos

Operador		Exemplo	
&&	E	(5 > 3) && (4 > 2)	**true**
||	OU	(4 >= 6) || (5 <= 2)	**false**
!	NÃO	!true	**false**

X	Y	X && Y
V	V	V
V	F	F
F	V	F
F	F	F

X	Y	X || Y
V	V	V
V	F	V
F	V	V
F	F	F

X	!X
V	F
F	V

Entrada de Dados

Em Java, para recebermos dados do teclado, é utilizada a classe Scanner.

```
import java.util.Scanner;
public class EntradaDados {

   public static void main(String[] args) {

      Scanner teclado = new Scanner(System.in);

      System.out.print("Digite um Nome: ");
      String nome = teclado.nextLine();

      System.out.print("Digite uma Nota: ");
      float nota = teclado.nextFloat();
   }
}
```

É necessário **importar a classe** java.util.scanner. Em geral a própria IDE faz a importação (ou lembra que ela deve ser feita).

Ao usar a classe Scanner, estamos utilizando um conceito de Orientação a Objeto e criando um instância da classe. Uma vez importada, é necessário criar um **objeto para ativar a classe.** Aqui, o objeto recebeu o nome de **teclado**.

Métodos para Ler Valores


```
import java.util.Scanner;
public class EntradaDados {

    public static void main(String[] args) {

     Scanner teclado = new Scanner(System.in);

      System.out.print("Digite um Nome: ");
      String nome = teclado.nextLine();

      System.out.print("Digite uma Nota: ");
      int rm = teclado.nextInt();

      System.out.print("Digite uma Nota: ");
      float nota = teclado.nextFloat();
    }
}
```

nextLine(): método para ler strings

String nome = teclado.nextLine();

nextInt(): método para ler números inteiros (int)

int nota = teclado.nextInt();

nextFloat(): método para ler números reais (float)

float nota = teclado.nextFloat();

Comentários em Java


```
// Comentário de Uma Linha na Linguagem Java
```

```
/* Comentário de
* Multiplas Linhas
* na Linguagem Java
*/
```

```
/**
Comentário JavaDoc
@author Rafael
@version 1.0
@since 2019-07-25
*/
```

Referências Bibliográficas

https://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=010150040706&id=010150040706#.XTm-6-hKjIU

http://www.inf.ufsc.br/~edla.ramos/projeto/geoplano/java.html

https://www.ibm.com/developerworks/br/java/tutorials/j-introtojava1/index.html

https://www.caelum.com.br/apostila-java-orientacao-objetos/o-que-e-java/#onde-usar-e-os-objetivos-do-java

https://www.devmedia.com.br/java-operadores-de-atribuicao-aritmeticos-relacionais-e-logicos/38289

https://www.caelum.com.br/apostila-java-orientacao-objetos/variaveis-primitivas-e-controle-de-fluxo/#casting-e-promoo

http://www.bosontreinamentos.com.br/java/metodos-printf-print-e-println-curso-de-programacao-em-java/

http://www.bosontreinamentos.com.br/java/como-concatenar-strings-em-java/

https://www.cursoemvideo.com

Material desenvolvido pelo
Prof. Rafael da Silva Polato
rafael.polato@etec.sp.gov.br