

Introdução ao desenvolvimento de software

Programação de soluções computacionais

Prof. Dr. Fernando Kakugawa

fernando.kakugawa@animaeducacao.com.br



Agenda

Compilação e execução do programa

Tipos de dados e Operadores

Estrutura de um programa/comando de saída

Material

<https://fkakugawa.github.io/ProgramacaoSolucoesComputacionais/>

Compilação e execução do programa

Linguagem de Máquina

- Os computadores entendem linguagem de máquina

11010 10001 10010

10010 10001 11010

10001 10010 11010

- Seria muito complicado escrever programas dessa forma

A chance de cometer erros é muito grande

Encontrar erros no código é uma tarefa árdua

Linguagem de Máquina

- A linguagem assembly.
 - pode ser usada para escrever código de máquina.
 - representam uma tradução um-para-um com cada código de máquina.
- Exemplo de código assembly:

```
MV    0,    SUM
MV    NUM,  AC
ADD   SUM,  AC
```
- Ainda assim a codificação na linguagem assembly é muito difícil.

Linguagem de Alto Nível

- O ideal é utilizar uma linguagem mais próximo do nosso idioma.
 - linguagem de alto nível.
- Precisaremos utilizar um tradutor (compilador):
 - para traduzir (compilar) o código;
 - linguagem de alto nível Ñ linguagem de baixo nível.

Como é a compilação na linguagem Java?

- Linguagem Java é uma linguagem de alto nível
 - Sintaxe próximo do Inglês
 - programa feito em Java precisa ser compilado para linguagem de máquina.
- Após a compilação gera-se um executável do programa
 - .jar ou .exe (Windows)
- O arquivo executável pode ser executado pelo computador!

Compilação e Interpretação

- Java é uma linguagem híbrida em termos de compilação;
 - Possui um processo de compilação e um processo de interpretação.

Compilação:

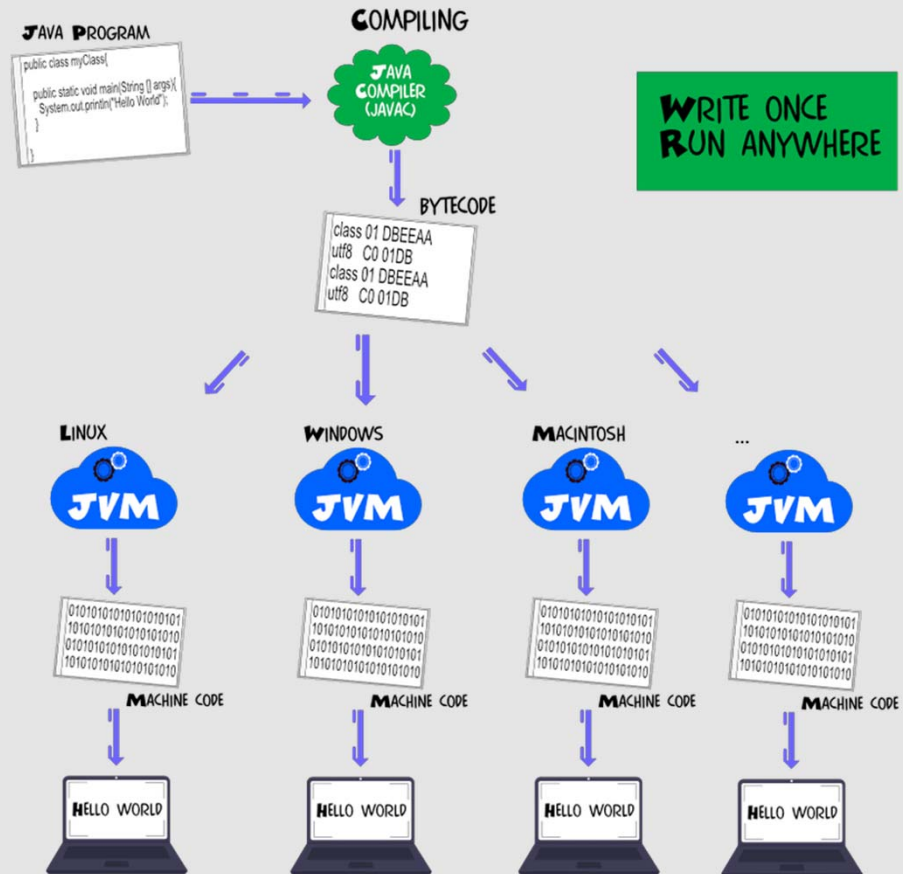
- – Código java é compilado para bytecodes.

Interpretação:

- Bytecodes são interpretados pela Java Virtual Machine (JVM)

Compilação e Interpretação

- Java é uma linguagem multiplataforma;
- Cada sistema operacional precisa ter uma implementação da JVM.



Constantes e tipos de dados

Constantes e tipos de dados

- **Constantes são valores fixos em um algoritmo.**
- **Classificados em:**
 - Literal (caracter ou sequência de caracteres);
 - Numérico (inteiro / real);
 - Lógico (verdadeiro ou falso).

Constantes e tipos de dados

Primitivos		Específicos para Linguagem Java	
Tipos de dados	Definição	Tipos de dados	Capacidade
Literal também conhecido como texto ou caractere	Poderá receber letras, números e símbolos	char/String	16 bits (2 bytes)
inteiro	poderá receber números inteiros positivos ou negativos	int	32 bits (4 bytes) - 2.147.438.648 a 2.147.483.647

Constantes e tipos de dados

Primitivos		Específicos para Linguagem Java	
Tipos de dados	Definição	Tipos de dados	Capacidade
real também conhecido como ponto flutuante	poderá receber números reais, isto é, com casas decimais, positivos ou negativos	float/double	32 bits (de -3,4E-38 até +3,4E+38) 64 bits (8 bytes) (de -1,7E-308 até +1,7E+308)
Lógico também conhecido como booleano	Poderá recebe verdadeiro (1) ou falso (0)	boolean	8 bits (true ou false)

Será que entendi tudo?

- Qual é o tipo de dado/constante a seguir?
 - false lógico (boolean)
 - 21 inteiro (int)
 - “O resultado é:” literal (String)
 - 3.1415 real (double)
 - “true” literal (String)
 - ‘h’ literal (char)

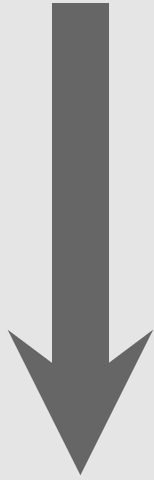
Operadores

Operadores Aritméticos

Operação	Operador	Expressão	Resultado
Adição	+	6 + 4	10
Subtração	-	7 - 9	-2
Multiplicação	*	12 * 3	36
Divisão	/	44/2	22
Módulo (Resto da Divisão)	%	10 %3	1

Precedência de Operadores

menor



+ -

/ *

()

maior

$6 * 4 + 5 - 3$

Precedência de Operadores

menor



maior

+ -

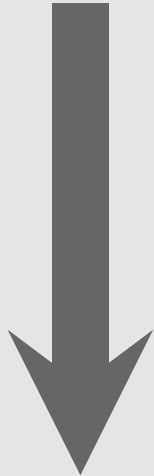
/ *

()

$$\boxed{6 * 4} + 5 - 3$$

Precedência de Operadores

menor



maior

+ -

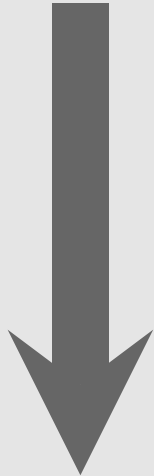
/ *

()

$$\boxed{6 * 4} + 5 - 3$$

Precedência de Operadores

menor



maior

+ -

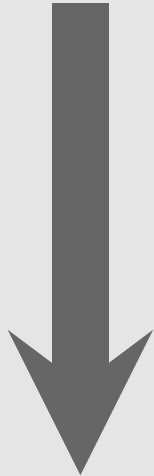
/ *

()

$$6 * 4 + 5 - 3$$

Precedência de Operadores

menor



maior

+ -

/ *

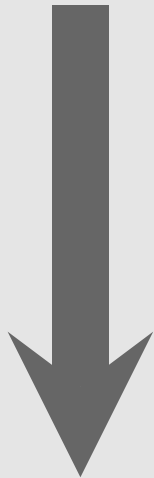
()

$$6 * 4 + 5 - 3$$

$$3 + (2 - 2) * 5$$

Precedência de Operadores

menor



maior

+ -

/ *

()

$$6 * 4 + 5 - 3$$

$$3 + (2 - 2) * 5$$

Precedência de Operadores

menor



maior

+ -

/ *

()

$$6 * 4 + 5 - 3$$

$$3 + (2 - 2) * 5$$

Precedência de Operadores

menor



maior

+ -

/ *

()

$$6 * 4 + 5 - 3$$

$$3 + (2 - 2) * 5$$

Operadores Relacionais

Operador	Representação	Exemplo
Maior que	$>$	$a > b$: Se o valor de a for maior do que o valor de b, retornará true . Senão, retornará false .
Maior ou igual a	\geq	$a \geq b$: Se o valor de a for maior ou igual ao valor de b, retornará true . Senão, retornará false .
Menor que	$<$	$a < b$: Se o valor de a for menor que o valor de b, retornará true . Senão, retornará false .

Operadores Relacionais

Operador	Representação	Exemplo
Menor ou igual a	<code><=</code>	<code>a <= b</code> : Se o valor de <code>a</code> for menor ou igual ao valor de <code>b</code> , retornará true . Senão, retornará false
igual a	<code>==</code>	<code>a == b</code> : se o valor de <code>a</code> for igual ao valor de <code>b</code> , retornará true . Senão, retornará false .
diferente de	<code>!=</code>	<code>a != b</code> : se o valor de <code>a</code> for diferente do valor de <code>b</code> , retornará true . Senão, retornará false .

Operadores Lógicos

- **Utilização:**

- concatenar expressões que estabelecem uma relação de comparação entre valores;
- o resultado dessas expressões é sempre um valor lógico;
- **true** / **false**

- **Tabela Verdade**

- representam todas as possíveis combinações das variáveis de entrada de uma função;
- e os seus respectivos valores de saída.

Operadores Lógicos

- **Operador E (&&)**

- somente resulta em verdadeiro (true):
- se todas as expressões condicionais forem verdadeiras.

- **Operador Ou (||)**

- se apenas uma expressão condicional for verdadeira:
- o resultado é verdadeiro (true).

- **operador Não (!)**

- se a expressão condicional for false:
- o resultado é verdadeiro. Caso contrário é false.

Operadores	Precedência
!	Alta
&&	Média
	Baixa

Operadores Lógicos

- Tabela verdade para os operadores `&&` e `||`
 - `&&` somente resulta em verdadeiro quando todas as sentenças avaliadas forem verdadeiras;
 - `||` somente resulta em falso quando todas as sentenças avaliadas forem falsas.
- Exemplo:

A	B	A && B	A B
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

Operadores Lógicos

- O operador ! faz a negação de uma sentença:
- Exemplo:

A	!A
V	F
F	V

Estrutura de um programa/ comando de saída

Estrutura de um programa Java

- **Java é uma linguagem orientada a objeto.**
 - todo o código de um programa deve estar definido dentro de uma classe.
- **Além disso:**
 - deve estar definido dentro de um **método especial** da classe chamado **main()**.
 - o método **main()** é o ponto de partida na execução de um programa em java.

Estrutura de um programa em Java

```
public class PrimeiroPrograma{  
    public static void main(String[] args){  
        // Aqui serão digitados os passos do algoritmo  
        instrução1;  
        instrução2;  
        instrução3;  
        instrução4;  
    }  
}
```

início da classe

início do main

comentários: servem para
documentar o algoritmo.

São ignorados pelo compilador

fim do main

fim da classe

Saída

- **O resultado do processamento de um algoritmo:**
 - pode ser armazenado em uma variável e/ou...
 - pode ser exibido diretamente em um dispositivo de saída.
- **Monitor** é o dispositivo de saída padrão.

Comando de saída

- Utilizado para escrever dados e resultados na saída padrão (monitor)
- `System.out.println("");`

Código deve estar armazenado em um arquivo com o mesmo nome da classe: ExemploSaida.java

```
1 public class ExemploSaida {  
2  
3     public static void main(String[] args) {  
4         System.out.println("Ola Mundo");  
5         System.out.println("Minha idade é " + 18 + " anos");  
6         System.out.println("O resultado é: " + (3 + 5 <= 22));  
7     }  
8 }
```

```
Ola Mundo  
Minha idade é 18 anos  
O resultado é: true
```

Exercícios

Exercício 1

- **Faça um algoritmo para imprimir em cada linha:**
 - o seu nome completo;
 - o seu RA;
 - o seu curso.

Exercício 2

- Faça um algoritmo para imprimir em cada linha o resultado lógico das seguintes expressões:
 - $((120 - 30) = (3 \wedge 30))$
 - $(\text{não } ((20 \bmod 4) = 1) \text{ ou } (9 \neq 9))$
 - $((5 \bmod \frac{2}{2}) > 3)$
 - $(a = A)$
- **Dica:** Faça uma linha de cada vez.
 - faça, teste, e avance para a próxima;
 - não tente fazer todo código de uma única vez.

Exercício 3

- **Faça um algoritmo para apresentar:**
 - o cálculo da média das seguintes notas 8.0, 7.5, 4.5 e 9.

Exercício 4

- **Faça um algoritmo para apresentar:**
 - o cálculo da área de um quadrado de 350 metros de lado.

Exercício 5

- **Construir um algoritmo para imprimir:**
 - o cálculo da área de um círculo com raio de 5 cm. ($\pi = 3.14159$)
 - área do círculo = πr^2

Exercício 6

- **Faça um algoritmo que calcule o valor de cada expressão a seguir:**
 - $(20 - 15)/2$
 - $2 ^ (5/20) + 30 / (15 ^ 2)$
 - $35 / (6 + 2))$
 - 23 módulo 4

Leitura para casa

- **Ler o capítulo 1 do livro:**
 - Fundamentos da Programação de Computadores;
 - Ana Fernanda Gomes Ascencio;

Obrigado

fernando.kakugawa@animaeducacao.com.br

