

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL

AULA 02

LINGUAGEM DE PROGRAMAÇÃO 2
JAVA



PROF. JANIHERYSON FELIPE

CONTEÚDO DESSA AULA

- INTRODUZIR OS PRINCIPAIS ELEMENTOS DE UM PROGRAMA JAVA**
- CRIAR O PRIMEIRO PROGRAMA JAVA (HELLO WORLD);**
- DISCUSSÕES E DÚVIDAS GERAIS.**

PRIMEIRO PROGRAMA

```
class PrimeiroPrograma{  
    public static void main(String[] args){  
        System.out.println("Ola mundo! ! !");  
    }  
}
```

PRIMEIRO PROGRAMA (HELLO WORLD)

```
class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

- Declaração da Classe
- Nome da Classe
- Método
- Modificadores de Acesso do Método
- Tipo de Retorno do Método
- Nome do Método
- Parâmetros/Argumentos do Método
- Nossa código
- Blocos estão entre {} (chaves)

COMPILENDNO UM PROGRAMA NO TERMINAL...

A compilação pelo terminal ou CMD é feita com os comando abaixo;

Compilar

`javac NomePrograma.java`

Executar

`java NomePrograma`

PRIMEIRO PROGRAMA COM ARGUMENTOS

```
class PrimeiroPrograma{  
    public static void main(String[] args){  
        System.out.println("Você digitou: " + args[0]);  
    }  
}
```

```
PS C:\Users\janih\OneDrive\Área de Trabalho> java PrimeiroProgramma "Francisco"  
Você digitou: Francisco  
PS C:\Users\janih\OneDrive\Área de Trabalho> |
```

EXERCÍCIO

Crie um programa que receba 3 argumentos passados via console (terminal), de forma que seja impressa o texto:

Meu nome é SeuNome
Tenho Sualdade anos
Estudo Bacharealdo em TI

RESOLUÇÃO

```
class PrimeiroPrograma{  
    public static void main(String[] args){  
        System.out.println("Meu nome é " + args[0]);  
        System.out.println("Tenho " + args[1]);  
        System.out.println("Meu curso é " + args[2]);  
    }  
}
```

```
PS C:\Users\janih\OneDrive\Área de Trabalho> javac .\PrimeiroPrograma.java  
PS C:\Users\janih\OneDrive\Área de Trabalho> java PrimeiroProgramma "Francisco" 22 "Bacharelado em TI"  
Meu nome é Francisco  
Tenho 22  
Meu curso é Bacharelado em TI
```

ELEMENTOS BÁSICOS DA LINGUAGEM

VARIÁVEIS

Segue sempre o padrão

- <tipo> <nome da variavel>;
- <tipo> <nome da variavel> = <valor>;
- Primeira letra: a-z _ &;
- Outras letras: a-z _ & 0-9;
- CamelCase;
- Case sensitive

VARIÁVEIS

Tipos primitivos

- byte, short
- int
- long
- float
- double
- char
- boolean

Exemplos

```
int decVal = 26;  
int hexVal = 0x1a;  
int octVal = 032;  
int binVal = 0b11010;
```

Tipos compostos

- String

PALAVRAS RESERVADAS

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

Sequência de Escape	Descrição
\t	tab
\b	backspace
\n	nova linha
\r	retorno de carro
\f	avanço de página
\'	aspas simples
\"	aspas duplas
\\\	barra invertida
\ddd	constante octal
\uxxxx	constante hexadecimal

OPERADORES ARITMÉTICOS

Operador	Descrição
+	adição (e mais unário)
-	subtração (e menos unário)
*	multiplicação
/	divisão
%	módulo
++	incremento (pos ou pré fix)
--	decremento (pos ou pré fix)

OPERADORES RELACIONAIS

Operador	Descrição
<code>==</code>	igual a
<code>!=</code>	diferente de
<code>></code>	maior que
<code><</code>	menor que
<code>>=</code>	maior ou igual que
<code><=</code>	menor ou igual que

OPERADORES LÓGICOS

Operador	Descrição
&	AND
	OR
^	XOR
	OR curto circuito
&&	AND curto circuito
!	NOT

LENDODADOS DO TECLADO

Classe Scanner

Scanner scan = new Scanner (System.in)

import java.util.Scanner;

```
String nome = scan.nextLine();
int idade = scan.nextInt();
double altura = scan.nextDouble();
```

LENDODADOS DO TECLADO

```
import java.util.Scanner;

class Aula01{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        String leitor = scan.nextLine();

        System.out.println(leitor);
    }
}
```

CONDICIONAIS IF-ELSE

```
class Aula01{
    public static void main(String[] args) {
        int x = 4;

        if (x == 2) {
            System.out.println("Igual a 2");
        }else if(x == 3) {
            System.out.println("Igual a 3");
        }else {
            System.out.println("Maior que 3");
        }
    }
}
```

CONDICIONAIS IF-ELSE - ATIVIDADE

Construa um programa que receba três notas no terminal, em seguida calcule media (simples) e fale se o aluno foi aprovado ($\text{media} \geq 7$), está em recuperação ($\text{media} < 7$ e $\text{media} \geq 3$) e reprovado ($\text{media} < 3$);



CONDICIONAIS SWITCH-CASE

```
class Aula01{  
    public static void main(String[] args){  
  
        int dia = 4;  
  
        switch(dia) {  
            case 1: System.out.println("Domingo"); break;  
            case 2: System.out.println("Segunda"); break;  
            case 3: System.out.println("Terça"); break;  
            case 4: System.out.println("Quarta"); break;  
            case 5: System.out.println("Quinta"); break;  
            case 6: System.out.println("Sexta"); break;  
            case 7: System.out.println("Sabado"); break;  
            default: System.out.println("Dia invalido");  
        }  
    }  
}
```

LOOPS WHILE

```
class Aula01{  
    public static void main(String[] args) {  
  
        int x = 0;  
  
        while(x < 10) {  
            System.out.println("Valor de x: " + x);  
            x++;  
  
        }  
    }  
}
```

LOOPS DO - WHILE

```
class Aula01{  
    public static void main(String[] args){  
  
        int x = 10;  
        do {  
  
            System.out.println("Valor de x: " + x);  
            x++;  
  
        }while(x < 10);  
    }  
}
```

LOOPS FOR

```
class Aula01{  
    public static void main(String[] args) {  
  
        for(int x = 0; x < 10; x++) {  
            System.out.println("Valor: " + x);  
        }  
    }  
}
```

LOOPS FOR

```
class Aula01{  
    public static void main(String[] args){  
  
        int x = 0;  
  
        for( ; x < 10 ; ) {  
            System.out.println("Valor: " + x);  
            x++;  
        }  
    }  
}
```

LOOPS FOR-EACH

Cabeçalho:

- Para cada elemento element da coleção collection, faça ...
- Atenção para o tipo do elemento.

□Corpo:

- Ações a serem realizadas.

```
for (elementType element : collection) {  
    loop body  
}
```

LOOPS FOR-EACH

```
/**  
 * Lista todas as notas no bloco de notas.  
 */  
public void listNotes()  
{  
    for (String note : notes) {  
        System.out.println(note);  
    }  
}
```

COMANDOS BREAK E CONTINUE

Break:

- A palavra-chave break é usada para interromper a execução de um loop ou de um switch-case.
- Quando o break é encontrado dentro de um loop, o controle do programa é transferido imediatamente para fora do loop.

Continue:

- A palavra-chave continue é utilizada para pular a execução do restante do código dentro de um loop e avançar para a próxima iteração do loop.
- Quando o continue é encontrado, o código abaixo dele no loop é ignorado, e a próxima iteração do loop é iniciada.

ARRAY E MATRIZES

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
  
        int[] vetorx = {10, 20, 30, 40, 50};  
  
        for (int i : vetorx) {  
            System.out.println(i);  
        }  
    }  
}
```

ARRAY E MATRIZES

```
public class Main {  
    public static void main(String[] args) {  
  
        int[] vetorx = new int[10];  
  
        vetorx[0] = 1;  
        vetorx[1] = 2;  
        vetorx[2] = 3;  
        vetorx[3] = 4;  
  
        for (int i : vetorx) {  
            System.out.println(i);  
        }  
    }  
}
```

ARRAY E MATRIZES

```
public class Main {  
    public static void main(String[] args) {  
  
        int[][] matrizx = new int[10][10];  
  
        matrizx[0][0] = 1;  
        matrizx[0][1] = 2;  
        matrizx[0][2] = 3;  
        matrizx[0][3] = 4;  
  
        for (int i = 0; i < 10; i++) {  
            for(int j = 0; j < 10; j++){  
                System.out.println(matrizx[i][j]);  
            }  
        }  
    }  
}
```

FUNÇÕES EM JAVA

- As funções em java são semelhantes as vistas em linguagens, como C++.
- Toda passagem de parâmetros no Java é feita por valor (Tipos primitivos, como int, float, char, etc., são passados por valor. Isso significa que uma cópia do valor real é passada para o método).

```
public class Exemplo {  
    public static void main(String[] args) {  
        int x = 5;  
        modificarValor(x);  
        System.out.println(x);  
    }  
  
    static void modificarValor(int valor) {  
        valor = 10;  
    }  
}
```

FUNÇÕES EM JAVA

- Passagem por Referência (Passagem da Referência do Objeto por Valor): Quando você passa um objeto para um método, a referência ao objeto é passada por valor. Isso significa que, embora o valor da referência seja copiado, ambas as referências (a original e a copiada) apontam para o mesmo objeto na memória. Portanto, as alterações feitas no objeto dentro do método são refletidas fora do método.

```
public class Pessoa {  
    String nome;  
  
    public Pessoa(String nome) {  
        this.nome = nome;  
    }  
  
    public class Exemplo {  
        public static void main(String[] args) {  
            Pessoa pessoa = new Pessoa("João");  
            modificarReferencia(pessoa);  
            System.out.println(pessoa.nome);  
        }  
  
        static void modificarReferencia(Pessoa p) {  
            p.nome = "Maria";  
        }  
    }  
}
```

FUNÇÕES EM JAVA

- Em Java o uso explícito de ponteiros não é comum, pois a linguagem foi projetada para ser gerenciada automaticamente em relação à memória. A gestão de memória em Java é realizada pelo coletor de lixo (*garbage collector*), o que elimina a necessidade de manipulação direta de ponteiros como ocorre em linguagens de baixo nível, como C e C++.



