

Programação I

Tipos, Variáveis e Estruturas de Controle

Samuel da Silva Feitosa

Aula 3

A Linguagem Java

A Linguagem Java

- Segundo sua especificação, Java poderia ser caracterizado como:

“...uma linguagem de programação de propósito geral, concorrente, baseada em classes e objetos. Projetada para ser simples o bastante para que a maioria dos programadores se torne fluente na linguagem...”



Breve Histórico

- Início em 1991 pela Sun Microsystems.
 - Projeto liderado por James Gosling.
 - Ideia era fazer parte de plataforma para dispositivos portáteis.
 - Foi liberada formalmente em 1995.
 - Kit de desenvolvimento liberado gratuitamente.
- ● Atualmente a plataforma tem 3 segmentos:
 - JavaME, JavaSE, JavaEE.
 - Muitas funcionalidades novas desde seu início.
 - Genéricos, Expressões Lambda, etc.
- Sun foi comprada pela Oracle em 2009.

Características Principais

- Orientada a objetos.
 - Independência de plataforma.
 - Sem ponteiros.
 - Performance.
 - Segurança.
 - Multithreaded.
-
- É uma das linguagens utilizadas no desenvolvimento de aplicações para Android.

Tipos de Dados

Tipos de dados primitivos

- Estabelecem um conjunto de valores que podem ser representados em um programa.

Categoria	Tipo	Tamanho	Intervalo
Inteiro	byte	1	de -128 a +127
	short	2	de -32.768 a +32.767
	int	4	de -2.147.483.648 a +2.147.483.647
	long	8	números inteiros grandes
Real	float	4	números reais
	double	8	números reais grandes
Caractere	char	2	de \u0000 a \uFFFF
Lógico	boolean	1	false e true

Variáveis

- Uma variável é um nome definido pelo programador.
 - Associa-se um valor pertencente a um certo tipo.
 - Nome da variável deve descrever seu propósito.
 - Informações nas variáveis ficam em memória.
- Regras para nomes de variáveis:
 - Sequência de um ou mais caracteres alfanuméricos.
 - Deve iniciar com letra, underline ou cifrão (\$).
 - Não podem ter símbolos gráficos ou espaços.
 - Em Java normalmente use-se o padrão CamelCase.

Declaração e Escopo de Variáveis

- Declarar uma variável é associar um tipo, um nome e um valor.
`<Tipo> <nome1> [, <nome2> [, <nomeN>]];`
- Exemplos:

```
int i = 0;  
double total, valorMedio;
```
- O conjunto de locais no qual uma declaração de variável tem validade é denominada escopo.
 - Variáveis em Java podem ser declaradas em qualquer lugar, tendo seu escopo dentro do bloco.

Entrada e Saída

Entrada e Saída básicas

- Entrada:
 - Vamos utilizar a classe Scanner, que permite a entrada formatada de dados.
 - Principais métodos: `nextInt()`, `nextByte()`, `nextInt()`, `nextFloat()`, `nextLong()`, `nextShort()`.
 - Para ler uma String, usamos `next()` ou `nextLine()`.
- Saída:
 - Saída simples: `System.out.println("Mensagem");`
 - Saída formatada: `System.out.printf("Mensagem");`

Exemplos de Entrada formatada

- Vejamos exemplos do uso da classe Scanner.
 - Ao utilizar esta classe, é necessário fazer a importação da biblioteca: **import java.util.Scanner;**

```
Scanner sc = new Scanner(System.in);

System.out.println("Digite um número: ");
int valor = sc.nextInt();

System.out.println("Valor lido: " + valor);
sc.close();
```

- Caso seja necessário ler diferentes tipos de dados (especialmente Strings), é possível utilizar diferentes construções da classe Scanner.

Exemplos de Saída formatada

- Permite a utilização de marcadores para intercalar valores/variáveis na mensagem.

- Similar ao printf da linguagem C/C++.

```
// Mostra na tela: Hello World! e quebra linha (%n)
System.out.printf("Hello World!%n");
// Mostra na tela: Valor inteiro: 20 e quebra linha
System.out.printf("Valor inteiro: %d%n", 20);
// Mostra na tela: 10,00 / 3,00 = 3,333333 e quebra linha
double x = 10.0, y = 3.0;
System.out.printf("%.2f / %.2f = %f%n", x, y, x / y);
```

- Marcadores comuns: %d para inteiros, %f para reais, %c para caractere e %s para String.

Operadores e Expressões

Operadores e Expressões

- A linguagem Java oferece um conjunto amplo de operadores destinados às operações:
 - Atribuição
 - Aritméticas
 - Lógicas
 - Relacionais
 - Bit a bit (bitwise)
- São representados por um ou mais símbolos, como por exemplo: +, &&, etc.

Operador de Atribuição

- Atribuição é a operação que define o valor de uma variável. Sintaxe:
 - `variável = expressão;`
- Significa: “variável recebe o resultado da expressão”. Exemplos:
 - `boolean result = false;`
 - `i = 0;`
 - `y = a * x + b;`
 - `m = n = p = q = 0;`
 - `soma += num;`

Operadores Aritméticos

- Java possui vários operadores matemáticos:
 - Adição (+), Subtração (-), Multiplicação (*), Divisão (/)
 - Resto da divisão (%), às vezes chamado **mod**.
 - Sinais (+ e -), Incremento (++), Decremento (--).
- Exemplos:
 - `var = 10 + x;`
 - `resto = 45 % 10;`
 - `x = -15;`
 - `--x;`
 - `cont++;`

Operadores Relacionais

- Java também possui operadores relacionais:
 - Igual (==), Diferente (!=), Maior (>), Menor (<).
 - Maior ou igual (>=), Menor ou igual (<=).
- Não confundir o operador de igualdade com o de atribuição.
- Exemplos:
 - `a == b`
 - `idade >= 18`
 - `senha != 1234`

Operadores Lógicos

- Os operadores lógicos conectam logicamente o resultado de diferentes expressões aritméticas ou relacionais. São eles:
 - “E” lógico (&&), “Ou” lógico (||), Negação (!).
 - “Ou exclusivo” (^).
- Exemplos:
 - `idade >= 18 && idade <= 65`
 - `!aprovado`
 - `finalSemana || feriado`

Estruturas de Controle

- Um programa de computador é um conjunto de instruções organizadas de modo a produzir a solução de um determinado problema.
 - Naturalmente, as instruções de um programa são executadas em sequência, o que se denomina fluxo sequencial de execução.
 - Porém, muitas vezes é necessário executar as instruções em uma ordem diferente da sequencial.
 - Necessidade de **repetição** de instruções ou **desvio** do fluxo de execução.

Diretivas e Blocos

- Formalmente, as instruções de um programa são chamadas de diretivas (statements).
 - Escritas uma após a outra e separadas por ponto-e-vírgula e quebra de linha.
- Essas diretivas podem ser tratadas individualmente ou como um conjunto (bloco).
 - Um bloco é um grupo de diretivas delimitadas por chaves ({ e }).
- As estruturas de controle podem operar sobre uma diretiva individual ou um bloco.

Tipos de Estruturas de Controle

- Diferentes estruturas para controlar cada tipo de fluxo de execução.
 - **Estruturas de desvio de fluxo:** está associado à avaliação de uma expressão, como uma tomada de decisão. No Java: **if/else** e **switch**.
 - **Estruturas de repetição simples:** efetua a repetição de diretivas ou blocos um número determinado de vezes. No Java: **for**.
 - **Estruturas de repetição condicionais:** efetua a repetição associada a avaliação de uma condição. No Java: **while** e **do while**.

Outras Estruturas de Controle

- Além dessas estruturas, existem ainda:
 - **Mecanismo de modularização:** representa a divisão de um programa em partes menores. Em Java utilizam-se **classes** e **métodos** para este fim.
 - **Estruturas de controle de erro:** realizam o desvio de fluxo de forma especial, quando é detectado um erro em tempo de execução. Java utiliza o mecanismo de **tratamento de exceções** para este fim.

Estruturas de desvio de fluxo

- As estruturas de desvio de fluxo são capazes de modificar a ordem em que as diretivas de um programa são executadas.
 - O Java possui as diretivas if/else e switch para desvio condicional.
- Além disso, também existe a possibilidade de realizar desvio incondicional.
 - Java oferece os comandos break e continue, geralmente usados em conjunto com o switch ou laços de repetição.

Diretiva IF/ELSE

- O if é uma diretiva de desvio simples do fluxo de execução, capaz de selecionar um entre dois caminhos distintos.
 - Depende do resultado falso ou verdadeiro obtido da expressão lógica associada.
 - Sintaxe:

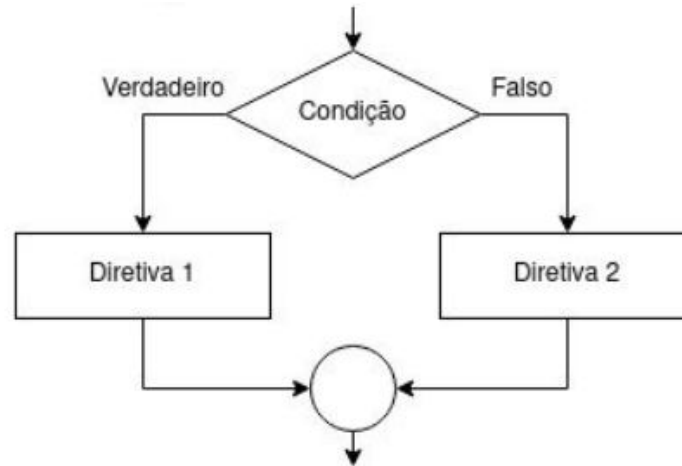
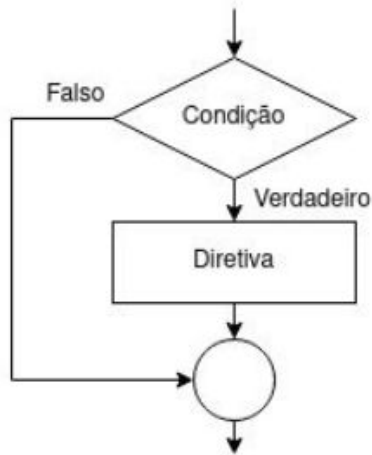
```
if ( <condição> ) {  
    diretiva1;  
}
```

```
if ( <condição> ) {  
    diretiva1;  
} else {  
    diretiva2;  
}
```

Funcionamento Diretiva IF/ELSE

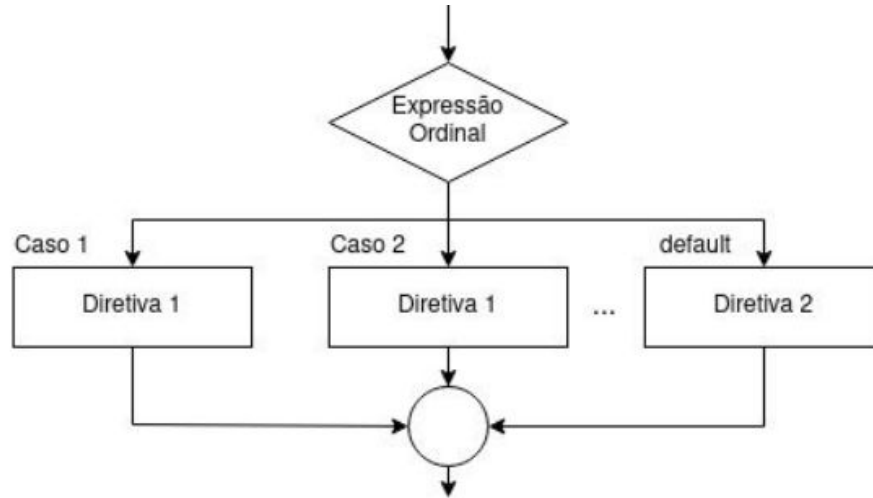
- Também é possível **aninhar** diretivas:
- Funcionamento:

```
if ( <condição1> ) {  
    if ( <condição2> ) {  
        diretiva1;  
    }  
} else {  
    diretiva2;  
}
```



Diretiva SWITCH

- É um desvio múltiplo de fluxo.
 - Avalia uma expressão ordinal para escolher um caminho de execução dentre vários possíveis.
 - Uma expressão é ordinal quando seu resultado pertence a um conjunto. Ex.: números, letras, etc.



Diretiva SWITCH

- Sintaxe:

```
switch ( <expressão_ordinal> ) {  
    case <ordinal1>:  
        diretiva1;  
        break;  
    case <ordinal2>:  
        diretiva2;  
        break;  
    default:  
        diretiva_default;  
}
```

- Se um caso é selecionado, todas as diretivas encontradas a partir de seu início são executadas até o final do corpo do switch ou até encontrar um **break**.
- A partir da versão 7 do Java, o switch passou a aceitar expressões do tipo **String**, flexibilizando sua utilização.

Considerações Finais

- Nesta aula estudamos os principais operadores da linguagem Java.
 - Aritméticos, Relacionais e Lógicos.
- E vimos as principais estruturas de controle
- condicional.
 - Estrutura de seleção simples (if/else).
 - Estrutura de seleção múltipla (switch).
- Todos estes conceitos têm extrema importância na programação de computadores, e são usados em praticamente todas as aplicações.

Plataforma de Exercícios

- Para esta primeira parte da disciplina, vamos utilizar a plataforma *beecrowd* para o desenvolvimento e entrega de exercícios.
 - Acessem: www.beecrowd.com.br
 - Criem uma conta de usuário.
- Fazer login na plataforma.
 - Acessar o endereço: `uoj.app/d-10569`
 - Digitar a chave: `f0mwMiM`
- Resolver os exercícios da lista:
 - Lista 1 - Estruturas condicionais.