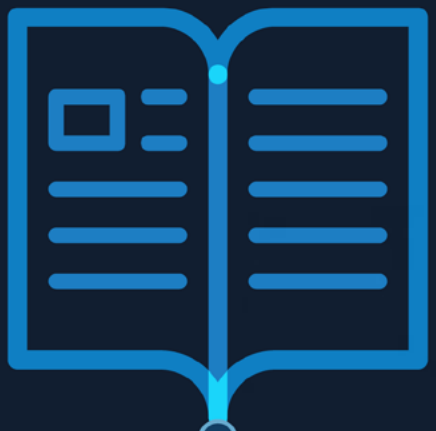
TEXTO DE APOIO



AULA 1

Programação de Sistemas I

Professora Regiane Moreno

EaD | Universidade Presbiteriana Mackenzie

Sumário

PROGRAMAÇÃO DE SISTEMAS I

INTRODUÇÃO AO JAVA

CONCEITOS BÁSICOS

Estrutura básica de um programa: Um programa em Java deve ser criado sempre em uma classe.Dentro dessa classe, tem o método main, e é neste método que você deve escrever seu código.

```
public class <nomeprog>

  <rotinas-e-variaveis opcionais>

 public static void main (String [] args) {

  <comandos>

    }

  <rotinas-e-variaveis opcionais>

}
```

Regras para formação de nomes:

- Devem começar com letra ou _
- Podem ter letras, dígitos ou _
- Nomes de classes devem começar com letra maiúscula
- Nomes de variáveis e subrotinas devem começar com letra minúscula e, se forem nomes compostos, estes devem iniciar com letra maiúscula (Exemplos: verSaldo, nomeAluno, calculaMedia etc.)
- A linguagem JAVA é “Case Sensitive”. Isso significa que letras maiúsculas e minúsculas fazem diferença. Se uma variável com o nome **total** for declarada, ela será diferente de **Total, TOTAL, ToTaL** ou **tOtAl.**

Comentários:

// comentários em uma linha

/* comentários */ bloco de comentário

Tipos primitivos de dados:

Os tipos primitivos são dados que fazem parte da linguagem e não são instâncias de outras classes. Cada tipo de dado tem características e limitações diferentes:

Tipo	Tamanho em bits	Valores	Padrão
[Observação: a representação de um boolean é específica à Java Virtual Machine em cada plataforma.]			
boolean		**true ou false**	
char	16	'\u000'a'\uFFF'(0a65535)	(conjunto de caracteres Unicode ISO)
byte	8	-128a+127(-2^7a 2^7-1)	
short	16	-32.768a+32.767(-2^{15} -2^{15} -1)	
int	32	-2.147.483.648a+2.1474 83+647(2^{31} a2^{31} -1)	
long	64	-9.223+372.036.854.775.808a	
float	32	+9.223.372.036.854.775.807 (-2^{63} a2^{63}-1)	
double	64	.....	(IEEE 754, ponto flutuante)
		Intervalo negativo	
		-3,4028234663852886E+38a	
		-1,40129846432481707e-45	
		Intervalo positivo	
		1,40129846432481707e-45a	
		3,4028234663852886E+38	(IEEE 754, ponto flutuante)
		Intervalo negativo	
		-1,7976931348623157E+308a	
		-4,94065645841246544e-324	
		Intervalo positivo	
		4,94065645841246544e-324a	
		1,7976931348623157E+308	

Fonte: DEITEL(2017).

Declaração de variáveis

Sintaxe: <tipo-variavel><nome variável>;

Exemplos:

int nroEstudantes;

String nome;

double x,y;

boolean fim;

char inicial;

Comando de atribuição

Sintaxe: <variavel> = <valor/expressão>;

Exemplos:

nroEstudantes = 0;

nome = “Ana”;

y= 3.5;

fim = true;

inicial = ‘A’;

Operadores Aritméticos

Os operadores aritméticos comuns (+ - * /) são utilizados no Java para adição, subtração, multiplicação e divisão.

O operador / denota divisão de inteiros se ambos os argumentos forem números inteiros ou, do contrário, divisão de ponto flutuante.

O operador para resto de divisão de números inteiros, que resulta em um valor inteiro, é o %

Incremento/decremento

O Java possui operadores para adicionar ou subtrair 1 de uma variável numérica. O operador de incremento é o ++ e de decremento é o –

Exemplos:

cont = cont + 1; ou cont++;

tot = tot – 1; ou tot--;

Operadores relacionais

Operador	Descrição
==	igualdade
!=	diferença
>, >=	maior, maior ou igual
<, <=	Menor, menor ou igual

Operadores lógicos

Operador	Descrição
&&	e
||	ou
!	não

Precedência de Operadores

Operadores	Associatividade
[ ]. ()(chamada de método)	Da esquerda para direita
! ~ ++ +(unário)-(unário)()	Da direita para esquerda

(coerção) new	Da esquerda para direita
* / %	Da esquerda para direita
+-	Da esquerda para direita
<< >> >>>>	Da esquerda para direita
< <= > >= instanceof	Da esquerda para direita
== ! =	Da esquerda para direita
&	Da esquerda para direita
^	Da esquerda para direita
|	Da esquerda para direita
&&	Da esquerda para direita
||	Da esquerda para direita
?:	Da direita para esquerda
= += -= *= /= %= &= |= ^= <<= >>= >>>>=	Da direita para esquerda

Type Cast:

Algumas conversões são feitas automaticamente no momento de uma atribuição ou de um cálculo .

Exemplo:

```
int v1;

int v2;

double v3;

v1 = 10;

v2 = 30;

v3 = v1 + v2;
```

A conversão da soma que resulta o valor 40 (do tipo **int**) é feita automaticamente para **double**, uma vez que v3 é double

Nas conversões não automáticas, deve-se usar Type Cast:

Exemplo:

```
int v1;

short v2;

v1 = 17;

v2 = (short) v1;


int calc;

calc = (int) v1 * 3.75;
```

Saída de dados:

O comando para saída de dados usa o atributo out, definido na classe System, que, por sua vez, executa o método print/println();

Sintaxe: System.out.print/println(conteúdo);

Formatação da saída de dados:

System.out.printf("%código de formatação");

Alguns códigos de formatação: %d inteiro

%f double

%s String

Exemplo: System.out.printf("%.2f",saldo);

Exemplo básico de programa

```
publicclass Teste {

    publicstaticvoid main (String [] args) {

            /* declaração das variávies */

    double montante;

    double taxa;

    double renda;

                    /* processamento */

    montante = 17000;

    taxa = 0.07;

    renda = montante * taxa;

    montante = montante + renda;

            /* saída dos resultados */

    System.out.print(“A renda obtida = “);

    System.out.printf(“%.2f”,renda);

    System.out.println();

    System.out.print(“Valor do investimento após 1 ano = “);

    System.out.printf(“%.2f”,montante);

    } // fim da main()

} // fim da classe Teste
```

Entrada de dados:

A entrada de dados em Java, utilizando a entrada padrão, pode ser feita porrecursos contidos na biblioteca java.util; para esta finalidade, podemos utilizar aclasse Scanner.

Scanner identificador = new Scanner (System.in);

Exemplo:

Scanner entrada = new Scanner (System.in);

Este identificador(entrada) armazenará o dado digitado, porém precisamosatribuir o dado digitado para uma variável cujo tipo nos interessa.

Isso é feito pelo método next, de forma diferente para cada tipo de dado – int, float,String e outros:

Exemplo:

int idade;

System.out.println("Digite sua idade");

Scanner entrada = new Scanner (System.in);

idade = entrada.nextInt();

Veja as variações do método Scanner para os outros tipos de dados de entrada:

next()	Entrada em formato String com uma única palavra
nextLine()	Entrada em formato String com uma ou várias palavras Exemplo: entrada.nextLine();

nextTipo()	Entrada no formato especificado (tipo), que pode ser: inteiro (int, byte, short ou long), real (float ou double) ou booleano (boolean).
	Exemplos:
	entrada.nextInt();
	entrada.nextFloat();
	entrada.nextDouble();
	entrada.next().charAt(0); //para entrada de dado do tipo char

ESTRUTURAS DE CONTROLE

Bloco de instruções:todo bloco de instruções deve ser delimitado por chaves.

```
{
        <comandos>
}
```

Uma variável declarada dentro de um bloco é invisível e inacessível fora deste bloco (variável local).

Estrutura de Decisão:

```
if <condição> {
<comandos-se-verdadeiro>
}

else {
<comandos-se-falso>
}
```

Exemplo:

```
if (idade!=0 && altura>1)
        System.out.println("Valor aceitos");
```

Estrutura de Seleção múltipla: para comparações de igualdade, em Java, ao invés de utilizar if encadeado, existe a opção de utilizar o comando switch que simplifica a escrita do código.

```
switch (expressão) {


case <valor1>:
    <comandos>
    break;


case <valor2>:
    <comandos>
    break;
.
.


default:
    <commandos>
break;
}
```

Estrutura de Repetição (while): para utilizar laço com teste no início.

```
while (condição) {
```

<comandos>

}

Exemplo:

idade= entrada.nextInt();

while (idade<0){

System.out.println(“A idade não pode ser negativa”);

idade= entrada.nextInt();

}

Estrutura de Repetição (do...while):para utilizar laço com teste no final.

do {

<comandos>

} while (condição);

Exemplo:

do{

idade= entrada.nextInt();

} while (idade<0);

Estrutura de Repetição (for):para fazer uma repetição com variável de controle.

for (<inicialização>;<condição>;<atualização>) {

<comandos>

}

Exemplo:

```
for (i=0; i<5; i++)
    System.out.println(i)
```

Laços (outros comandos):

```
break;          //encerra um loop
continue;       // volta para o início de um looping
```

Exemplo básico de programa:

```
import java.util.Scanner;
publicclass Teste1 {
publicstaticvoid main (String[] args) {
        int diasem;  //armazena o dia da semana
        while (true){
            // o trecho será repetido um número indefinido de vezes
            System.out.println("Informe o dia da semana: ");
```

```
System.out.println(“1=Domingo / 2=segunda / etc. “);
System.out.println(“Digite 0 para encerrar”);
Scanner ent = new Scanner (System.in);
diasem = ent.nextInt();
if (diasem == 0)
        break;          // saída do looping
switch (diasem) {
case 1:
        System.out.println(“Hoje tem Fantastico”);
        break;
case 2:
        System.out.println(“Que preguiça...”);
        break;
case 3:
        System.out.println(“Esta chegando a metade...”);
        break;
case 4:
        System.out.println(“Posso ver a luz no fim do tunel”);
        break;
```

```
                    case 5:
                        System.out.println("Agora sim!");
                        break;
                    case 6:
                        System.out.println("M-A-R-A-V-I-L-H-A");
                        break;
                    case 7:
                        System.out.println("Ufa! Chegou o sábado!");
                        break;
                    default:
                        System.out.println("Dia inválido!!");
                        break;

                    }               // fim do switch

                    System.out.println();
            } // fim do while
System.out.println("Tchau!");

}   //fim da main()
}   // fim da classe Teste
```

```
System.out.println(“1=Domingo / 2=segunda / etc. “);
System.out.println(“Digite 0 para encerrar”);
Scanner ent = new Scanner (System.in);
diasem = ent.nextInt();
if (diasem == 0)
        break;          // saída do looping
switch (diasem) {
case 1:
        System.out.println(“Hoje tem Fantastico”);
        break;
case 2:
        System.out.println(“Que preguiça...”);
        break;
case 3:
        System.out.println(“Esta chegando a metade...”);
        break;
case 4:
        System.out.println(“Posso ver a luz no fim do tunel”);
        break;
case 5:
        System.out.println(“Agora sim!”);
```

```
                    break;
                case 6:
                    System.out.println("M-A-R-A-V-I-L-H-A");
                    break;
                case 7:
                    System.out.println("Ufa! Chegou o sábado!");
                    break;
                default:
                    System.out.println("Dia inválido!!");
                    break;

                }               // fim do switch

                System.out.println();
            } // fim do while
System.out.println("Tchau!");

}   //fim da main()
}   // fim da classe Teste
```

REFERÊNCIAS

DEITEL, H.; DEITEL, P. *Java – Como Programar*. 10. ed. São Paulo: Pearson PrenticeHall, 2017. ISBN 9788543004792

HORSTMANN, C. S.; CORNELL, G. *Core Java*. 8. ed. São Paulo: Pearson Prentice Hall, 2010. ISBN 978857603576