

## PROGRAMAÇÃO I - CCT0827

### Semana Aula: 2

#### Unidade 1 - Introdução à linguagem de programação

### Tema

Características da linguagem Java

### Palavras-chave

Características da linguagem Java, Tipos de dados, Constantes e Variáveis, Operadores e Expressões, Comandos de Controle de Fluxo, Estruturas de Repetição, Entrada e Saída de Dados, Conversão de tipos.

### Objetivos

O aluno deverá ser capaz de:

- Conhecer os tipos de dados, constantes e como declarar variáveis em Java
- Conhecer operadores e expressões da linguagem Java.
- Conhecer e aplicar os comandos das estruturas condicionais.
- Conhecer e aplicar os comandos das estruturas de repetição.
- Desenvolver programas simples em Java que utilizem tipos de dados primitivos, variáveis, constantes, operadores, expressões, saída de dados e estruturas de condicionais e de repetição.

### Estrutura de Conteúdo

#### **Características da linguagem Java**

A linguagem Java tem boa parte de suas características herdadas da linguagem C, muitos dos seus operadores, formação de identificadores, comandos de controle de fluxo e várias outras características são compartilhados entre estas duas linguagens.

Todas as instruções da linguagem Java devem terminar por um símbolo de ponto-e-vírgula “;”.

Os blocos de comandos em Java são delimitados por { (abrir) e } (fechar) chaves.

Em Java são permitidos identificadores que comecem com letras (maiúsculas ou minúsculas), ou um símbolo de “\$” (dólar) ou “\_” (underscore /underline). Números podem ser usados, mas não para iniciar um identificador.

Java é uma linguagem de programação sensível a caixa (alta ou baixa ou case sensitive).

## **Comentários**

1) Comentário de linha: faz com que todo o seu conteúdo, após as duas barras, seja desconsiderado pelo compilador

```
// comentário de linha
```

2) Comentário de bloco: podemos comentar não apenas um trecho de uma linha, mas todo um conjunto de linhas.

```
/*  
  
comentário de linha  
  
*/
```

3) Comentário de documentação: existe uma ferramenta na linguagem Java que é responsável por extrair de um projeto (com várias classes) todos os comentários de documentação e montar um documento com todo este conteúdo.

```
/**  
  
comentário de documentação  
  
*/
```

## **Tipos de dados**

- Primitivos (armazenam apenas valores):
  - Tipo lógico (boolean).
  - Tipo caractere (char).
  - Tipos inteiros (byte, short, int e long).
  - Tipos reais (float e double).
- Tipo Especial
  - Tipo texto (String).

## **Constantes e Variáveis**

Variáveis e constantes em Java devem obrigatoriamente possuir um tipo, isso ocorre porque Java é uma linguagem de programação fortemente tipada.

Variáveis são declaradas através de um tipo e um identificador, sem que sejam necessárias outras informações. A boa prática em programação Java determina que todas as variáveis comecem por letras minúsculas e somente se tiver mais de uma palavra, o inicial da segunda palavra em diante deverá começar por letras maiúsculas.

A definição de constantes precisam do modificador “final”, que indica que uma vez que ocorreu uma atribuição a variável, seu conteúdo não poderá ser mudado.

## Operadores e Expressões

1. Operadores Aritméticos: + , - , \* , / , %
2. Operadores de atribuição compostos: += , -= , \*= , /= , %=
3. Operadores de Incremento e decremento: ++ , --
4. Operadores Relacionais: > , < , <= , >= , == , !=
5. Operadores Lógicos: && , || , !
6. Operadores de Bits: & , ^ , |

## Comandos de Controle de Fluxo

if	if ... else	if ... else if ... else	switch ... case
<b>if</b> (condição) { // instruções; }	<b>if</b> (condição) { // instruções; } <b>else</b> { // instruções; }	<b>if</b> (condição1) { // instruções; } <b>else if</b> (condição2) { //instruções; } <b>else if</b> (condição3) { // instruções; } <b>else</b> { //instruções; }	<b>switch</b> (variável) { // tipo inteiro case 1 : // instruções; break; case 2 : // instruções; break; case 3 : // instruções; break; default: // instruções para condições não previstas  // anteriormente; }

## Estrutura de Repetição

### 1. for

```
for( (1)inicialização; (2)condição de controle; (3)passo) {  
  
    // instruções  
  
}
```

#### Repetição controlada por uma variável:

```
for (int c=1; c<=limite; i++) {  
    instruções;  
}
```

#### Repetição controlada por duas variáveis:

```
for (a=1, b=2; a*b<limite; a++, b+=2) {  
    instruções;  
}
```

#### Repetição sem fim:

```
for ( ; ; ) {  
    instruções;  
}
```

### 2. while

Esta estrutura realiza a repetição de um conjunto de instruções enquanto a condição determinada for verdadeira, caso a condição seja falsa no primeiro teste, nenhuma instrução será executada.

```
while (condição) {  
  
    instruções;  
  
}
```

### 3. do ... while

Esta estrutura de repetição é semelhante à anterior, mas com o diferencial de que a condição ser verificada apenas no final da estrutura, obrigando que as instruções sejam sempre executadas pelo menos uma vez.

```
do  
  
{  
  
instruções;  
  
} while (condição);
```

## Entrada e Saída de Dados

### 1. Entrada de Dados

- Classe Scanner
- É usada para realizar a entrada de dados.
- Pertence ao pacote java.util.
- Por não estar definido em uma biblioteca padrão, para usar a classe Scanner precisamos informar onde ela se encontra. Isso é feito através do comando **import: import java.util.Scanner;**
- É necessário criar o objeto para realizar as entradas de dados:

```
public class EntradaDados {  
  
    public static void main(String[] args) {  
  
        Scanner entrada = new Scanner(System.in);  
  
    }  
  
}
```

- Existem vários métodos na classe Scanner para fazer a entrada de dados:

Para entrada de texto (String):
String nome;  Nome = entrada.nextLine(); // não precisa de conversão, apenas da entrada

Para entrada de valor real:

```
double nota1;  
  
nota1=Double.parseDouble(entrada.nextLine());  
  
// a entrada de dados em texto precisa de conversão para double.
```

Para entrada de valor inteiro:

```
int idade;  
  
idade = Integer.parseInt(entrada.nextLine());  
  
// a entrada de dados em texto precisa de conversão para int.
```

## 2. Saída de Dados

A saída de dados em modo texto pode ser realizada pela classe System, e o método out.print (não pula linha), out.println (pula linha) ou outros métodos:

Apenas uma mensagem:

```
System.out.println("Entre com a Nota A1.....: ");
```

Mensagem e conteúdo de variáveis:

```
System.out.println(" Nome: " + nome + " Idade: " + idade + " Nota 1: " + nota1);
```

## Conversão de Tipos

### 1. Por cast

Usado para converter valores de um tipo para outro, com cast basta indicar o tipo que você quer que a variável ou valor seja convertido colocando o tipo desejado entre parênteses:

```
int a = 10;
```

```
double b = 0;
```

```
b = (double) a;
```

```
a = (int) b;
```

## 2. Por uso de classes para conversão de textos em valores

```
double nota1 = Double.parseDouble("7.8");
```

```
int idade = Integer.parseInt("34");
```

```
float valor = Float.parseFloat("2.15");
```

```
long valor2 = Long.parseLong("3456789");
```

### A boa Prática em Programação Java

A boa prática em programação Java leva em conta um conjunto de regras que facilitam o desenvolvimento de aplicações e melhoram bastante o trabalho em conjunto realizado por equipes.

#### Regra 1. Variáveis auxiliares, atributos, métodos e objetos

Devem ser identificados iniciando por letras minúsculas. Quando houver mais de uma palavra, deve-se começar cada nova palavra com uma letra maiúscula.

Exemplos:

```
int idade;
```

```
int maiorIdade;
```

```
String nome;
```

```
String nomeCompleto;
```

#### Regra 2: Constantes

Devem ser identificados por letras maiúsculas em todo o seu nome, mesmo quando temos mais de uma palavra, todo o identificador deve ficar em maiúsculas.

Exemplos:

```
final int idade;
```

```
final int maiorIdade;
```

```
final String nome;
```

```
final String nomeCompleto;
```

#### Regra 3: Classes e Interfaces (tipo especial de classe)

Devem iniciar por letras maiúsculas. Quando houver mais de uma palavra, deve-se começar cada nova palavra com uma letra maiúscula.

Exemplos:

```
public class Carro { ... }  
  
public class Carro Hibrido { ... }  
  
public interface Basico { ... }  
  
public interface MetodosBasicos { ... }
```

## Estratégias de Aprendizagem

Implementar programas em Java, explorando expressões, estruturas de fluxo de controle e estruturas de repetição.

## Indicação de Leitura Específica

### **Operadores:**

DEITEL, Paul. Java: como programar (Biblioteca Virtual). 10. ed. São Paulo: Pearson, 2017. [Páginas: 41 – 46]

### **Comandos de Controle de Fluxo:**

DEITEL, Paul. Java: como programar (Biblioteca Virtual). 10. ed. São Paulo: Pearson, 2017. [Páginas: 81 – 87]

### **Estrutura de Repetição**

DEITEL, Paul. Java: como programar (Biblioteca Virtual). 10. ed. São Paulo: Pearson, 2017. [Páginas: 120 – 132]

### **Conversão de Tipos:**

FURGERI, Sérgio. Java 8 – ensino didático: desenvolvimento e implementação de aplicações. São Paulo: Érica, 2015. [Páginas: 35 – 38]

### **Entrada e Saída de Dados:**



FURGERI, Sérgio. Java 8 – ensino didático: desenvolvimento e implementação de aplicações. São Paulo: Érica, 2015. [Páginas: 38 – 39]

### Aplicação: articulação teoria e prática

1. Faça um programa em Java que leia uma temperatura em graus Célsius e apresentá-la convertida em graus Fahrenheit. A fórmula de conversão é:  $F = (9 * C + 160) / 5$ . Onde F é a temperatura em Fahrenheit e C é a temperatura em Célsius.

2. Faça um programa em Java que leia o nome de uma pessoa, seu respectivo salário bruto e o valor do salário mínimo, ao final calcular a quantidade de salários mínimos recebida e exibir a mensagem:

Funcionário: xxx recebe yyy salários mínimos.

3. Um hotel cobra R\$ 160.00 a diária e mais uma taxa de serviços. A taxa de serviços é de:

- R\$ 55,00 por diária, se o número de diárias for maior que 15;
- R\$ 60,00 por diária, se o número de diárias for igual a 15;
- R\$ 80,00 por diária, se o número de diárias for menor que 15.

Construa um programa em Java que mostre o nome e o total da conta de um cliente.

4. Um banco concederá um crédito especial aos seus clientes, variável com o saldo médio no último ano. Faça um programa em Java que leia o saldo médio de um cliente e calcule o valor do crédito de acordo com a tabela abaixo. Mostre uma mensagem informando o saldo médio e o valor do crédito.

Saldo médio Percentual

de 0 a 2000.00 nenhum crédito

de 2000.01 a 4000.00 20% do valor do saldo médio

de 4000.01 a 6000.01 30% do valor do saldo médio

acima de 6000.00 40% do valor do saldo médio

5. Faça um programa em Java que leia um número e informe se é um número Primo.

### Considerações Adicionais

