
LTP 1 – Java Básico

Aula 2

Tópicos

- 1 - O objeto de saída da aula passada
- 2 - Outros métodos de saída
 - 2.1 - Usando o `System.out.print()`
 - 2.2 - Usando o `System.out.printf()`
 - 2.3 - Caracter de escape em java
- 3 - Entrando com valores pelo teclado usando Scanner
 - 3.1 - Para ler um string
 - 3.2 - Para ler um inteiro
 - 3.3 - Para ler um float
 - 3.4 - Para ler um double
- 4 - Operadores aritméticos em Java
- 5 - Exercícios

1 - O objeto de saída da aula passada

```
System.out.println("Primeiro Programa!");
```

Para imprimir uma string de caracteres, contida entre aspas duplas, podemos utilizar o objeto `System.out`

O método `System.out.println` exibe uma linha de texto e coloca o cursor no início da linha de baixo (quebra de linha), que termina com um ponto e vírgula.

O argumento desse método é a string entre as aspas duplas.

2 - Outros métodos de saída

2.1 - Usando o `System.out.print()`

Diferente do `System.out.println()`, ao terminar de exibir o conteúdo entre as aspas duplas, o programa não posiciona o cursor no início da próxima linha.

Exemplo1:

```
public class Ex1
{
    public static void main(String args[ ]) {
        System.out.print("Primeiro ");
        System.out.println("Programa");
    }
}
```

Na Tela:

Primeiro Programa

Exemplo2:

```
public class Ex1
{
    public static void main(String args[ ]) {
        System.out.println("Primeiro ");
        System.out.print("Programa");    }
}
```

Na Tela:

Primeiro

Programa

2.2 - Usando o System.out.printf()

Neste caso, semelhante ao C, você pode exibir dados formatados. Com o método printf() em java podemos, por exemplo:

- Arredondar os valores do ponto flutuante
- Representar um ponto flutuante em formato exponencial
- Alinhar à esquerda ou à direita

Exibindo inteiros

%d exibe um inteiro decimal (na base 10)

Exemplo:

```
int x=2;
System.out.printf("valor = %d",x);
```

Na tela:

valor = 2

Observação: não pode usar %i

Exibindo número com ponto flutuante

%f exibe um número com pelo menos um dígito à esquerda

%e (ou %E) exibe um número em notação científica (notação exponencial)

Exemplo:

```
float x=2.236;
System.out.printf("valor = %e\n",x);
System.out.printf("valor = %E\n",x);
System.out.printf("valor = %f\n",x);
```

Na tela:

possible loss of precision

required: float

found: double

Esse erro ocorre por considerar 2.236 um número do tipo double. Para corrigir esse problema poderia declarar x com sendo do tipo double, porém o mais correto seria apenas acrescentar a letra f depois do número para marcar que o mesmo é do tipo float.

Exemplo:

```
float x=2.236f;  
System.out.printf("valor = %e\n",x);  
System.out.printf("valor = %E\n",x);  
System.out.printf("valor = %f\n",x);
```

Na tela:

```
valor = 2.236000e+00  
valor = 2.236000E+00  
valor = 2,236000
```

Imprimindo strings e caracteres

%c exibir caracteres individuais

%s exibir strings (utiliza implicitamente o método toString())

%C exibir caracteres individuais em letras maiúsculas

%S exibir strings em letras maiúsculas

Observações:

- Geralmente a classe String é uma das primeiras que utilizamos pois representa um texto (um conjunto de caracteres) e sua inicialização pode ser feita semelhante a inicialização de variáveis de tipos primitivos.
- Ao inicializar uma variável do tipo char coloque o caracter entre aspas simples.
- Ao inicializar uma variável do tipo String coloque o texto entre aspas duplas.

Exemplo:

```
char letra = 'a';  
String nome = "Salvador Melo";  
System.out.printf("%c\n",letra);  
System.out.printf("%C\n",letra);  
System.out.printf("%s\n",nome);  
System.out.printf("%S\n",nome);
```

Na tela:

```
a  
A  
Salvador Melo  
SALVADOR MELO
```

Imprimindo com larguras e precisões de campos

O tamanho exato de um campo pode ser especificado por uma largura de campo. Se a largura do campo for maior que os dados a serem impressos, os mesmos serão alinhados à direita.

Exemplo:

```
int x=2;  
System.out.printf("%d\n",x);  
System.out.printf("%5d\n",x);  
System.out.printf("%10d\n",x);
```

Na tela:

```
2  
2
```

2

A largura e a precisão podem ser combinadas da seguinte forma:

Exemplo:

```
float x=2.753f;  
System.out.printf("%f\n",x);  
System.out.printf("%10.3f\n",x);  
System.out.printf("%20.1f\n",x);
```

Na tela:

2,753000

2,753

2,8

2.3 - Caracter de escape em java

O Java suporta também uma série de sequências de escape usando a barra invertida (\), que é chamada de caracter de escape. A barra invertida indica que um caracter especial deve ser enviado para a saída quando o próximo caracter é combinado com ela, formando uma sequência de escape, por exemplo:

\n	Inserir uma quebra de linha
\t	Inserir uma tabulação
\'	Mostrar uma aspa simples
\"	Mostrar uma aspa dupla
\\	Mostrar uma barra invertida

3 - Entrando com valores pelo teclado usando Scanner

A classe Scanner é do pacote java.util. Ela possui métodos muito úteis para trabalhar com Strings, em especial, diversos métodos já preparados para pegar números e palavras já formatadas através de expressões regulares.

Quando declaramos uma variável para associar a um objeto, na verdade, essa variável não guarda o objeto, e sim uma maneira de acessá-lo, chamada de referência. É por esse motivo que, diferente dos tipos primitivos como int e float, precisamos utilizar new depois de declarada a variável:

```
Scanner entrada;// variável entrada do tipo Scanner  
entrada = new Scanner (System.in);
```

ou

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

O correto aqui, é dizer que “entrada” se refere a um objeto. Não é correto dizer que entrada é um objeto, pois entrada é uma variável referência, apesar de, depois de um tempo, os programadores java falem “Tenho um objeto “entrada” do tipo Scanner”, mas apenas para encurtar a frase “Tenho uma referência entrada a um objeto do tipo Scanner”.

Para utilizar a classe Scanner você precisa importá-la usando:

```
import java.util.Scanner;
```

3.1 - Para ler um string

```
String var;  
Scanner x;  
x=new Scanner(System.in);  
System.out.println("digite um nome");  
var = x.nextLine();  
System.out.println("valor armazenado: "+var);
```

Na tela:

digite um nome
salvador
valor armazenado: salvador

3.2 - Para ler um inteiro

```
int var;  
Scanner x;  
x=new Scanner(System.in);  
System.out.println("digite um numero inteiro");  
var = x.nextInt();  
System.out.printf("valor armazenado: %d\n",var);
```

Na tela:

digite um numero inteiro
7
valor armazenado: 7

3.3 - Para ler um float

```
float var;  
Scanner x;  
x=new Scanner(System.in);  
System.out.println("digite um numero com ponto flutuante");  
var = x.nextFloat();  
System.out.printf("valor armazenado: %f\n",var);
```

Na tela:

digite um numero com ponto flutuante
2,3
valor armazenado: 2,300000

Observação: ao entrar com a casa decimal, use vírgula e não ponto.

3.4 - Para ler um double

```
double var;  
Scanner x;  
x=new Scanner(System.in);  
System.out.println("digite um numero com ponto flutuante grande");  
var = x.nextDouble();  
System.out.printf("valor armazenado: %f\n",var);
```

Na tela:

digite um numero com ponto flutuante grande

2345,75

valor armazenado: 2345,750000

4 - Operadores aritméticos em Java

São semelhantes àqueles vistos em C:

Soma	+
Subtração	-
Multiplicação	*
Divisão normal	/
Resto da divisão	%

Observações:

- Um quociente de inteiros produz um número inteiro, qualquer parte fracionária do inteiro é descartada e não arredondada;
- Também usamos os mesmos critérios de precedência vistos em C.

5 - Exercícios

a) Crie a classe chamada DeclaracaoVariaveis.java e dentro do método main faça o seguinte:

- Declare uma variável do tipo String para armazenar o nome de um carro;
- Declare uma variável do tipo int para armazenar a velocidade máxima desse carro;
- Declare uma variável do tipo float para armazenar o tempo de aceleração de 0 a 100 desse carro;
- Declare uma variável do tipo double para armazenar o preço desse carro;
- Leia pelo teclado o nome, a velocidade máxima, o tempo de aceleração de 0 a 100 e preço de um carro, e armazene nas referidas variáveis;
- Mostre uma frase com todas essas informações.

b) Desenvolva um programa em Java que lê quatro valores (A, B, C e D), calcula e mostra a diferença entre o produto de A e B e do produto de C e D.

$$\text{Resultado} = (A*B) - (C*D)$$

c) Desenvolva um programa em Java que lê o nome de um vendedor, o salário base dele e o total de vendas efetuadas em um mês. Em seguida o programa deverá mostrar o salário total deste vendedor no final do mês. Obs: o vendedor ganha 5% em cima do total das vendas.

d) Desenvolva um programa em java que lê o nome de um aluno e suas notas das quatro provas. O programa deve mostrar o nome do aluno e a média aritmética das referidas notas.

$$\text{média} = \frac{n1+n2+n3+n4}{4}$$

e) Desenvolva um programa em java que lê uma temperatura em graus Celsius, converte para graus Fahrenheit e apresenta na tela, seguindo esta fórmula:

$$F = \frac{9*C + 160}{5}$$