

Programação Estruturada

Aula 03 - Entrada e saída (E/S) via console

Apresentação

Um dos papéis fundamentais dos programas de computador é a interação com o mundo externo, ou seja, a troca de dados entre computador e usuários ou entre computador e outros equipamentos, então esse processo é comumente chamado de entrada e saída (E/S) de dados. Em Java, o sistema de entrada e saída está estruturado na forma de um conjunto de funções que você verá melhor durante a aula. Existe uma biblioteca de funções de E/S por meio do console e outra por meio de arquivo.

Nesta aula, você irá conhecer e aprender a utilizar a biblioteca de funções para entrada e saída de dados via console, ou seja, a entrada de dados do usuário pelo teclado e a saída de dados para o usuário pela impressão dos dados na tela do computador.



Vídeo 01 - Apresentação

Objetivos

Após a leitura desta aula, você será capaz de:

- Saber utilizar os recursos da linguagem Java para realizar a entrada de dados via console/teclado;
- Saber imprimir dados na tela do usuário (console).

1. Escrita de Strings

Como você já deve ter notado, um comando que você irá usar bastante nas nossas aulas é o **System.out.println**, cujo objetivo é o de escrever dados em forma de texto na saída padrão (o monitor). Por isso, vamos explorar esse comando em um pouco mais de detalhes. Como você está vendo Java de uma forma mais simples, imperativa e não orientada a objetos, considere que o nome do comando é `System.out.println` (na disciplina de POO, você verá que o nome da rotina que imprime na tela é `println` e que `System.out` tem um significado particular). Veja alguns exemplos de como você pode utilizar esse comando a partir do programa a seguir:

```
1 public class ProgramaPrintln {
2     public static void main(String[] args) {
3         int x = 10;
4         double y = 10.2;
5         System.out.println("Esta mensagem será impressa na tela!");
6         System.out.println("O valor da variável x é " + x);
7         System.out.println("Os valores de x e y são " + x + " e " + y);
8         System.out.println("Linha 1\nLinha 2");
9     }
10 }
```

Observe no código do **ProgramaPrintln** o uso do comando `System.out.println` nas linhas 5, 6 e 7. Na linha 5, o comando está recebendo como parâmetro uma `String`, mais especificamente um texto digitado pelo programador. Já no caso das linhas 6 e 7, o parâmetro passado para esse comando é :

"O valor da variável x é " + x

O operador `+` está sendo utilizado nesse caso para concatenar o primeiro argumento (texto "O valor da variável x é") com o valor da variável `x` declarada na linha 3 do programa. O resultado dessa operação é o texto "O valor da variável x é 10", já que o valor da variável `x` é igual a 10. Na linha 7, temos a concatenação de duas variáveis (`x` e `y`) e de dois pedaços de texto ("Os valores das variáveis `x` e `y` são " e " e "). O resultado dessa operação é o texto "Os valores das variáveis `x` e `y` são 10 e 10.2". Por fim, a linha 8 faz uso de um caractere especial (`\n`) que representa uma quebra de linha. Veja como fica o resultado da execução desse programa:

```
1 Esta mensagem será impressa na tela!
2   O valor da variável x é 10
3   Os valores de x e y são 10 e 10.2
4   Linha 1
5   Linha 2
```



Vídeo 02 - Comando Impressão

Outros caracteres especiais importantes são mostrados no Quadro 1.

`\"` Representa o caractere " (aspas duplas)

`\t` Representa uma tabulação (lembra do botão TAB do teclado?)

`\\` Representa o caractere \

`\'` Representa o caractere ' (aspas simples)

Quadro 1 - Exemplos de caracteres especiais da linguagem Java.

Veja abaixo linhas de comando para imprimir na tela uma aspa simples, uma barra, aspas duplas e a tabulação:

```
1 System.out.println("Aspas simples: \' ");
2   System.out.println("Barra invertida: \\ ");
3   System.out.println("Aspas duplas: \\" );
4 System.out.println("Tabulação:\tnome\tsobrenome");
```

A saída da execução desses programas é a seguinte:

```
1 Aspas duplas: "
2   Barra invertida: \
3   Aspas simples: '
4 Tabulação: nome sobrenome
```

Uma outra observação sobre a função `System.out.println` é que existe uma variação dela chamada `System.out.print`, a qual faz a mesma coisa, só que não realiza a quebra de linha. Por exemplo, os dois comandos abaixo são equivalentes:

```
1 System.out.println("Olá!");  
2 System.out.print("Olá!\n ");
```

Dessa forma, geralmente usamos o comando que tem o **ln** no final, pois evita termos que colocar o caractere especial `\n`, que na impressão na tela realiza o efeito de quebra de linha. Faça o teste!



Vídeo 03 - Caracteres Especiais

2. Escrita de outros tipos de valores

O comando `System.out.println` pode ser usado para imprimir os valores de variáveis de tipos primitivos também. Veja o seguinte exemplo de código:

```
1 public class Programa2 {  
2     public static void main(String[] args) {  
3         int a = 3 - 2 * 5;  
4         int b = (3 - 2) * 5;  
5         System.out.println(a);  
6         System.out.println(b);  
7     }  
8 }
```

O que ocorre é que existem na verdade várias rotinas `System.out.println`, cada uma delas recebendo como parâmetro um dos tipos de dados primitivos existentes em Java, além de outros como o do tipo `String`.

Atividade 01

1. Escreva e rode o código do exemplo ProgramaPrintln, no qual mostramos como imprimir valores na tela. Adicione a ele os comandos mostrados para imprimir caracteres especiais, como o de barra invertida e de aspas duplas.
2. Faça um programa que imprima a mensagem **Meu nome é "Nome Sobrenome"**, onde nome e sobrenome são os seus dados pessoais.
3. Crie um programa para calcular e imprimir a área de um retângulo de lados 20 e 30.

3. Leitura do teclado

Você já sabe bem como imprimir valores no console do usuário. Vejamos agora como você pode fazer para ler dados do teclado via console. Para essa tarefa, você usará funções de uma classe chamada de Scanner. Veja um exemplo de seu uso através do código abaixo:

```
1 import java.util.Scanner;
2
3 public class ProgramaLeitorTeclado {
4     public static void main(String[] args) {
5         Scanner leitor = new Scanner(System.in);
6         System.out.println("Digite seu nome:");
7         String nome = leitor.nextLine();
8         System.out.println("Olá, " + nome);
9     }
10 }
```

Para lermos do teclado, precisamos primeiro criar um leitor. Isso é feito na linha 05. Declaramos uma variável do tipo Scanner, que é uma classe definida na biblioteca de Java de nome **java.util**. Para fazer o uso de classes definidas em bibliotecas que não as padrão (java.lang), precisamos importá-las. Isso é feito

através do comando mostrado na linha 01, usando-se a palavra chave *import* seguida do nome da biblioteca e da classe que estamos interessados em usar. Em Java, chamamos essas bibliotecas de pacotes.

Bom, depois de declarar um leitor do tipo Scanner, nós inicializamos seu valor através da instrução **new Scanner**. Essa instrução *new* realiza toda a preparação necessária para que as rotinas e variáveis declaradas na classe Scanner possam funcionar. Além disso, é passado como parâmetro para o Scanner o valor System.in, que representa o sistema de entrada de dados padrão, ou seja, o teclado.

Uma vez inicializado um leitor (Scanner) do teclado, podemos ler dados através de uma de suas várias funções. No programa mostrado como exemplo, na linha 15, usamos a função `nextLine()` para ler uma String do usuário. Observe que a função é provida pelo Scanner, por isso é necessário colocar o nome da variável antes do nome da função, separando-as por um ponto.

Observe, a seguir, a lista das principais funções de leitura de dados da classe Scanner.

<code>nextLine()</code>	Lê um texto digitado pelo usuário
<code>nextInt()</code>	Lê um número inteiro
<code>nextLong()</code>	Lê um número inteiro longo
<code>nextFloat()</code>	Lê um número float
<code>nextDouble()</code>	Lê um número double

Quadro 2 - Comandos para leitura de dados do teclado via classe Scanner.

Veja o uso dessas funções no programa do leitor modificado conforme mostrado a seguir:

```
1 import java.util.Scanner;
2
3 public class ProgramaLeitorTeclado2 {
4     public static void main(String[] args) {
5         Scanner leitor = new Scanner(System.in);
6         System.out.println("Digite seu nome:");
7         String nome = leitor.nextLine();
8         System.out.println("Olá, " + nome + ", digite agora sua idade:");
9         int idade = leitor.nextInt();
10        System.out.println(idade + " anos, muito bem!");
11        System.out.println("Digite agora sua altura:");
12        double altura = leitor.nextDouble();
13        System.out.println("Ok, entendi, você tem " + altura + "m de altura.");
14    }
15 }
```

Note o uso de `nextInt()` para ler números inteiros e de `nextDouble()` para ler números que possuem casas decimais.

Um problema que você pode enfrentar ao usar o `Scanner` é pedir para o usuário digitar um número inteiro e ele digitar um tipo diferente como uma letra, por exemplo. Isso irá gerar um erro na execução do programa, chamado de **`InputMismatchException`**. O tratamento desses casos, porém, será assunto de aula da disciplina de Orientação a Objetos. Por enquanto, vamos considerar que o usuário dos nossos programas digita exatamente o tipo de informação que estamos pedindo.



Vídeo 04 - Leitura Scanner

Atividade 02

1. Rode o exemplo mostrado em ProgramaLeitorTeclado2.
2. Escreva um programa que leia 3 frases digitadas pelo usuário, que entram através do teclado, e que as imprima na tela na ordem inversa de entrada.
3. Escreva um programa que lê o nome de um aluno, o nome de uma disciplina e as notas de 4 provas realizadas nela. Calcule e apresente na tela a média aritmética alcançada por essas notas. Imprima essa informação através de uma mensagem que informe também o nome do aluno e da disciplina cursada.

Conclusão

Bem, você chegou ao final da aula! É importante que tenha aprendido a utilizar todas essas funções para a E/S de dados via console. Você já está preparado para iniciar a próxima aula, na qual irá aprender a utilizar **comandos de seleção**. Realizando a autoavaliação a seguir, você poderá revisar o assunto dado durante toda a aula, sendo assim de grande importância a realização de todas as atividades, pois é através delas que são fixados os conteúdos.

4. Resumo

Nesta aula, você viu as diversas funções disponíveis na linguagem Java para ler dados do teclado e imprimir dados na tela do computador (console). Dentre as principais funções vistas, temos as várias funções `System.out.println`, que recebem parâmetros de tipos diferentes e que são utilizadas para imprimir esses dados (parâmetros) na tela. Existe uma implementação dessa função para cada tipo de dado primitivo existente em Java. Já para leitura de dados, temos a classe `Scanner` que fornece várias operações para leitura de textos e valores, usualmente a partir do teclado. Entre essas funções, temos a `nextInt()`, `nextDouble()` e `nextLine()`, que leem valores de tipos distintos (inteiro, número real e `String`, respectivamente).

5. Autoavaliação

1. Descreva os comandos em Java que podemos usar para ler e escrever dados via console.
2. Descreva com suas palavras a importância de aprender a utilizar a Entrada e Saída de dados via console e de que forma ela irá ajudá-lo no desenvolvimento de programas na linguagem Java.

6. Referências

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores:** algoritmos, Pascal, C/C++ e Java. São Paulo: Editora Pearson, 2008.

THE JAVA tutorials. Disponível em: <http://download.oracle.com/javase/tutorial/>. Acesso em: 6 dez. 2011.