



Instituto Federal
Campus Goiânia

Bacharelado em Sistemas de Informação

POO I

Prof. Dory Gonzaga Rodrigues



Ementa

- Classe Scanner



Ementa > Classe Scanner

A Classe `java.util.Scanner`

Como atribuir valores para uma variável usando o teclado?

Vimos nas disciplinas de programação que utilizam linguagens estruturadas, como C, por exemplo, tem-se a função `scanf()`.

No Java, temos disponível a classe **Scanner** do pacote **java.util**. Essa classe implementa as operações de entrada de dados pelo teclado no console.

A classe Scanner funciona separando a entrada dos textos em blocos, gerando os conhecidos **tokens**, que são sequências de caracteres separados por delimitadores que por padrão correspondem aos espaços em branco, tabulações e mudança de linha.

Com essa classe podem ser convertidos textos para tipos primitivos, sendo que esses textos podem ser considerados como objetos do tipo String, InputStream e arquivos



Ementa > Classe Scanner

A Classe java.util.Scanner

Para utilizar a classe Scanner em uma aplicação Java deve-se proceder da seguinte maneira:

1) importar o pacote java.util:

```
1 import java.util.Scanner;  
2
```

2) Instanciar e criar um objeto Scanner que utiliza um InputStream através da variável IN da classe System:

```
6  
7 Scanner dados = new Scanner(System.in);  
8
```

**Ementa** ➤ **Classe Scanner**

Métodos da Classe Scanner

Veja abaixo como são invocados os principais métodos que correspondem com a assinatura que retorna um valor do tipo que foi invocado. Ou seja, para cada um dos primitivos existe uma chamada do método para retornar o valor especificado na entrada de dados, sempre seguindo o formato `nextTipoDado()`.

`nextInt()` `next()` `nextBoolean();`

`nextFloat()` `nextLine()`

`nextDouble()`

`nextByte();`

`nextLong();`



Ementa > Classe Scanner

Exemplo de uso da Classe Scanner

Abaixo um exemplo simples de entrada de dados, onde o conteúdo lido é atribuído à respectiva variável de tipo primitivo.

```
1 import java.util.Scanner;
2
3 public class ClasseScanner {
4
5     public static void main(String[] args) {
6
7         Scanner dados = new Scanner(System.in);
8
9         System.out.print("Digite um número inteiro: ");
10        int i = dados.nextInt();
11
12        System.out.print("Digite um texto: ");
13        String s = dados.next();
14
15        dados.close();
16    }
17 }
```

**Ementa > Classe Scanner**

Métodos da Classe Scanner

Na tabela abaixo são apresentados os principais métodos da classe Scanner, além dos apresentados anteriormente:

Método	Descrição
<code>close()</code>	Fecha o escaneamento de leitura.
<code>findInLine()</code>	Encontra a próxima ocorrência de um padrão ignorando máscaras ou strings ignorando delimitadores.
<code>hasNext()</code>	Retorna um valor booleano verdadeiro (true) se o objeto Scanner tem mais dados de entrada.
<code>hasNextXyz()</code>	Retorna um valor booleano como verdadeiro (true) se a próxima entrada a qual Xyz pode ser interceptada como Boolean, Byte, Short, Int, Long, Float ou Double.

**Ementa > Classe Scanner**

Métodos da Classe Scanner

Na tabela abaixo são apresentados os principais métodos da classe Scanner, além dos apresentados anteriormente:

Método	Descrição
<code>match()</code>	Retorna o resultado da pesquisa do último objeto Scanner atual.
<code>next()</code>	Procura e retorna a próxima informação do objeto Scanner que satisfazer uma condição.
<code>nextXYZ()</code>	Varre a próxima entrada a qual XYZ pode ser interceptado como boolean, byte, short, int, long, float ou double.
<code>nextLine()</code>	Mostra a linha atual do objeto Scanner e avança para a próxima linha.

**Ementa > Classe Scanner**

Métodos da Classe Scanner

Na tabela abaixo são apresentados os principais métodos da classe Scanner, além dos apresentados anteriormente:

Método	Descrição
<code>radix()</code>	Retorna o índice atual do objeto Scanner.
<code>remove()</code>	Essa operação não é suportada pela implementação de um Iterator.
<code>skip()</code>	Salta para a próxima pesquisa de um padrão especificado ignorando delimitadores.
<code>string()</code>	Retorna uma string que é uma representação do objeto Scanner.



Ementa > Classe Scanner

Exemplo de uso da Classe Scanner

Concluindo o estudo, vamos implementar através das classes Scanner e System a saída de dados (leitura e impressão) do conteúdo de um arquivo texto no console .

```
1+ import java.io.File;
4
5 public class ClasseScannerArq {
6
7-     public static void main(String[] args) throws FileNotFoundException {
8
9         File arquivo = new File("teste.txt");
10        Scanner dados = new Scanner(arquivo);
11        while(dados.hasNext()){
12            System.out.print(dados.nextLine());
13            System.out.println();
14        }
15        dados.close();
16    }
17 }
```