

- +
-
-

Instruções de entrada e saída

Profa. Andréa Aparecida Konzen
Fundamentos de Programação
Escola Politécnica - PUCRS



A close-up photograph of a person's hand using a computer mouse. The mouse has a vibrant, multi-colored light effect with shades of blue, purple, and pink. The hand is positioned over the mouse, with fingers ready to click. The background is dark, making the glowing mouse stand out.

Instruções de Entrada

Como o usuário como deve informar os dados (via teclado)

Scanner

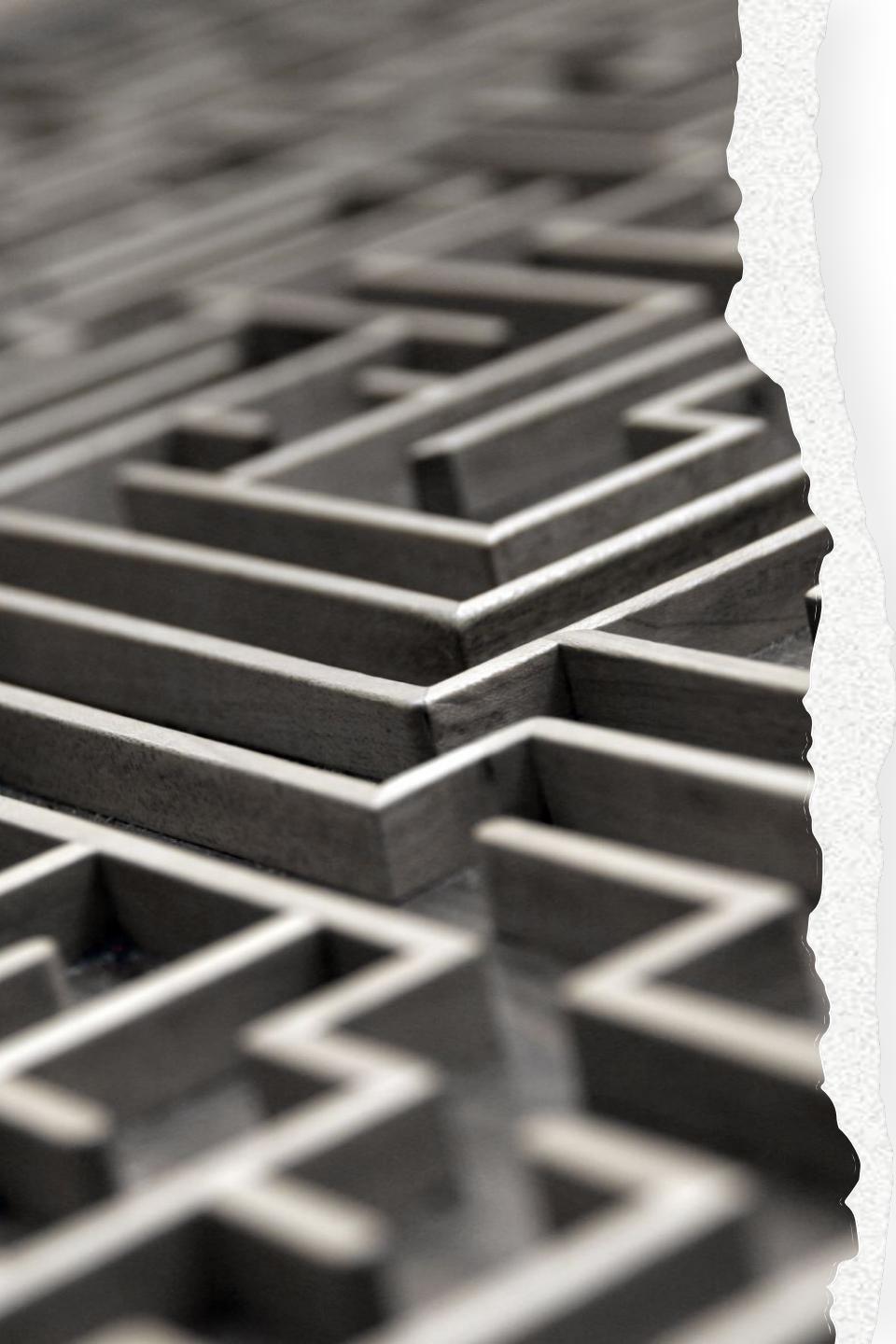
```
import java.util.Scanner
```

Criar o Scanner dentro da classe

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

Receber via teclado o que deve ser guardado na variável

```
nomedavariavel = teclado.nextLine();
```



Instruções de Saída

Como o usuário deve informar a saída (tela)

```
System.out.print("Informe o nome:");
```

```
System.out.print(variavel);
```

Criação das variáveis

```
double valor1;
```

```
int valor2;
```

```
String nome;
```

Criação das variáveis e constantes

1. Constantes

2. Instruções de Entrada

3. Instruções de Saída

- são valores que não se alteram.
- por convenção, seus identificadores são escritos em caixa alta.
- declaração de constantes exige a palavra reservada final.

final tipo identificadorConstante = valor;

Exemplo: **final double TAXA = 2.5;**

Criação das variáveis e constantes

1. Constantes

2. Instruções de Entrada

3. Instruções de Saída

- permitem receber os valores informados pelo usuário.
- usamos a classe **Scanner** para ler os dados informados via teclado.
- necessário
 - importar a classe Scanner :
`import java.util.Scanner;`
 - e instanciar um objeto Scanner:
`Scanner in = new Scanner(System.in);`

Criação das variáveis e constantes

1. Constantes

2. Instruções de Entrada

3. Instruções de Saída

- Para ler um
 - **int** use : **nextInt()**
 - **double** use: **nextDouble()**
 - **String** use: **nextLine()**
- **Exemplo:** **double** raio = in.**nextDouble()** ;

Criação das variáveis e constantes

1. Constantes

2. Instruções de Entrada

3. Instruções de Saída

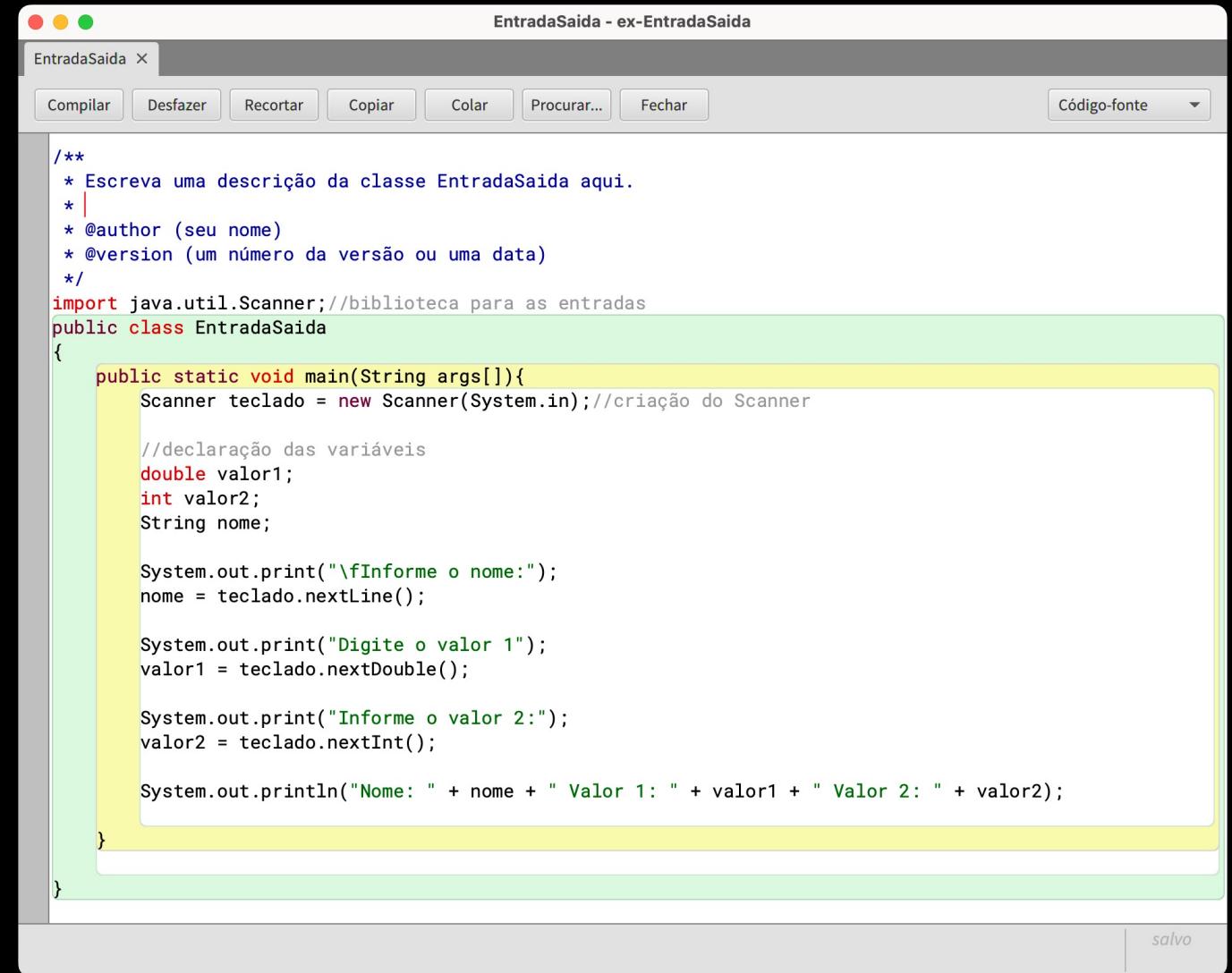
- Escreve na tela informações para o usuário
 - `System.out.print(item+ item+ ..);` ou
 - `System.out.println(item+ item+ ..);`
- **Exemplo:** `System.out.print("Informe o valor do raio:");`

Veja como fica em Java...

```
import java.util.Scanner;
import java.lang.Math;

public classe VolumDaEsfera{
    public static void main(String args[ ]) {
        Scanner teclado = new Scanner(System.in);
        double raio, volume;
        System.out.print("Informe o raio: ");
        raio = teclado.nextDouble();
        volume = 4.0/3 * Math.PI * Math.pow(raio,3);
        System.out.println("Volume: " + volume);
    }
}
```

Exemplo complete em Java (blueJ)



The screenshot shows the blueJ IDE interface with a Java code editor. The title bar reads "EntradaSaida - ex-EntradaSaida". The menu bar includes "EntradaSaida X", "Compilar", "Desfazer", "Recortar", "Copiar", "Colar", "Procurar...", "Fechar", and "Código-fonte". The code editor displays the following Java code:

```
/**  
 * Escreva uma descrição da classe EntradaSaida aqui.  
 * |  
 * @author (seu nome)  
 * @version (um número da versão ou uma data)  
 */  
import java.util.Scanner;//biblioteca para as entradas  
public class EntradaSaida  
{  
    public static void main(String args[]){  
        Scanner teclado = new Scanner(System.in);//criação do Scanner  
  
        //declaração das variáveis  
        double valor1;  
        int valor2;  
        String nome;  
  
        System.out.print("\fInforme o nome:");  
        nome = teclado.nextLine();  
  
        System.out.print("Digite o valor 1");  
        valor1 = teclado.nextDouble();  
  
        System.out.print("Informe o valor 2:");  
        valor2 = teclado.nextInt();  
  
        System.out.println("Nome: " + nome + " Valor 1: " + valor1 + " Valor 2: " + valor2);  
    }  
}
```

A yellow rectangular selection highlights the entire main method block. The status bar at the bottom right shows the word "salvo".