

Funções de entrada e saída

As funções de entrada e saída realizam a conexão do algoritmo com o ambiente externo, recebendo e enviando dados.

Saída de dados

A saída de dados que utilizaremos aqui é a tela. Para mostrar uma mensagem na tela, utilizaremos a função abaixo, acompanhada da mensagem que desejamos apresentar na tela.

```
System.out.println();
```

Exemplo

- Para apresentar na tela a mensagem Olá Mundo:

```
System.out.println("Olá Mundo");
```

Saída na tela:

Olá Mundo

Atenção

Perceba que aqui estamos observando apenas a função de saída. Para que ela possa funcionar de maneira correta no algoritmo, é necessário que ela esteja dentro de uma **estrutura básica**.

```
public class Main{  
    public static void main(String args){  
        System.out.println("Olá Mundo");  
    }  
}
```

Saída na tela:

Olá Mundo

Uso de armazenamento intermediário

No desenvolvimento de algoritmos, podemos utilizar armazenamento intermediário de valores. Para tal, devemos indicar qual é o **tipo de dado** que deverá ser guardado e um **identificador** desse armazenamento.

Informação

O uso desse armazenamento é um tópico de grande importância no desenvolvimento de algoritmos.

Exemplo - Armazenando um valor e apresentando na tela

```
int altura;  
altura = 50; //identificador que permite armazenar um número inteiro  
System.out.println(altura); //saída: 50
```

Saída na tela:

50

É possível também apresentar diversos valores de uma única vez em uma única saída.

Exemplo

```
String nome = "Dunga";  
int idade = 35;  
System.out.println(nome + " tem " + anos " de idade.");
```

Saída na tela:

Dunga tem 35 anos de idade

Exercícios

1. Desenvolva um algoritmo que mostra na tela a mensagem *Hoje é dia de aula de Algoritmos*.
2. Desenvolva um algoritmo que dentro dele possui um armazenamento intermediário de caracteres chamado *mes_ferias_1* e *mes_ferias_2*, onde cada um deve possuir os valores *janeiro* e *julho*. Seu algoritmo deve informar na tela uma mensagem informando que estes são os meses de férias previsto no calendário acadêmico.

Entrada de dados

Para que a entrada de dados possa ser realizado é necessário passar ao algoritmo uma informação adicional, com o identificador em que o valor deverá ser armazenado.

Para a entrada de dados pode-se utilizar a seguinte função:

```
Scanner entrada = new Scanner(System.in); //entrada é um identificador
```

Em seguida é possível ler o dado desejado utilizando `.next()` ou `.nextInt()`

```
entrada.next(); //para dados do tipo caractere
```

ou

```
entrada.nextInt(); //para dados do tipo inteiro
```

Finalizada a leitura de dados, deve-se chamar a função `.close()`.

```
entrada.close();
```

Atenção

Para que a função de entrada possa ser utilizada é necessário importar a biblioteca também é necessário importar a biblioteca `Scanner`.

Para tal, é necessário incluir no início do código a linha

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

Atenção

O código `Scanner entrada = new Scanner(System.in);` realiza uma associação do identificador `entrada` com a entrada padrão (comumente, o teclado).

Após a finalização das entradas, é necessário desassociar o identificador, com a função `entrada.close()`.

Conhecendo a função que realiza a leitura de dados da entrada padrão, devemos informar qual identificador será responsável por armazenar o dado recebido na entrada.

Exemplo - recebendo valores na entrada e armazenando

recebendo da entrada um dado do tipo inteiro:

```
int numero;  
Scanner entrada = new Scanner(System.in);  
numero = entrada.nextInt();  
entrada.close();
```

recebendo da entrada um dado do tipo caractere: javascript String
palavra; Scanner entrada = new Scanner(System.in); palavra =
entrada.next(); entrada.close();

Observação

- Perceba que no exemplo, o nome do usuário será armazenado na região de armazenamento identificada como `nome`.

Exemplo - Pergunte ao usuário seu nome e idade. Em seguida, mostre na tela as informações digitadas.

```
//Identificadores
```

```
String nome;  
int idade;
```

```
//Entrada
```

```
System.out.println("Nome: ");  
nome = entrada.next();  
System.out.println("Idade: ");  
idade = entrada.nextInt();
```

```
//Saída
```

```
System.out.println(nome + " tem " + idade + " anos de idade.");
```

Observação - No exemplo, o nome do usuário será armazenado no espaço identificado como `nome` e a idade em `idade`.

Exercícios

1. Solicite ao usuário que digite um número. Em seguida, mostre na tela o número digitado.
2. Desenvolva um algoritmo que pergunta ao usuário o nome do usuário, e em seguida responde “Boa noite, **user**”, substituindo **user** pelo nome digitado.
3. Faça um algoritmo que pergunta ao usuário a sua idade, e em seguida informa a mensagem “Você tem **X** anos”, substituindo **X** pela idade digitada.