

Controle de Fluxo e Estrutura de Dados

Sumário

- Instruções Condicionais
 - if-else
 - switch
 - for..
 - while
 - do while
 - else Oscilante
 - break continue e break
- Estruturas de Dados
 - Array
 - Simples e Multidimensionais

Controle de Fluxo

■ Instruções Condicionais

- ❑ if-else
- ❑ for
- ❑ switch
- ❑ while
- ❑ do while
- ❑ else Oscilante

- ❑ break e continue

Controle de Fluxo

if else

```
if (expressão booleana)
{ instrução ou bloco de comandos }
else
{ instrução ou bloco de comandos }
```

```
if (cont >= 0)
{
    System.out.Println("Erro !!!");
}
else
{
    System.out.println("Ok !");
}
```

Controle de Fluxo

If's aninhados

if else

```
// ALTERAR:  
if (numero1 == numero2)  
    System.out.println(«Os numeros sao iguais »);  
else  
    { if (numero1 > numero2)  
        maior=numero1;  
      else  
        maior=numero2;  
      System.out.println(« O maior numero e « + maior);  
    }  
} // fim método main  
} // fim classe
```

Controle de Fluxo

for

```
for (expr_inicial; expr_booleana; expr_increm)
    { bloco de comandos }
```

```
for (int x = 0, int y=0; x <10; x++, y--)
{
    System.out.println("Valor do X : " + x);
}
```

Controle de Fluxo

for

```
// Programa Frase invertida

import java.util.Scanner;

public class Palavra {
    public static void main ( String args[] ) {
        int contador=0;
        String arrayString[] = new String[5];
        Scanner entrada = new Scanner(System.in);

        for (contador=0; contador<5; contador++)
        {
            System.out.println("Digite uma palavra: ");
            arrayString[contador] = entrada.nextLine();
        }
        for (contador=4; contador>=0; contador--)
            System.out.println(« Valor: « + arrayString[contador]);

    } // fim método main
} // fim classe
```

Controle de Fluxo

switch

```
switch (expressão short,int,byte ou char)
{
    case expressão2:
        comandos;
        break;
    case expressão3:
        comandos;
        break;
    default:
        comandos;
        break;
}
```


Controle de Fluxo

switch

```
int cor=0;
switch (cor)
{
    case 0:
        setBackground(Color.black);
        break;
    case 2:
        setBackground(Color.red);
        break;
    default:
        setBackground(Color.white);
        break;
}
```

Controle de Fluxo

while

```
while (expr_booleana)
    { bloco de comandos }
```

```
int cont = 0;
while (cont < 100)
{
    System.out.println("contando " + cont);
    cont++;
}
```

Controle de Fluxo

do while

```
do
    { bloco de comandos }
while { condição }
```

```
int x = 0;
do
{
    x++;
} while (x <10);
```

Controle de Fluxo

break e continue

break

```
int i = 0;
while( true ) {
    if (++i==10) break;
    System.out.println(i);
}
```

continue

```
int i = 0;
while ( true ) {
    if (++i%2 == 1)
        continue;
    System.out.println(i);
}
```

```
while (!terminado) {
    passePagina();
    if (alguemChamou == true) {
        break; // caia fora deste loop
    }

    if (paginaDePropaganda == true) {
        continue; // pule esta iteração
    }

    leia();
}

restoDoPrograma();
```

Estruturas de Dados: Array

- Todo array é um objeto
- Para determinarmos o seu tamanho podemos usar o método **length**:

```
int lista [] = new int [10];  
for (int j = 0; j < lista.length; j++)  
{  
    System.out.println(lista[j]);  
}
```

Estruturas de Dados: Array

Declaração

```
char s[ ];           // declaração
s = new char[3];     // Criação
s[0] = 'A';          // atribuição
s[1] = 'B';          // atribuição
s[2] = 'C';          // atribuição
```

Estruturas de Dados: Array

- Declarando, criando e iniciando um array

```
char s[ ]={'A','B','C'};    // declaração,  
                             // criação e  
                             // inicialização
```

Estruturas de Dados: Array

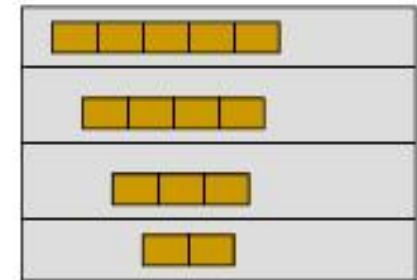
```
import java.util.Scanner;
public class CriaArray {
    public static void main (String[] args) {

        Scanner entrada = new Scanner(System.in);

        // Criação e Definição do tamanho
        String[] nomes = new String[10];
        int[] idades    = new int[10];
        ...
        // Atribuição de valores
        nomes[i] = entrada.next();
        idades[1] = entrada.nextInt();
        ...

    }
}
```


Estruturas de Dados: Array



Multi-dimensionais

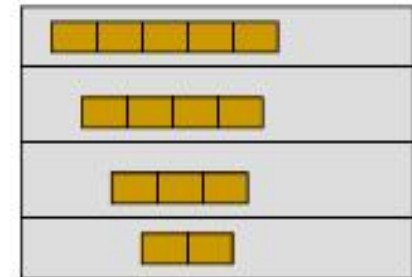
linhas

colunas

```
int duasDim [ ] [ ] = new int [4] [ ];  
duasDim [0] = new int [5];  
duasDim [1] = new int [4];  
duasDim [2] = new int [3];  
duasDim [3] = new int [2];  
duasDim [0][0] = 300;  
duasDim [1][3] = 600;
```

Estruturas de Dados: Array

Multi-dimensionais



```
int duasDim [ ] [ ] = new int [4] [ ];  
duasDim [0] = new int [5];  
duasDim [1] = new int [4];  
duasDim [2] = new int [3];  
duasDim [3] = new int [2];  
duasDim [0][0] = 300;  
duasDim [1][3] = 600;
```

TESTE

```
for (int coluna=0; coluna< 4;coluna++)  
    System.out.println("Valor da posicao: " + duasDim[1][coluna]);
```

Estruturas de Dados: Array

```
import java.util.Scanner;
public class CriaMatriz {
    public static void main (String[] args) {

        Scanner entrada = new Scanner(System.in);

        // Criação e Definição do tamanho
        double[][] consumo = new double[5][2];

        ...

        // Atribuição de valores
        consumo[i][0] = entrada.nextDouble();
        consumo[i][1] = entrada.nextDouble();

        ...

    }
}
```