

# Arrays

P. O. O.

Prof. Grace

---

# Maratona InterFatecs

## Participem e divulguem!



- Enviar nome do time + dados dos participantes (nome + e-mail + CPF + tamanho camiseta)
- Prazo: até dia 14/04.
- Enviar pelo chat do teams ou e-mail [graceapborges@fatecsp.br](mailto:graceapborges@fatecsp.br)

- Saiba mais: Palestra dia 01/04 (2ª feira) das 18h às 19h
- Lab. Informática – Sala 6 (Azul)

# Aula 06

- Arrays
- Arrays Bidimensionais
- Exercícios

# O que são Arrays?

- Arrays ou vetores são objetos que contêm uma coleção de elementos do mesmo tipo
- Cada elemento pode ser acessado ou referenciado individualmente por um índice numérico
- Esse valor inteiro representa a posição do elemento dentro do vetor
- Elementos podem ser de tipo primitivo, objetos ou outros arrays

# Exemplo: Array de 12 elementos

Nome do array (c) →

|         |      |
|---------|------|
| c[ 0 ]  | -45  |
| c[ 1 ]  | 6    |
| c[ 2 ]  | 0    |
| c[ 3 ]  | 72   |
| c[ 4 ]  | 1543 |
| c[ 5 ]  | -89  |
| c[ 6 ]  | 0    |
| c[ 7 ]  | 62   |
| c[ 8 ]  | -3   |
| c[ 9 ]  | 1    |
| c[ 10 ] | 6453 |
| c[ 11 ] | 78   |

Índice (ou subscrito)  
do elemento no array c

# Arrays

- Índice:

- Número da posição entre colchetes.
- Deve ser inteiro positivo ou expressão de inteiro.
- Primeiro elemento tem o índice zero:

```
a = 5;  
b = 6;  
c[ a + b ] += 2;
```

- Adiciona 2 ao elemento c[ 11 ].

# Cuidado com o tipo do índice

- Utilizar um valor de tipo `long` como um índice de *array* resulta em um erro de compilação.
- Um índice deve ser um valor `int` ou um valor de um tipo que possa ser convertido para `int` — `byte`, `short` ou `char`, mas não `long`.

# Declarando e criando arrays

- São criados dinamicamente com a palavra-chave **new**

```
int c[] = new int[ 12 ];
```

Equivalente a:

```
int c[];           // declaração  
c = new int[ 12 ]; // criação
```

- Também podemos criar arrays de objetos:

```
String b[] = new String[ 100 ];
```



# Erro comum de programação

- Especificar o número de elementos entre os colchetes da declaração

Ex. `int c[ 12 ];` // Erro

- Múltiplas variáveis em única declaração:

Ex.:

`int[] a, b, c;` // Um vetor e dois inteiros?

// Ou 3 vetores?

`int a[], b, c;` // Um vetor e dois inteiros

# Exemplo: Criando um vetor

```
public class InitArray
{
    public static void main( String args[] )
    {
        int a[]; // declara um array
        a = new int[ 10 ]; // aloca espaço para o array

        // imprime títulos das colunas
        System.out.printf( "%s %8s\n", "Indice", "Valor" );

        for ( int i = 0; i < a.length; i++ )
            System.out.printf( "%6d %8d\n", i, a[ i ] );

    } // fim do main
} // fim da classe
```

# Definindo constantes

- Também são chamadas
  - constantes identificadas
  - variáveis de leitura (*read-only*).
- Tornam programas mais legíveis
  - Ex.: constante TAM indica tamanho do vetor;

# Uso de Constante e Inicialização com valores calculados

```
public class InitArray3
{
    public static void main( String args[] )
    {
        final int TAM = 10; // declara constante
        int a[] = new int[ TAM ]; // cria o array

        for ( int i = 0; i < TAM; i++ )
            a[ i ] = 2 + 2 * i;

        System.out.printf( "%s %8s\n", "Indice", "Valor" );
        for ( int i = 0; i < TAM; i++ )
            System.out.printf( "%6d %8d\n", i, a[ i ] );

    } // fim do main
} // fim da classe
```

# Cuidados com constantes

- Atribuir um valor a uma constante depois de a variável ter sido inicializada é um erro de compilação.
- Tentar utilizar uma constante antes de ela ser inicializada é um erro de compilação

# Inicializando *array* com uma lista

- Utiliza a *lista inicializadora*:
  - Itens entre chaves ({ }).
  - Itens em uma lista separada por vírgulas.  
`int n[] = { 10, 20, 30, 40, 50 }`
  - Cria um *array* de cinco elementos.
  - Indexa valores de 0, 1, 2, 3, 4.
- Não precisa da palavra-chave **new**.

# Atividade: Dias do mês

```
public class InitArray2
{
    public static void main( String args[] )
    {
        int diasDoMes[] =
            {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

        String meses[] = { "", "Jan", "Fev", "Mar", "Abr", "Mai",
            "Jun", "Jul", "Ago", "Set", "Out", "Nov", "Dez" };

        System.out.printf( "%s %9s\n", "Mês", "Dias" );

        for ( int i = 1; i < diasDoMes.length; i++ )
            System.out.printf( "%02d (%3s) %4d\n", i,
                meses[ i ], diasDoMes[ i ] );

    } // fim do main
} // fim da classe
```

**Implemente**

# Arrays Bidimensionais

- Tabelas com linhas e colunas:
  - Array bidimensional ( $m$  linhas por  $n$  colunas).

|         | Coluna 0    | Coluna 1    | Coluna 2    | Coluna 3    |
|---------|-------------|-------------|-------------|-------------|
| Linha 0 | a[ 0 ][ 0 ] | a[ 0 ][ 1 ] | a[ 0 ][ 2 ] | a[ 0 ][ 3 ] |
| Linha 1 | a[ 1 ][ 0 ] | a[ 1 ][ 1 ] | a[ 1 ][ 2 ] | a[ 1 ][ 3 ] |
| Linha 2 | a[ 2 ][ 0 ] | a[ 2 ][ 1 ] | a[ 2 ][ 2 ] | a[ 2 ][ 3 ] |

Índice de coluna  
Índice de linha  
Nome do array



# Arrays de arrays

- Declarando um array bidimensional `a[2][3]`

```
int a[][] = new int [2][3];
```

- Declarando e inicializando um array bidimensional `b[2][3]`

```
int b[][] = { { 1, 2 , 3 }, {4 , 5 , 6 } };
```

- 1, 2 e 3 inicializam `b[0][0]`, `b[0][1]` e `b[0][2]`.
- 4, 5 e 6 inicializam `b[1][0]`, `b[1][1]` e `b[1][2]`.

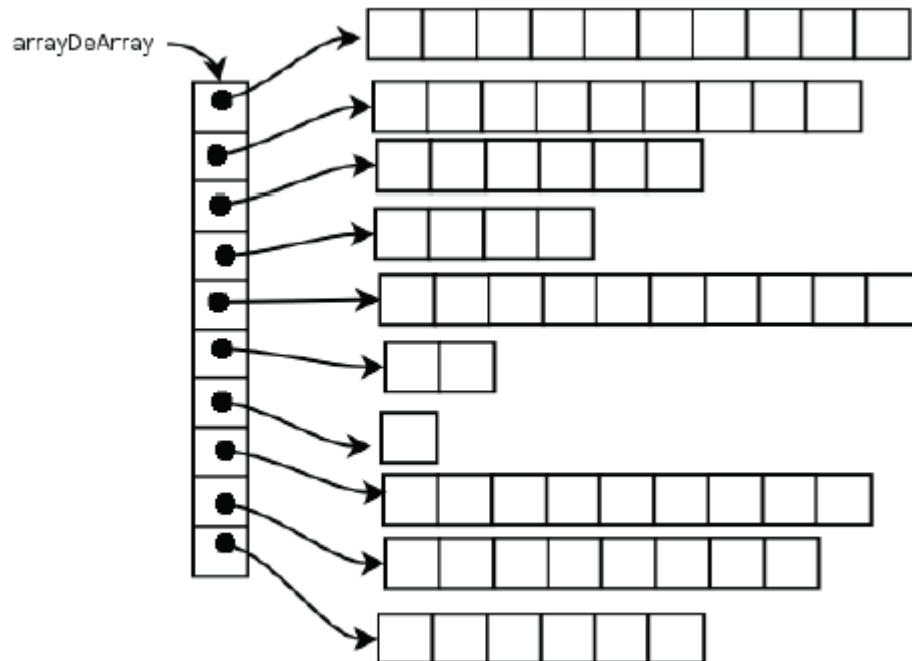
- Comprimentos de linhas diferentes.

```
int b[][] = { { 1, 2 }, { 3, 4, 5 } };
```

- A linha 0 contém elementos 1 e 2.
- A linha 1 contém elementos 3, 4 e 5.

# Arrays com comprimento de linhas diferentes

- Arrays bidimensionais não precisam ser retangulares



# Arrays com comprimento de linhas diferentes

```
1 public class InitArray5
2 {
3     public static void main( String args[] )
4     {
5         int a1[][] = {{1, 2}, {3}, {4, 5, 6}};
6         System.out.println("\nImpressao por linha");
7         exibeArray( a1 );
8     } // fim do main
9
10    //Metodo para exibir Array bidimensional
11    public static void exibeArray(int v[][] )
12    {
13        int lin, col;
14        for (lin = 0; lin < v.length; lin++ )
15        {
16            for ( col = 0; col < v[ lin ].length; col++)
17                System.out.printf("%d  ", v[ lin ][ col ]);
18
19            System.out.println();
20        } // fim do for-externo
21    } // fim do metodo exibeArray
22 } // fim da classe
```

# Exemplo – Notas da Turma

```
1 import java.util.Scanner;
2 public class NotasTurma
3 {
4     public static void main( String args[] )
5     {
6         final int ALUNOS = 5;
7         final int PROVAS = 2;
8
9         int lin, col;
10        double media;
11
12        double notas[][] = new double[ ALUNOS ][ PROVAS ];
13        Scanner entrada = new Scanner(System.in);
14
15        for (lin = 0; lin < ALUNOS; lin++ )
16        {
17            System.out.printf("\nDigite as notas do %do aluno: ", lin+1);
18            for ( col = 0; col < PROVAS; col++)
19                notas[ lin ][ col ] = entrada.nextDouble();
20        } // fim do for-externo
```

# Exemplo – Notas da Turma (cont.)

```
21
22     System.out.println();
23     System.out.println("Notas e Média" );
24
25     for (lin = 0; lin < ALUNOS; lin++ )
26     {
27         System.out.printf("\nAluno %d: ", lin+1);
28         media = 0;
29         for (col = 0; col < PROVAS; col++)
30         {
31             System.out.printf("%2.2f ", notas[ lin ][ col ]);
32             media += notas[ lin ][ col ];
33         } // fim do for-interno
34
35         System.out.printf(" - Média %2.2f ", media/PROVAS);
36     } // fim do for-externo
37
38     } // fim do main
39 } // fim da classe
```

# Atividade – Média da Turma

- Altere o programa anterior, de modo que o mesmo calcule a média da turma.
- Durante a impressão das notas, além de exibir a média de cada aluno, deve ser informado se o aluno ficou, acima, abaixo ou na média da turma.
- Exemplo:
  - Supondo que a média da turma seja 8.2:
  - Aluno 1: 6.0 10.0 – Média: 8.0 – Abaixo da média
- Se necessário, altere as dimensões do *array*.

# Envio de Atividades

- Enviar pelo TEAMS
  - Dias do mês
  - Médias da Turma