

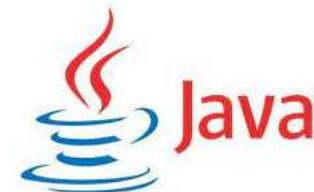


Residência em Tecnologia da Informação e Comunicação

Funções e Métodos em Java

Professor:

Alvaro Degas Coelho



INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



ARRAY

INSTITUIÇÃO EXECUTORA

CEPEDI UESC

COORDENADORA

MCTI FUTURO Softex

APOIO

GOVERNO FEDERAL
MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INovação

BRASIL
UNião e Reconstrução

Definição

Dentro de um bloco, podemos declarar variáveis e usá-las.

```
int idade1;  
int idade2;  
int idade3;  
...  
int idade9;  
int idade10;
```

- Quando são criadas mais de uma variável, para armazenar um mesmo tipo de informação

com um mesmo tipo de dados, poderia ser substituído por um array. O Exemplo acima

poderia ser substituído por:

```
int [ ] idades;
```



Definição

O int [] é um tipo. Um array é sempre um objeto, portanto, a variável idades é uma referência. Vamos precisar criar um objeto para poder usar o array. Como criamos o objeto-array?

```
idades = new int[10];
```



Residência
em Software

Definição

Aqui o que fizemos foi criar uma array de int de 10 posições, e atribuir o endereço o qual ela foi criada. Agora podemos acessar e atribuir valores dentro de cada posição do array.

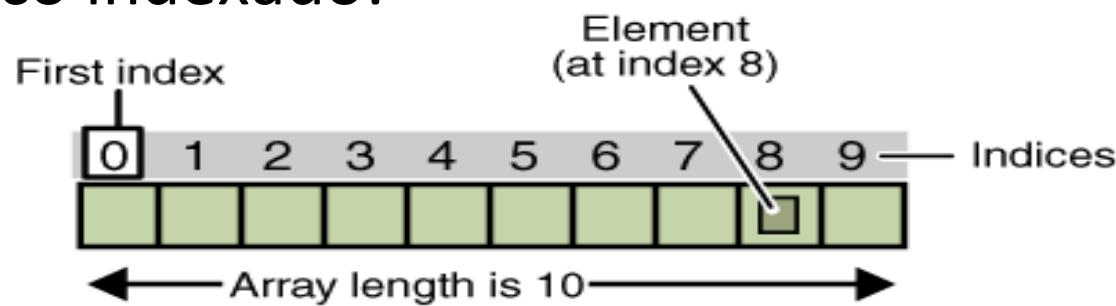
```
idades[5] = 10;
```

O código acima altera a sexta posição do array. No Java, os índices do array vão de 0 a $n-1$, onde n é o tamanho dado no momento que você criou a array. Se você tentar acessar uma posição fora desse alcance, um erro ocorrerá durante a execução.



Definição

- Objeto que armazena um número fixo de valores de um mesmo tipo
- A quantidade de elementos é definida na criação do *array*
- Os elementos são armazenados em um único bloco indexado:





Residência
em Software

Características de um array

Um array é útil quando queremos agrupar objetos do mesmo tipo. Possui as seguintes características:

- Todos os elementos devem ser do mesmo tipo
- Cada elemento pode armazenar um único item
- Os itens podem ser primitivos ou referências a objetos
- O tamanho do array é fixado quando de sua criação



Residência
em Software

Características de um array

Arrays podem conter qualquer tipo de elemento valorado (tipos primitivos ou objetos), mas você não pode armazenar diferentes tipos em um simples array.

Ou seja, você pode ter um array de inteiros, ou um array de Strings, ou um array de array,
mas você não pode ter um array que contenha ambos os objetos Strings e inteiros.



Declaração e Inicialização

- Pode ser declarado e inicializado ao mesmo tempo;

```
int n[] = new int[3];
```

- Pode ser inicializado com {};

```
int a[] = new int[] {1, 2, 3, 5};
```

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



Exemplo 1

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int n[] = new int[3];  
    n[0] = 1;  
    n[1] = 10;  
    n[2] = 100;  
    for(int i = 0; i < n.length; i++) {  
        System.out.println(n[i]);  
    }  
}
```

Exemplo 2

- No exemplo ao lado, um *array* unidimensional de 5 elementos do tipo *double* está sendo declarado e inicializado.
- No “*for*”, uma variável de controle (por exemplo, o famoso “*i*”) acessa os elementos do *array*, atuando como um índice

Main.java

```
public static void main(String[] args) {
    // TODO code application logic here
    double n[] = new double[5];
    n[0] = 2.45;
    n[1] = 1.3;
    n[2] = 14;
    n[3] = 15.1;
    n[4] = 1.9;
    for(int i = 0; i < n.length; i++){
        System.out.println(n[i]);
    }
}
```

29:20 |INS|

Output - jappArrays (run)

```
init:
deps-jar:
Compiling 1 source file to E:\DOCS\FATEC\IGN - 2006
compile:
run:
2.45
1.3
14.0
15.1
1.9
BUILD SUCCESSFUL (total time: 0 seconds)
```

INSTITUIÇÃO EXECUTORA

COORDENADORA

APOIO



Exemplo 3

- Neste exemplo, um array de Strings armazena alguns nomes e um for os mostra na tela

```
public class Exemplo1 {  
  
    public static void main(String[] args) {  
        String nomes[] = new String[3];  
        nomes[0] = "Flabiana";  
        nomes[1]= "Felipe";  
        nomes[2] = "Madeira";  
        for(int i = 0; i<nomes.length;i++){  
            System.out.println(nomes[i]);  
        }  
    }  
}
```



Residência
em Software

Exemplo 4

- Declarar um vetor com 10 posições já inicializadas com valores de 1 a 10 e apresente em tela este mesmo vetor:



Residência
em Software

```
public class Array1 {  
    public static void main (String args [ ]) {  
        int[ ] numeros1a10 = {1,2,3,4,5,6,7,8,9,10};  
        for (int i =0; i < numeros1a10.length; i++) {  
            System.out.println( " Conteudo do Array " + numeros1a10 [i]);  
        }  
    }  
}
```



Residência
em Software

Exemplo 5

Um outro exemplo, o que este faz ?

```
public class Array2 {  
    public static void main(String args[]) {  
        int[] idades = new int[10];  
        for (int i = 0; i < 10; i++) {  
            idades[i] = i * 10;  
        }  
        for (int i = 0; i < 10; i++) {  
            System.out.println (idades[i]);  
        }  
    }  
}
```



Residência
em Software

```
class AlgumaClasse{
    public static void main(String args[]) {
        int[] idades = new int[10]; //
        for (int i = 0; i < 10; i++) {
            idades[i] = i * 10;
        }
        // imprimindo toda a array
        for (int x : idades) {
            System.out.println(x);
        }
    }
}
```

Exemplo 6



Residência
em Software

Quais as limitações dos Arrays





Residência
em Software

Arrays não podem mudar de tamanho

- A partir do momento que uma array foi criada, ela não pode mudar de tamanho. Se você precisar de mais espaço, será necessário criar uma nova array, e antes de se referenciar para ela, copie os elementos da array velha.



Dimensões

- O par de colchetes de um *array[]* define a dimensão do mesmo;
- A dimensão representa a indexação do *array*;
- Isso significa que um vetor pode ter mais de uma dimensão, de forma que sua indexação pode ser composta. Nesse caso, costumamos chamá-lo de “matriz”;
- Um *array* de uma dimensão é chamado comumente de “vetor”.



Residência
em Software

Matriz

A partir do conceito de vetor, fica mais claro entender a matriz. Uma matriz é um conjunto de variáveis, cada uma com um valor associado, como se fossem variáveis simples, porém todas possuem o mesmo nome. A diferença é que a matriz pode ser considerada um vetor bidimensional, ou seja, possui duas dimensões.

Da mesma maneira que em um vetor, os índices são utilizados para referenciar os elementos deste conjunto.

O conceito de matriz na programação é equivalente ao da matemática.



Residência
em Software
Data [31]
Mês int[12]

Matriz-Exemplo

Data [31][12]

Data [0,0]= 1,1 (1 de Janeiro);

Data [0,1]= 1,2 (1 de Fevereiro);

Data [0,2]= 1,3 (1 de Março);

Data [1,0]= 2,1 (2 de Janeiro);

Data [2,1]=



Residência
em Software

Matriz -Exemplo 1

- Repare no exemplo que temos um *array* com dois pares de colchetes
- A quantidade de elementos do *array* pode ser definida por D1 x D2 x... Dn
- No exemplo, dois “for” aninhados “varrem” o *array* através de duas variáveis de controle, atuando com um índice composto de duas dimensões

The screenshot shows an IDE interface with a code editor and a terminal window.

Code Editor:

```
public static void main(String[] args) {
    // TODO code application logic here
    int[][] a2 = { { 1, 2, 3, 4},
                   { 1, 2, 3, 4} };

    for(int i = 0; i < a2.length; i++){
        for(int j = 0; j < a2[i].length; j++){
            System.out.print(a2[i][j] + ", ");
        }
        System.out.println("");
    }
}
```

Terminal Output:

```
32:36 | INS | Output - jappArrays (run)
init:
deps-jar:
Compiling 1 source file to E:\DOCS\FATEC\IGN - 2006-2\04
compile:
run:
1, 2, 3, 4,
1, 2, 3, 4,
BUILD SUCCESSFUL (total time: 0 seconds)
```



Residência
em Software

Matriz -Exemplo 2

- Preencha uma matriz de 10 linhas por 15 colunas com números aleatórios de 1000 a 2000. Em seguida exiba todo o seu conteúdo.
 - Gerar números aleatórios em Java?
 - Google!



package exerMatrizes;

Residência
em Software

public class Principal {

 public static void main(String[] args) {

 // Declaração da matriz.

 int m [][] = new int [10][15];

 // preenchimento da matriz com numeros de 1000 a 2000.(Na verdade só até 1999)

 for (int l = 0; l < 10; l++){

 for (int c = 0; c < 15; c++){

 m[l][c] = (int) (Math.random() * 1000) + 1000;

 }

 }

 //Exibição dos numeros contidos na matriz

 for (int l = 0; l < 10; l++){

 for (int c = 0; c < 15; c++){

 System.out.print(m[l][c]+ " ");

}

Matriz -Exemplo 2- Resolução