



Revisão: Vetores ou Arrays

LPA

Integrado em Informática -IFSP - Hortolândia



Objetivo

- Ao final desta aula você deve responder:
 1. O funcionamento e sintaxe da estruturas de controle:
 - repetição e
 - seleção.
 2. Sobre **vetor**:
 - quais suas características e *para que serve*.
 - qual a sua importância como **estrutura de dados** para armazenar informações.
 - como **declarar**, **inicializar** e **iterar**(percorrer) vetores.
 - como utilizar a instrução **for** aprimorada para iterar sobre vetores.



Conteúdo

- Estrutura de controle: repetição e seleção
- Vetores ou Arrays: declaração, inicialização e manipulação de vetores (revisão)
- A estrutura for aprimorada

Cap. 7 Array e ArrayLists – Java Como Programar (Deitel & Deitel)

Vetor: conceito

- Vetor ou Array: é um **grupo de variáveis** que contém valores **todos** do mesmo **tipo**. Ou seja, uma **estrutura de dados** (coleção de *itens de dados relacionados*) do mesmo tipo.
- O conteúdo do vetor é **manipulado** por meio do índice que indica a posição do elemento no vetor, começando por zero.

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
Índice (ou subcrito) do elemento no array c	c[11]	78



Declaração

vetor

- Declaração:

```
int [] c ;
```

- O *vetor* é um objetos.
- Os elementos dos vetores podem ser de tipos *primitivos* ou *tipos por referência*.
- O *índice* precisa ser um valor int ou um valor de tipo que possa ser promovido a int (byte, short, char).

c[0]	
c[1]	
c[2]	
c[3]	
c[4]	
c[5]	
c[6]	
c[7]	



Declaração e instanciação (criação) vetor

// declara a variável c como referência // para
um vetor de inteiro

```
int [] c;
```

// cria/instancia o vetor e atribui a
// variável c

```
c = new int[8];
```

// DECLARA e INSTANCIA o vetor

```
int [] c = new int[8];
```

c [0]	
c [1]	
c [2]	
c [3]	
c [4]	
c [5]	
c [6]	
c [7]	

Declaração

vetor

- Quando somente UMA variável do tipo vetor for declarada, o colchete pode ser colocado antes ou depois

```
int [ ] c = new int[8];
```

ou

```
int c[ ] = new int[8];
```

- Quando MAIS de UMA variável

```
int [ ] b = new int[10], x = new int [5];
```

c[0]	
c[1]	
c[2]	
c[3]	
c[4]	
c[5]	
c[6]	
c[7]	

- **Lembrando:** o vetor é um objeto, portanto na sua criação os seus valores são inicializados pela JVM com valores de acordo com o seu tipo.

```
int [ ] c = new int[8];
```

Qual o valores no vetor c,
após a execução da linha anterior?

E seu o tipo ao invés de int fosse: float, char,
objeto, boolean?

c [0]	??
c [1]	??
c [2]	??
c [3]	??
c [4]	??
c [5]	??
c [6]	??
c [7]	??

Manipulação dos elementos do vetor

- Os elementos do vetor são **acessados ou alterados** por meio do índice que indica a posição do elemento que deseja manipular.

```
int [ ] c = new int[8];
```

- Para deixar o vetor conforme a fig.

```
c[0] = -4;
```

```
c[1] = 6;
```

```
.....
```

```
c[7] = 3;
```

c [0]	-4
c [1]	6
c [2]	0
c [3]	72
c [4]	64
c [5]	53
c [6]	78
c [7]	3

Como faz para mostrar o elemento que esta na 3 posição? Qual o seu índice?



Inicialização

vetor

- Pode criar um vetor e inicializar seus elementos com uma declaração de inicialização de vetor:

```
int c [ ] = { -4 , 6 , 0 , 72 , 64, 53 , 78, 3 };
```

- A operação apropriada do new ocorre nos “bastidores” pela JVM.
 - `c = new int[8];`
 - `c [0] = -4;`
 - `c [1] = 6;`
 -

c [0]	-4
c [1]	6
c [2]	0
c [3]	72
c [4]	64
c [5]	53
c [6]	78
c [7]	3



Manipulação

vetor

- Para iterar sobre o vetor utilizamos a estrutura: **for**

```
for ( int i = 0; i < c.length ; i++)  
    System.out.printf("%d %d", i , c[i]);
```

```
// somar todos os elementos do vetor
```

```
int soma = 0;  
for ( int i = 0; i < c.length ; i++)  
    soma = soma + c[ i ]); // soma += c [ i ]
```

c [0]	-4
c [1]	6
c [2]	0
c [3]	72
c [4]	64
c [5]	53
c [6]	78
c [7]	3



Mostrando os dados de maneira tabulas

// mostrando o cabeçalho Índice e Valor

```
System.out.printf("%7s%8d", "Índice", "Valor");
```

// mostrando o valor do índice e o conteúdo

// índice na letra i de e valor debaixo da letra a

```
for ( int i = 0; i < c.length ; i++)
```

```
    System.out.printf("%5d%8d", i , c[i]);
```

Índice	Valor
0	3
1	5
2	1
3	8
4	9
5	0



A estrutura **for** aprimorada

vetor

- A estrutura **for** aprimorada itera pelos elementos sem usar contador.

```
for ( parâmetro: nomeDoVetor)  
    instrução
```

```
for ( int numero : c )  
    System.out.printf(“%d”, numero);
```

```
for ( int i = 0; i < c.length ; i++) // com contador  
    System.out.printf(“%d” , c[i]);
```

c [0]	-4
c [1]	6
c [2]	0
c [3]	72
c [4]	64
c [5]	53
c [6]	78
c [7]	3



A estrutura **for** aprimorada

vetor

- A estrutura **for** aprimorada para somar elementos do vetor

```
for ( int numero : c ) // for aprimorado  
    soma = soma + numero;
```

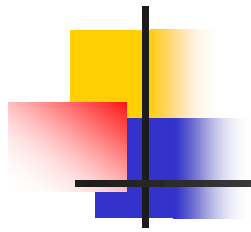
```
for ( int i = 0; i < c.length ; i++) // com contador  
    soma = soma + c[i]);
```



A estrutura for aprimorada

```
for ( int numero : c )  
    soma = soma + numero;
```

- **Simplifica** o código para iterar por um vetor.
- Não pode ser utilizada para **alterar** o elemento, apenas para obter (não tem o índice).
- Utilizada quando não precisamos saber do **índice** do elemento acessado!!



FIM!!!!



Prática de conteúdo

- PráticaConteúdo_01_RevisãoEstruturaControle_Vetor
- Próximas aulas:
 - Revisão de métodos: static, parâmetro e retorno.
 - Geração de números aleatórios. Jogos de dados.