

Vetores

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Vetores

Um vetor corresponde a uma coleção de dados de tamanho fixo, indexada, unidimensional e homogênea

- Indexada: os elementos são acessados por meio de índices
- Unidimensional: uma dimensão
- Homogênea: todos dados são do mesmo tipo

0	Maria
1	João
2	Carlos
3	Ana
4	Joaquim

A

Vetores

Vetores são também chamados de **arranjos** unidimensionais

Em Java a primeira posição de um vetor é a posição 0

Um arranjo deve ser alocado previamente, antes de ser utilizado. Uma vez alocado, sua quantidade de elementos é fixa.

0	Maria
1	João
2	Carlos
3	Ana
4	Joaquim

A

Como criar um vetor?

declaração



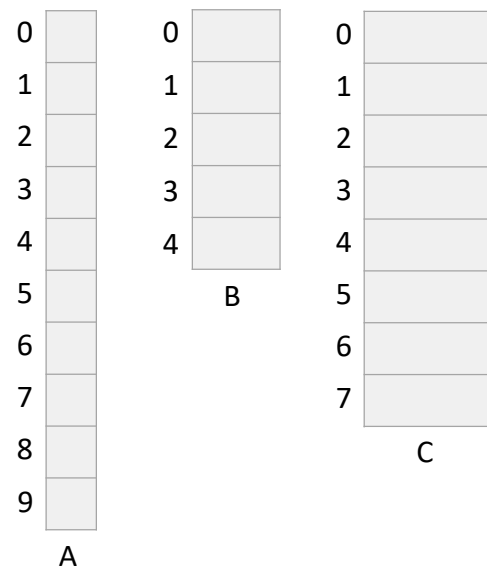
```
int[] A;  
double[] B;  
String[] C;
```

instanciação



```
A = new int[10];  
B = new double[5];  
C = new String[8];
```

Memória RAM



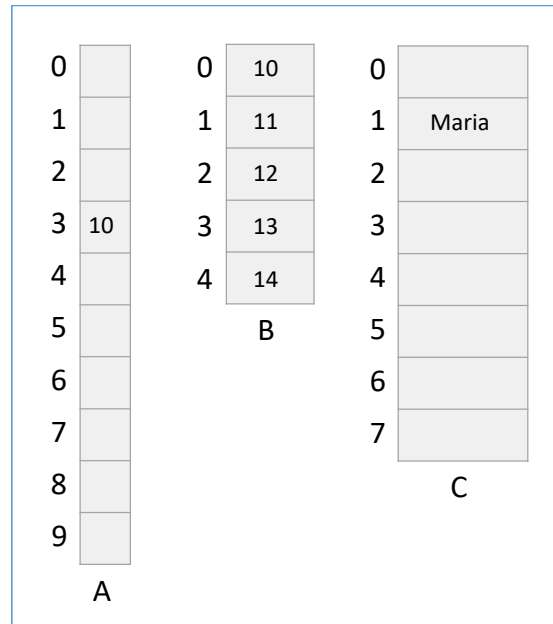
Como acessar os elementos de um vetor?

```
A[3] = 10;
```

```
for (int i=0; i<5; i++) {  
    B[i] = i + 10;  
}
```

```
C[1] = "Maria";
```

Memória RAM



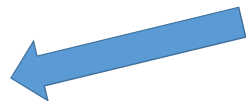
Problema exemplo

Fazer um programa para ler um número inteiro positivo N, depois ler N números quaisquer e armazená-los em um vetor. Em seguida, mostrar na tela todos elementos do vetor.

Entrada	Saída
4	10.5
10.5	4.2
4.2	-7.1
-7.1	15.0
15.0	

0	10.5
1	4.2
2	-7.1
3	15.0

vet



Entrada	Saída
4	10.5
10.5	4.2
4.2	-7.1
-7.1	15.0
15.0	

Resumo da aula

- Vetor: coleção de dados
 - Tamanho fixo
 - Arranjo unidimensional
 - Indexada
 - Homogênea
- Declaração: `double[] B;`
- Instanciação: `B = new double[5];`
- Acesso: `B[3] = 20;`
- Problema exemplo: ler e imprimir na tela um vetor

0	
1	
2	
3	
4	

B

Exercícios propostos PARTE 1: testes de mesa com vetores

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

```
a = 10;  
b = 20;  
c = (a + b) / 2;  
c = c - 40;  
v[4] = a + b + c;
```

0		0		
1		1		a
2		2		
3		3		b
4		4		
5		5		c
6		6		
v		w		d

```
a = 2;  
while (a < 6) {  
    v[a] = 10 * a;  
    a = a + 1;  
}
```

0		0		
1		1		a
2		2		
3		3		b
4		4		
5		5		c
6		6		
v		w		d

```
a = 7;  
b = a - 6;  
while (b < a) {  
    v[b] = b * a;  
    b = b + 2;  
}
```

0		0		
1		1		a
2		2		
3		3		b
4		4		
5		5		c
6		6		
v		w		d

Diagram illustrating the structure of the \mathbb{Z}_2 -module $\mathbb{Z}_2^{\mathbb{Z}_7}$ with three bases v , w , and d . Each basis is represented by a vertical column of 7 boxes, indexed 0 to 6. The shading indicates the value of the basis element at each index:

- Base v :** Boxes 0, 1, 2, 3, 4, 5 are shaded gray; box 6 is white.
- Base w :** Boxes 0, 1, 2, 3, 4 are white; boxes 5, 6 are shaded gray.
- Base d :** Boxes 0, 1, 2 are white; boxes 3, 4, 5, 6 are shaded gray.

0		0		
1		1		a
2		2		
3		3		b
4		4		
5		5		c
6		6		
v	w	d		

Diagram illustrating the state of three registers v , w , and d after the first instruction. Each register is represented by a vertical stack of 7 boxes, indexed 0 to 6. Register v contains 'v' in boxes 0-5 and is empty in box 6. Register w contains 'w' in boxes 0-5 and is empty in box 6. Register d is empty in boxes 0-5 and contains 'd' in box 6.

0		0		10
1		1		a
2		2		20
3		3		b
4	5	4		15 -25
5		5		c
6		6		
v		w		d

0		0		234 56
1		1		a
2	20	2		
3	30	3		b
4	40	4		
5	50	5		c
6		6		
v		w		d

0		0		7
1	7	1		a
2		2		838
3	21	3		7
4		4		b
5	35	5		c
6		6		d

```
for(a=0; a<3; a++) {
    v[a] = 5;
    w[a] = a;
}
```

0	5	0	0	0x2 3
1	5	1	1	a
2	5	2	2	
3		3		b
4		4		
5		5		c
6		6		
v		w		d

```
a = 2;
b = 5;
for (c=0; c<3; c++) {
    v[c] = a;
    w[c] = c * v[c];
}
```

0	2	0	0	2
1	2	1	2	a
2	2	2	4	5
3		3		b
4		4		0x2 3
5		5		c
6		6		
v		w		d

```
v[0] = 2;
for (d=1; d<4; d++) {
    v[d] = v[d-1] * 2;
}
for (d=0; d<4; d++) {
    w[d] = v[d] * 10;
}
```

0	2	0	20	
1	4	1	40	a
2	8	2	80	
3	16	3	160	b
4		4		
5		5		c
6		6		0x2 3
v		w		d