



Vetores



Agenda

- Vetores

Index	0	1	2	3	4
0	65,340	12,483	138,189	902,960	633,877
1	5,246	424,642	650,380	821,254	866,122
2	89,678	236,781	601,691	329,274	913,534
3	103,902	4,567	733,611	263,010	85,550
4	2,778	658,305	128,788	978,155	620,702
5	45,024	55,058	705,586	89,672	384,605
6	780	47,538	523,784	556,801	617,107
7	32,667	350,890	834,753	638,108	85,188
8	56,083	145,582	775,040	548,322	756,587
9	41,123	543,542	537,738	513,048	418,482



Vetores

- Simplifica a manipulação de diversas variáveis de um mesmo tipo de dado
- Armazenam um conjunto de itens que tenham o mesmo tipo de informação
- Os dados são acessados através de uma única variável
- São diferenciados e referenciados por um índice numérico



Vetores

- Estão presentes em praticamente todas as linguagens de programação
- Constituem um dos aspectos mais importantes e facilitadores no desenvolvimento de aplicações

Tipo da variável [] Nome da variável, Nome da variável ;

Ex.:

int [] notas ;

double [] [] vendasPorMes ;



Vetores

Embora a declaração de um vetor seja idêntico ao de qualquer variável, é necessário que seja reservado a quantidade de itens que poderão ser armazenados no vetor antes de sua utilização

Ex.:

```
int [] notas = new int [ 5 ] ;
```

```
double [][] vendasPorMes = new double [3] [12] ;
```



Vetores

- Para referenciarmos um vetor utilizamos o nome da variável, colchetes e uma literal ou variável que representa a posição dentro do vetor
- Esta posição não pode ultrapassar os limites inferior e posterior do vetor

```
int[ ] notas = new int[5];  
notas[0] = 5;
```

```
int vendas = 0;  
int mes = 0;  
double[ ][ ] vendasPorMes = new double[3][12];  
vendasPorMes[vendas][mes] = 345.93;
```



Estruturas de repetição

- Muitas vezes necessitamos de percorrer um vetor para efetuar uma operação
- Poderemos utilizar qualquer uma das três estruturas de repetição para efetuar qualquer operação.
- Normalmente utilizamos uma variável para controlar a posição no vetor
- Esta variável deve ser inicializada e incrementada a fim para que seja possível endereçar todos os itens do vetor



While

A variável “i” representa o ponteiro do vetor

```
int[ ] notas = new int[5];  
  
int i = 0;  
while(i < notas.length) {  
    notas[i] = leInteiro("Informe a ", i+1, "ª nota");  
    i++;  
}
```




Do...While

Será necessário inverter o teste condicional ao se utilizar repita

```
int[ ] notas = new int[5];  
  
int i = 0;  
do {  
    notas[i] = leInteiro("Informe a ", i+1, "ª nota");  
    i++;  
} while(i < notas.length);
```



For

A estrutura de repetição “for” é a mais simples de se utilizar para estas operações

```
int[ ] notas = new int[5];  
  
for(int i = 0; i < notas.length; i++) {  
    notas[i] = leInteiro("Informe a ", i+1, "ª nota");  
}
```



Vetor bidimensional

- Num vetor bidimensional necessitamos de dois índices para referenciarmos um elemento do vetor
- Cada índice endereça uma dimensão no vetor

```
double[ ][ ] vendasPorMes = new double[3][12];

for (int i = 0; i < vendasPorMes.length; i++) {
    for (int j = 0; j < vendasPorMes[i].length; j++) {
        vendasPorMes[i][j] = leReal("Informe o ", i+1,
                                     "° valor de vendas para o ", j+1, "° mês");
    }
}
```