

O que você precisa saber sobre Vetores

Vetores ou Arrays

1. Definição

Vetores em computação não possuem a mesma definição em relação à disciplina de Física. Em física, um vetor é um segmento de reta orientado que possui módulo, direção e sentido.

Na computação, vetores, ou arrays, são estruturas de dados lineares e estáticas. São compostas por um número fixo (finito) de elementos que tem de ser de um mesmo tipo de dado (inteiro, real ou char). Vamos ver alguns exemplos de criação de vetores.

1.1 Criando vetores em pseudocódigo

Vamos a um exemplo criando um vetor de tamanho 10.

```
var  
  numeros: vetor[1..10] de inteiro;
```

O que pode-se dizer da linha acima?

- Foi criada uma variável que é um vetor cujo nome é: numeros.
- Sua dimensão é 10. Vemos isso pela faixa [1..10]. Tem que ser escrito exatamente assim; são apenas 2 pontos na faixa [1..10], começa sempre por 1 e o valor 10 pode ser qualquer valor; indica a dimensão do vetor. Mas o que é dimensão? Dimensão é a quantidade de valores que um vetor poderá ter. Como assim? Quando se declarou um vetor com essa faixa [1..10] podemos visualizar o seguinte esquema ilustrando o vetor:

[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]

Então, dentro de cada célula desta pode ser armazenado um número inteiro. Os índices [1], [2] etc. representam os índices do vetor, que são valores associados a cada célula. Para acessar uma célula basta indexar o nome do vetor ao seu índice. Por exemplo: `numeros[4]`, `numeros[9]` etc.

- O tipo do vetor é inteiro. Isso quer dizer que cada célula só pode armazenar números inteiros.

Observe também que a dimensão de um vetor é de responsabilidade do desenvolvedor. É obrigatório informar uma dimensão e você deve dimensionar uma quantidade que seja o suficiente e necessário para sua lógica.

1.2 Criando vetores em linguagem C

Vamos a um exemplo criando um vetor de tamanho 10.

```
int numeros[10];
```

Pôxa, é bem mais simples não? Vamos agora ao esquema do vetor em C. Agora os índices começam em zero em vez de 1. Veja:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

E, da mesma forma, cada célula dessa pode armazenar um número inteiro. Se por exemplo, em vez de

```
int numeros[10];
```

fosse declarado

```
float numeros[10];
```

Poderia ser colocado em cada célula números reais e não inteiros e, assim por diante. E isso vale também para o pseudocódigo.

1.3 Para que serve um vetor

Veja que variáveis simples assim:

```
int x,y;
```

Só podem armazenar um número inteiro em cada uma delas. Quando for necessário armazenar mais de um valor numa mesma variável, o vetor vai ser útil. Temos vários exercícios exemplos no decorrer das aulas a partir da aula 17, em que vetores são aplicados.

Referências

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de Programação. Editora Pearson Brasil, 2000.

MANZANO, José Augusto Navarro Garcia; OLIVEIRA, Jair Figueiredo de. Algoritmos: Lógica para o desenvolvimento de programação. São Paulo: Érica, 2004.