

O laço for para Arrays: o laço For each

Em toda sua vida de programador Java, será comum você usar o laço *for* para percorrer todos os elementos de um Array.

Se isso é tão utilizado e importante, poderia ter um atalho ou meio mais fácil de fazer isso, não? É, e tem.

Sem contar que, com [Arrays](#) e o [laço for](#) convencional, é bem fácil de se enrolar e cometer erros.

Aprenda uma maneira especial de usar o laço *for* e pare de ter problemas: é laço *foreach*

Fazendo o *for* percorrer todo o Array/Vetor

A utilização é bem simples e direta. Vamos usar essa variante do *for* que percorre sempre, do começo ao fim, todos os elementos de um Array.

É bem útil, também, em termos de precaução e organização, pois alguns programadores não gostam de usar o índice 0, usam direto o índice 1 do array, ou as vezes nos confundimos e passamos (durante as iterações) do limite do array. Isso pode ser facilmente evitado usando o laço *for* modificado para percorrer os elementos de um array.

A sintaxe do **for each** é a seguinte:

```
for ( tipo_variavel_do_tipo_do_seuArray : seuArray){  
    //seu código  
}
```

Se o 'seuArray' for de inteiro, ficaria:

```
for (int count : seuArray){  
    ...  
}
```

Como podemos interpretar esse laço *foreach*?

Muito simples, o inteiro 'count' vai receber, a cada iteração, todos os valores de 'seuArray'.

Ou seja, ele vai percorrer todo o seu Array e receber seus valores, na ordem (do começo para o fim), um por vez. E com esses valores você vai fazer o que desejar.

Por exemplo, suponha que as notas de cada aluno de uma escola é armazenada em um Array.

Como para calcular a média você precisará **sempre** percorrer todo o Array/Vetor, pode, e deve-se, usar esse laço *for* especial. Pois ele foi feito para isso, para facilitar nossa vida.

Programadores não devem perder tempo. Devem fazer as coisas sempre da maneira mais simples e prática possível. Não tente fazer arte ou gambiarra.

Se o próprio Java lhe fornece as opções e ferramentas, use-as.

É uma arma 'especial' que não se vê por aí, em qualquer linguagem de programação.

Se você já programa ou programou em outra linguagem deve estar percebendo como Java é cheio dessas ferramentas e pormenores.

Se estudar tudo direitinho, irá se tornar um eficiente programador, pois o Java realmente foi feito para facilitar a vida de desenvolvedores, é realmente uma linguagem voltada para o trabalho.

Exemplo de uso do *foreach* em Java:

Crie um aplicativo em Java que peça 5 números ao usuário.

Depois, mostre o resultado da soma desses números;

Passo 1:

Vamos declarar o vetor de inteiro e o inteiro 'soma' para receber a soma dos resultados que o usuário fornecer.

Usamos um laço *for*, convencional, para percorrer o array.

Note que usamos 'numero.length', que é uma boa prática.

Passo 2:

Agora temos que percorrer o array inteiro, e já que temos que fazer isso, vamos fazer usando o laço for modificado que percorre um array inteiro recebendo seus valores. Como o inteiro 'cont' vai receber os dados de todos os elementos do array 'numero', vamos somar o valor de 'cont' a cada iteração na variável 'soma'.

O código Java de nossa aplicação ficará:

```
import java.util.Scanner;

public class forParaArray {

    public static void main(String[] args){
        int[] numero = new int[5];
        int soma=0;
        Scanner entrada = new Scanner(System.in);

        for(int cont=0 ; cont< numero.length ; cont++){
            System.out.print("Entre com o número "+(cont+1)+" : ");
            numero[cont]=entrada.nextInt();
        }

        //exibindo e somando
        for(int cont : numero){
            soma += cont;
        }

        System.out.println("A soma dos números que você digitou é "+soma);
    }

}
```