

Aula X - Desenvolvimento de Aplicativos I

Revisão de algoritmos e programação - arrays.

Fontes:

<https://distancia.qi.edu.br/mod/book/view.php?id=18698>

REVISÃO DE CONCEITOS BÁSICOS

Arrays

Um array (em português: arranjo, vetor) é um tipo especial de objeto que contém zero ou mais valores primitivos ou referências. É como uma variável que pode armazenar mais de um valor (um conjunto de valores). Esses valores são mantidos nos elementos do array, que são variáveis não identificadas às quais fazemos referência pela sua posição ou índice. O tipo de um array é definido pelo seu elemento, e todos os elementos do array devem ser desse tipo.

Os elementos do array são numerados iniciando com zero, e índices válidos variam de zero ao número de elementos menos um. O elemento do array com índice 1, por exemplo, é o segundo elemento no array. O número de elementos em um array é seu comprimento. É especificado quando o array é criado e nunca muda seu tamanho.

Criando e inicializando arrays

Para criar um array em Java, você utiliza a palavra-chave new, assim como faz para criar um novo objeto. Os tipos array não possuem construtores, mas é exigido que você especifique um tamanho sempre que criar um array. O tamanho desejado do array deve ser especificado dentro de colchetes.

```
int[] jogoMegaSena; //Declaração
jogoMegaSena = new int[6]; //Criação
jogoMegaSena[0] = 23; //Inicialização da posição 0
jogoMegaSena[1] = 12; //Inicialização da posição 1
jogoMegaSena[2] = 55; //Inicialização da posição 2
jogoMegaSena[3] = 02; //Inicialização da posição 3
jogoMegaSena[4] = 07; //Inicialização da posição 4
jogoMegaSena[5] = 19; //Inicialização da posição 5
```

IMPORTANTE: O OPERADOR `new` ALOCA (RESERVA/CRIA NA MEMÓRIA RAM DO COMPUTADOR) UM LOCAL ONDE SERÁ ARMAZENADO O VETOR. ISTO É FEITO DINAMICAMENTE, OU SEJA, EM TEMPO DE EXECUÇÃO DO PROGRAMA (QUANDO ELE ESTÁ RODANDO).

Outra forma de inicialização de um *array*, mais enxuta:

```
int[] outroJogoSena = new int[]{23, 12, 55, 02, 07, 19};
```

No exemplo acima, acontecem quatro coisas em apenas uma linha:

1. Declaração de uma referência a um *array* de inteiros chamado `outroJogoSena`;
2. Criação de um *array* com seis posições;
3. Inicialização das posições com os valores 23, 12, 55, 02, 07 e 19;
4. Atribuição do novo objeto (*array*) a `outroJogoSena` como uma referência.

Outro atalho que a linguagem Java nos permite é o seguinte:

```
int[] outrojogoMegaSena = {23, 12, 55, 02, 07, 19};
```

Acessando elementos do array

```
String[] respostas = new String[2]; //Cria um array de dois Strings
respostas[0] = "Sim"; //configura o primeiro elemento do array
respostas[1] = "Não"; //configura o segundo elemento do array
//Agora lê esses elementos do array
System.out.println(questão + " ("
+ resposta[0] + "/" + resposta[1] + "): ");
Limites de array
```

Lembre-se de que o primeiro elemento de um *array* `a` é `a[0]`. O segundo é `a[1]` e o último é `a[a.length-1]`.

Iterando por arrays

É comum escrever laços que iteram (percorrem) por cada um dos elementos de um *array*, afim de realizar uma operação sobre ele:

```
int[] primos = {2,3,5,7,11,13,17,19};
```

```
int somaPrimos = 0;
```

```
for(int i = 0; i < primos.length; i++)  
    somaPrimos += primos[i];
```

Outra forma de fazer iteração de *Arrays* é utilizando o `for`. Esta forma de percorrer o vetor é mais rápida e torna o programa mais claro:

```
int[] vetor = {12,23,45,76,98};  
  
for (int x: vetor) { //Para cada int x de vetor faça...  
    System.out.println(x); //Aqui imprimimos o x, que recebe um int do  
    //vetor a cada repetição do for.  
}
```

Utilitários de arrays

A classe `java.util.Arrays` contém alguns métodos utilitários estáticos para trabalhar com *arrays*, por exemplo:

- `sort()` - classificar *arrays*.
- `binarySearch()` – busca em *arrays*.
- `equals()` – comparar *arrays*.
- `Arrays.toString()` – converte o conteúdo do *Array* em um string.

Exercícios de fixação do conteúdo - para quem quiser fazer - estudo

- 1) Implemente a lógica para realização de saques em um caixa eletrônico, considerando que o mesmo armazena cédulas de R\$100,00, R\$50,00, R\$20,00, R\$10,00, R\$5,00 e R\$2,00 e devem ser entregues ao cliente o menor número possível de cédulas.
- 2) Escreva um programa em Java para ler um vetor X de 10 elementos inteiros. Logo após copie os elementos do vetor X para um vetor Y fazendo com que o 1º elemento de X seja copiado para o 10º de Y, o 2º de X para o 9º de Y e assim sucessivamente (inverter o vetor X no vetor Y). Após o término da cópia, imprimir o vetor Y.
- 3) Escreva um programa em Java para ler um vetor A de 10 elementos inteiros e um valor X. A seguir imprimir os índices do vetor A em que aparece um valor igual a X.
- 4) Escreva um programa em Java para ler um vetor A de 10 elementos inteiros e um valor X. A seguir imprimir "ACHEI" se o valor X existir em A e "NÃO ACHEI" caso contrário (buscar o X).
- 5) Escreva um programa em Java para ler um vetor A de 10 elementos e um valor X. Copie para um vetor S (sem deixar elementos vazios entre os valores copiados) os elementos de A que são maiores que X (buscar os maiores que X). Logo após imprimir o vetor S.
- 6) Escreva um programa em Java para ler a quantidade de elementos que serão armazenados em um vetor A. Crie um vetor com o tamanho apropriado. Depois o programa deve solicitar os números conforme a quantidade informada pelo usuário, e armazená-los no vetor A. Logo após imprimir o vetor A.

Link do vídeo da correção dos exercícios:

<https://www.youtube.com/watch?v=iSiMcZi9Cic>