Arranjos

Emerson C. Lima

Programação Java

Objetivos dessa lição

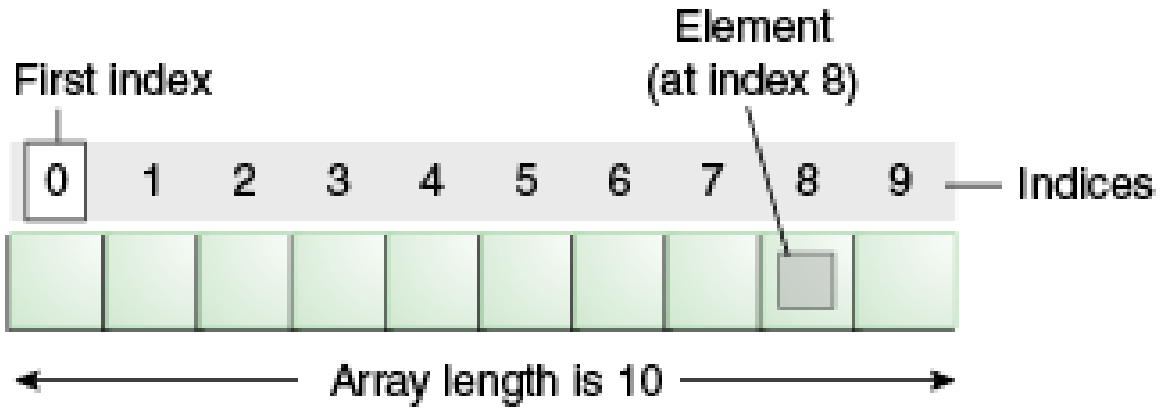
- Saber declarar, inicializar e acessar elementos de arranjos
- Saber realizar cópia de dados entre arranjos de maneira eficiente
- Conhecer as operações de arranjos disponibilizadas na biblioteca padrão do Java

Conteúdo

- Arranjos
- Declarando arranjos
- Criando arranjos
- Acessando elementos de arranjos
- Arranjos de arranjos
- Copiando dados entre arranjos
- Operações com arranjos

Arranjos

- Um arranjo é um objeto contâiner que armazena um número fixo de valores de um mesmo tipo.



Declarando arranjos

```java
// declara um arranjo de inteiros
int[] umArranjo;
```

```java
byte[] umArranjoDeBytes ;
short[] umArranjoDeShorts ;
long[] umArranjoDeLongs ;
float[] umArranjoDeFloats ;
double[] umArranjoDeDoubles ;
boolean[] umArranjoDeBooleans ;
char[] umArranjoDeChars ;
String[] umArranjoDeStrings ;
```

Criando um arranjo

```
1 // cria um arranjo de inteiros
2 umArranjo = new int[10];
```

```
1 umArranjo[0] = 100; // atribui 100 como o primeiro elemento
2 umArranjo[1] = 200; // atribui o segundo elemento
3 umArranjo[2] = 300; // e assim por diante
```

Criando arranjos

- Declarando e inicializando um arranjo

```
1 int[] umArranjo = {
2     100, 200, 300,
3     400, 500, 600,
4     700, 800, 900, 1000
5 };
```

Aqui o comprimento do arranjo é o determinado pelo número de valores fornecidos entre as chaves e separados por vírgula.

Acessando elementos de arranjos

- Declarando e inicializando um arranjo

```
1 System.out.println("Elemento 1 no índice 0: "  + umArranjo[0]);
2 System.out.println("Elemento 2 no índice 1: "  + umArranjo[1]);
3 System.out.println("Elemento 3 no índice 2: "  + umArranjo[2]);
```

Arranjo de arranjos

- Arranjos de arranjos (arranjos multidimensionais) podem ser declarados usando dois ou mais conjuntos de colchetes, como nomes String[][].

```
1  class MultiDimArrayDemo  {
2      public static void main(String[] args) {
3          String[][] names = {
4              {"Mr. ", "Mrs. ", "Ms. "},
5              {"Smith", "Jones"}
6          };
7          // Mr. Smith
8          System.out.println(names[0][0] + names[1][0]);
9          // Ms. Jones
10         System.out.println(names[0][2] + names[1][1]);
11     }
12 }
```

O tamanho de arranjos

- Usando a propriedade length

```
1 System.out.println(umArranjo.length);
```

Copiando dados entre arranjos

- A classe System possui um método arraycopy que você pode utilizar para eficientemente copiar dados de um arranjo para outro:

```
1 public static void arraycopy(Object src, int srcPos,
2                               Object dest, int destPos, int length)
```

Copiando dados entre arranjos

- A classe System possui um método arraycopy que você pode utilizar para eficientemente copiar dados de um arranjo para outro:

```
1  class ArrayCopyDemo {
2      public static void main(String[] args) {
3          char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e',
4                  'i', 'n', 'a', 't', 'e', 'd' };
5          char[] copyTo = new char[7];
6
7          System.arraycopy(copyFrom, 2, copyTo, 0, 7);
8          System.out.println(new String(copyTo));
9      }
10 }
```

Operações com arranjos

- O Java oferece diversos métodos para realizar manipulação de arranjos (tarefas comuns, copia, ordenação e busca na classe java.util.Arrays.

```
1  class ArrayCopyOfDemo {
2      public static void main(String[] args) {
3
4          char[] copyFrom = {'d', 'e', 'c', 'a', 'f', 'f', 'e',
5              'i', 'n', 'a', 't', 'e', 'd'};
6
7          char[] copyTo = java.util.Arrays.copyOfRange(copyFrom, 2, 9);
8
9          System.out.println(new String(copyTo));
10     }
11 }
```