



INSTITUTO FEDERAL
Sudeste de Minas Gerais



Arrays e ArrayLists

Curso: Técnico em Informática Integrado

Prof.: Jean Henrique de Sousa Câmara

Contato: jean.camara@ifsudestemg.edu.br

Arrays

2

- Um array é um grupo de variáveis (chamados elementos ou componentes) que contém valores todos do mesmo tipo
- Para referenciar um elemento particular em um array, especificamos o nome da referência para o array e o número de posição do elemento no array
 - O número de posição do elemento é chamado de índice ou subscrito

Arrays

3

- Um índice deve ser um **inteiro não negativo**

Nome do array (c) →	c[0]	-45
	c[1]	6
	c[2]	0
	c[3]	72
	c[4]	1543
	c[5]	-89
	c[6]	0
	c[7]	62
	c[8]	-3
	c[9]	1
Índice (ou subscripto) do elemento no array c →	c[10]	6453
	c[11]	78

- A expressão **c.length** retorna o comprimento do array c

Declarando e criando arrays

4

- Como outros objetos, arrays são criados com a palavra-chave `new`
- Especifique o tipo dos elementos do array e o número de elementos

```
int[] c = new int[12];
```

- Quando um array é criado, cada um de seus elementos recebe um valor padrão
 - zero para os elementos de tipo primitivo numéricos
 - false para elementos boolean
 - null para referências

Exemplos que utilizam arrays

5

```
1 public class IniciaArray {  
2  
3     Run | Debug  
4     public static void main(String[] args) {  
5         // declara array variável e o inicializa com um objeto array  
6         int[] c = new int[10]; // cria o objeto array  
7  
8         System.out.printf(format: "%s%8s%n", ...args: "Índice", "Valor"); // títulos de coluna  
9  
10        // gera saída do valor de cada elemento do array  
11        for (int i = 0; i < c.length; i++)  
12            System.out.printf(format: "%5d%8d%n", i, c[i]);  
13    }
```

Exemplos que utilizam arrays

6

- Outra forma de criar e iniciar um array

```
1 public class IniciaArray {  
2  
3     Run | Debug  
4     public static void main(String[] args) {  
5         // declara array variável e o inicializa com um objeto array  
6         int[] c = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };  
7  
8         System.out.printf(format: "%s%8s%n", ...args: "Índice", "Valor"); // títulos de coluna  
9  
10        // gera saída do valor de cada elemento do array  
11        for (int i = 0; i < c.length; i++)  
12            System.out.printf(format: "%5d%8d%n", i, c[i]);  
13    }
```

Exemplos que utilizam arrays

7

```
1  public class SomaArray {  
    Run | Debug  
2      public static void main(String[] args) {  
3          int[] c = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };  
4  
5          int total = 0;  
6  
7          // adiciona o valor de cada elemento ao total  
8          for (int i = 0; i < c.length; i++)  
9              total += c[i];  
10  
11         System.out.printf(format: "Soma Total: %d\n", total);  
12     }  
13 }
```

Exemplos que utilizam arrays

8

```
1  public class GraficoBarras {  
    Run | Debug  
2  public static void main(String[] args) {  
3      int[] numeros = { 0, 0, 0, 0, 0, 0, 1, 2, 4, 2, 1 };  
4  
5      System.out.println(x: "Distribuição de Notas:");  
6  
7      // para cada elemento de array, gera saída de uma barra do gráfico  
8      for (int i = 0; i < numeros.length; i++){  
9          // gera saída do rótulo de barra ( "00-09: ", ..., "90-99: ", "100: ")  
10         if (i == 10)  
11             System.out.printf(format: "%5d: ", ...args: 100);  
12         else  
13             System.out.printf(format: "%02d-%02d: ", i*10, i*10 + 9);  
14  
15         // imprime a barra de asteriscos  
16         for (int estrelas = 0; estrelas < numeros[i]; estrelas++)  
17             System.out.print(s: "*");  
18  
19         System.out.println();  
20     }  
21 }  
22 }
```


Exemplos que utilizam arrays

9

```
1  import java.security.SecureRandom;
2
3  public class JogarDado {
4      Run | Debug
5      public static void main(String[] args) {
6          SecureRandom numeroAleatorio = new SecureRandom();
7
8          int[] frequencias = new int[7]; // array de contadores de frequência
9
10         // lança o dado 6.000.000 vezes; usa o valor do dado como índice de frequência
11         for (int jogada = 1; jogada <= 6000000; jogada++)
12             ++frequencias[1 + numeroAleatorio.nextInt(bound: 6)];
13
14         System.out.printf(format: "%s\t%s%n", ...args: "Face", "Frequência");
15         // gera saída do valor de cada elemento do array
16         for (int face = 1; face < frequencias.length; face++)
17             System.out.printf(format: "%d\t%d%n", face, frequencias[face]);
18     }
19 }
```

ArrayList

10

- Arrays não mudam automaticamente o tamanho em tempo de execução para acomodar elementos adicionais
- A classe de coleção **ArrayList<T>** (pacote java.util) pode alterar dinamicamente seu tamanho para acomodar mais elementos
 - ao declarar um novo ArrayList, substitua o T pelo tipo dos elementos que você deseja que o ArrayList armazene

```
ArrayList<String> list;
```

```
ArrayList<Integer> integers;
```

ArrayList (Alguns Métodos)

11

Método	Descrição
<code>add</code>	Adiciona um elemento ao <i>final</i> do <code>ArrayList</code> .
<code>clear</code>	Remove todos os elementos do <code>ArrayList</code> .
<code>contains</code>	Retorna <code>true</code> se o <code>ArrayList</code> contém o elemento especificado; caso contrário, retorna <code>false</code> .
<code>get</code>	Retorna o elemento no índice especificado.
<code>indexOf</code>	Retorna o índice da primeira ocorrência do elemento especificado no <code>ArrayList</code> .
<code>remove</code>	Sobrecarregado. Remove a primeira ocorrência do valor especificado ou o elemento no índice especificado.
<code>Size</code>	Retorna o número de elementos armazenados em <code>ArrayList</code> .
<code>trimToSize</code>	Corta a capacidade do <code>ArrayList</code> para o número atual de elementos.

Exemplo que utiliza ArrayList

12

```
1  import java.util.ArrayList;
2
3  public class ColecaoArrayList {
4      Run | Debug
5      public static void main(String[] args) {
6          // cria um novo ArrayList de strings com uma capacidade inicial de 10
7          ArrayList<String> cores = new ArrayList<String>();
8
9          cores.add(e: "vermelho"); // anexa um item à lista
10         cores.add(index: 0, element: "amarelo"); // insere "amarelo" no índice 0
11
12         // cabeçalho
13         System.out.println(x: "Exibe conteúdo da lista");
14
15         // exibe as cores na lista
16         for (int i = 0; i < cores.size(); i++)
17             System.out.printf(format: "%s ", cores.get(i));
18     }
19 }
```

Exemplo que utiliza ArrayList

13

```
14      // exibe as cores na lista
15      for (int i = 0; i < cores.size(); i++)
16          System.out.printf(format: "%s ", cores.get(i));
17
18      cores.add(e: "verde"); // adiciona "verde" ao fim da lista
19      cores.add(e: "amarelo"); // adiciona "amarelo" ao fim da lista
20
21      System.out.println(x: "\n\nExibe conteúdo da lista");
22      for (int i = 0; i < cores.size(); i++)
23          System.out.printf(format: "%s ", cores.get(i));
24  }
25 }
```

Exemplo que utiliza ArrayList

14

```
18 cores.add(e: "verde"); // adiciona "verde" ao fim da lista
19 cores.add(e: "amarelo"); // adiciona "amarelo" ao fim da lista
20
21 System.out.println(x: "\n\nExibe conteúdo da lista");
22 for (int i = 0; i < cores.size(); i++)
23     System.out.printf(format: "%s ", cores.get(i));
24
25 cores.remove(o: "amarelo"); // remove o primeiro "amarelo"
26
27 System.out.println(x: "\n\nExibe conteúdo da lista");
28 for (int i = 0; i < cores.size(); i++)
29     System.out.printf(format: "%s ", cores.get(i));
30
31 cores.remove(index: 1); // remove o item no índice 1
32
33 System.out.println(x: "\n\nExibe conteúdo da lista");
34 for (int i = 0; i < cores.size(); i++)
35     System.out.printf(format: "%s ", cores.get(i));
```

Exemplo que utiliza ArrayList

15

```
33      System.out.println(x: "\n\nExibe conteúdo da lista");
34      for (int i = 0; i < cores.size(); i++)
35          System.out.printf(format: "%s ", cores.get(i));
36
37      // verifica se um valor está na lista
38      System.out.printf(format: "\n\nVermelho %sestá na lista\n",
39          cores.contains(o: "vermelho") ? "" : "não ");
40
41      // exibe o número de elementos na lista
42      System.out.printf(format: "Tamanho: %s\n", cores.size());
43  }
44  }
```

Exercícios (3 pontos)

16

1. Ler um vetor de 10 elementos. Crie um segundo vetor, com todos os elementos na ordem inversa, ou seja, o último elemento passará a ser o primeiro, o penúltimo será o segundo e assim por diante. Imprima os dois vetores.
2. Ler um vetor com 10 nomes de pessoas, após pedir que o usuário digite um nome qualquer de pessoa. Escrever a mensagem “ACHEI”, se o nome estiver armazenado no vetor C ou “NÃO ACHEI” caso contrário.

Exercícios (3 pontos)

17

3. Fazer um programa para ler 5 valores e, em seguida, mostrar todos os valores lidos juntamente com o maior, o menor e a media dos valores.
4. Faça um programa que leia dois vetores de 10 elementos. Crie um vetor que seja a interseção entre os 2 vetores anteriores, ou seja, que contém apenas os números que estão em ambos os vetores. Não deve conter números repetidos.

Dúvidas?

18



jean.camara@ifsudestemg.edu.br