

Arrays e ArrayLists

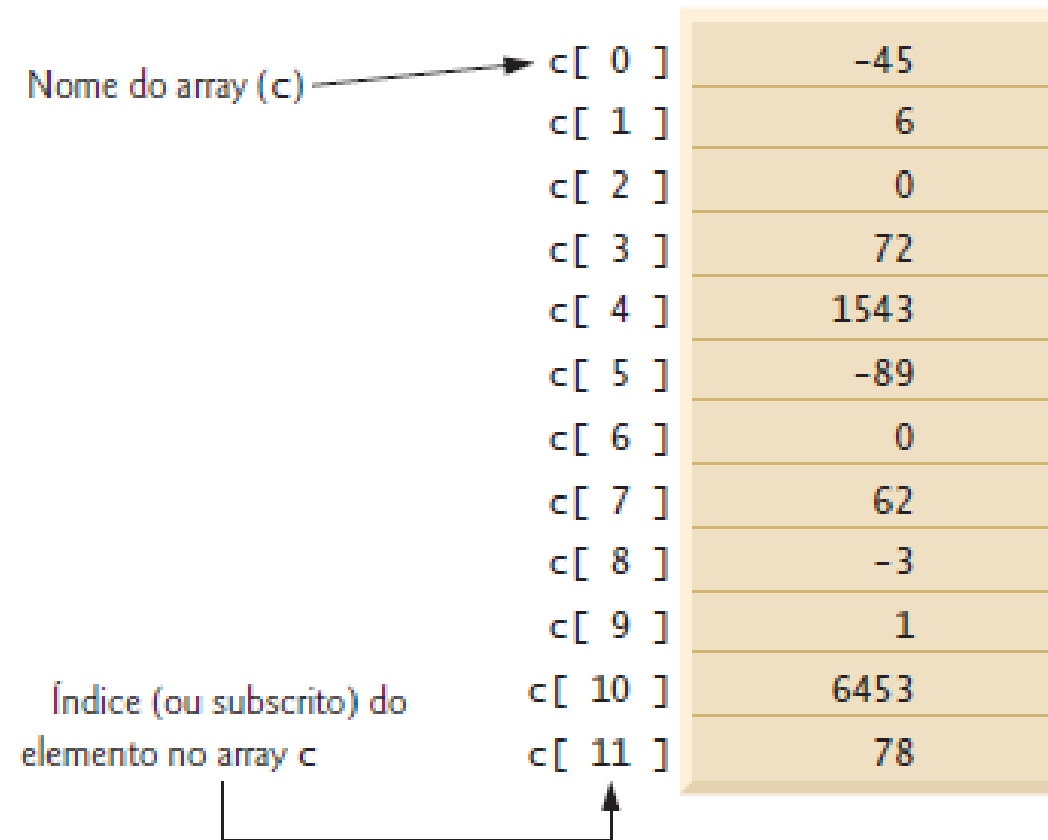
Prof. Ricardo P. Mesquita

Arrays

- Um array é um grupo de que contém valores todos do ***mesmo tipo***.
- Os arrays são ***objetos***, portanto, são considerados ***tipos por referência***.
- O que consideramos um array é, na verdade, uma ***referência*** a um ***objeto array*** na memória.
- Os ***elementos*** de um array podem ser ***tipos primitivos*** ou ***tipos por referência***.
- 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

- Visão lógica:



Arrays

- Exemplo simples:

```
public class IniciarArray {  
    public static void main( String[] args )  
    {  
        int[] arranjo; // declara um array chamado "arranjo"  
  
        arranjo = new int[ 10 ]; // cria o objeto array  
  
        System.out.printf( "%s%8s\n", "índice", "Valor" ); // nomes das colunas  
  
        // mostrando os elementos do array...  
        for ( int contador = 0; contador < arranjo.length; contador++ )  
            System.out.printf( "%5d%8d\n", contador, arranjo[ contador ] );  
    }  
}
```

Arrays

eclipse-workspace - IniciarArray/src/IniciarArray.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Aula3_grafico
- Aula3_POO
- Aula4_Craps
- Aula4_Escopo
- Aula4_POO
- Aula4_Smile
- DesAplicDist_1
- IniciarArray
 - JRE System Library [JavaSE-1.8]
 - src
 - (default package)
 - IniciarArray.java

Boletim.java BoletimTest... Janela.java DrawSmileyT... *IniciarArr... »

```
1 public class IniciarArray {
2     public static void main( String[] args )
3     {
4         int[] arranjo; // declara um array
5
6         arranjo = new int[ 10 ]; // cria o array
7
8         System.out.printf( "%s%s\n", "Iniciando array", arranjo );
9
10        // mostrando os elementos do array
11        for ( int contador = 0; contador < arranjo.length; contador++ )
12            System.out.printf( "%5d%8d\n", contador, arranjo[contador] );
13    }
14 }
15
16
17
```

<terminated> IniciarArray

índice	Valor
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0

Task List

Find

Outline

IniciarArray

Problems @ Javadoc Declaration Console

<terminated> IniciarArray [Java Application] C:\Program Files\Java\jdk-1.8.0_101\bin\java.exe

índice	Valor
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0

Ativar o Windows

Ative o Windows para obter o melhor desempenho e recursos. Clique aqui para obter mais informações.

Writable Smart Insert 17:1

Windows Taskbar

Windows Taskbar

Arrays

- Usando inicializador:

```
public class IniciarArray2 {  
    public static void main( String[] args )  
    {  
        int[] arranjo = { 35, 56, 87, 23, -12, 0, 456, 90, -24, 65 };  
  
        System.out.printf( "%s%8s\n", "indice", "Valor" ); // nomes das colunas  
  
        // mostrando os elementos do array...  
        for ( int contador = 0; contador < arranjo.length; contador++ )  
            System.out.printf( "%5d%8d\n", contador, arranjo[ contador ] );  
    }  
}
```

Arrays

The screenshot shows the Eclipse IDE with a Java project named 'IniciarArray2'. The main editor displays the following code:

```
1 public class IniciarArray2 {
2     public static void main( String[] args)
3     {
4         int[] arranjo = { 35, 56, 87, 23, -12, 0, 456, 90, -24, 65 };
5         System.out.printf( "%s\n", arranjo );
6
7         // mostrando os elementos do array
8         for ( int contador = 0; contador < arranjo.length; contador++)
9             System.out.printf( "%5d\n", arranjo[contador] );
10    }
11 }
```

The console output, titled '<terminated> IniciarArray2', shows the array elements printed in two columns:

índice	Valor
0	35
1	56
2	87
3	23
4	-12
5	0
6	456
7	90
8	-24
9	65

A red arrow points from the array declaration in the code to the console output. The Package Explorer on the left shows the project structure, and the Outline view on the right shows the class and method structure.

Arrays

- Usando constante:

```
public class IniciarArray3 {  
    public static void main( String[] args )  
    {  
        final int COMPRIMENTO_DO_ARRAY = 10; // declara constante  
        int[] arranjo = new int[COMPRIMENTO_DO_ARRAY];  
  
        for ( int contador = 0; contador < arranjo.length; contador++ )  
            arranjo[ contador ] = 2 + 2 * contador;  
  
        System.out.printf( "%s%8s\n", "índice", "Valor" ); // nomes das colunas  
  
        for ( int contador = 0; contador < arranjo.length; contador++ )  
            System.out.printf( "%5d%8d\n", contador, arranjo[ contador ] );  
    }  
}
```


Arrays

eclipse-workspace - IniciarArray3/src/IniciarArray3.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- > Aula3_grafico
- > Aula3_POO
- > Aula4_Craps
- > Aula4_Escopo
- > Aula4_POO
- > Aula4_Smile
- > DesAplicDist_1
- > IniciarArray
- > IniciarArray2
- ▼ IniciarArray3
 - > JRE System Library [JavaSE-1.8]
 - ▼ src
 - (default package)
 - IniciarArray3.java

Boletim.java DrawSmileyT... *IniciarArr...

```
1 public class IniciarArray3 {
2     public static void main( String[] args )
3     {
4         final int COMPRIMENTO_DO_ARRAY = 10; // de
5         int[] arranjo = new int[COMPRIMENTO_DO_ARR
6
7         for ( int contador = 0; contador < arranjo
8             arranjo[ contador ] = 2 + 2 * contad
9
10        System.out.printf( "%s%8s\n", "indice", "
11
12        for ( int contador = 0; contador < arranjo
13            System.out.printf( "%5d%8d\n", conta
14
15    }
16 }
17
18
```

<terminated> IniciarArray3

índice	Valor
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16
8	18
9	20

Task List

Find

Outline

- ▼ IniciarArray3
 - main(String[]) : void

Problems @ Javadoc Declaration Console

<terminated> IniciarArray3 [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (10 de mar de 2019 13:29:48)

índice	Valor
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16
8	18
9	20

Ativar o Windows

Acesse Configurações para ativar o Windows.

Writable Smart Insert 18 : 1

Windows Search

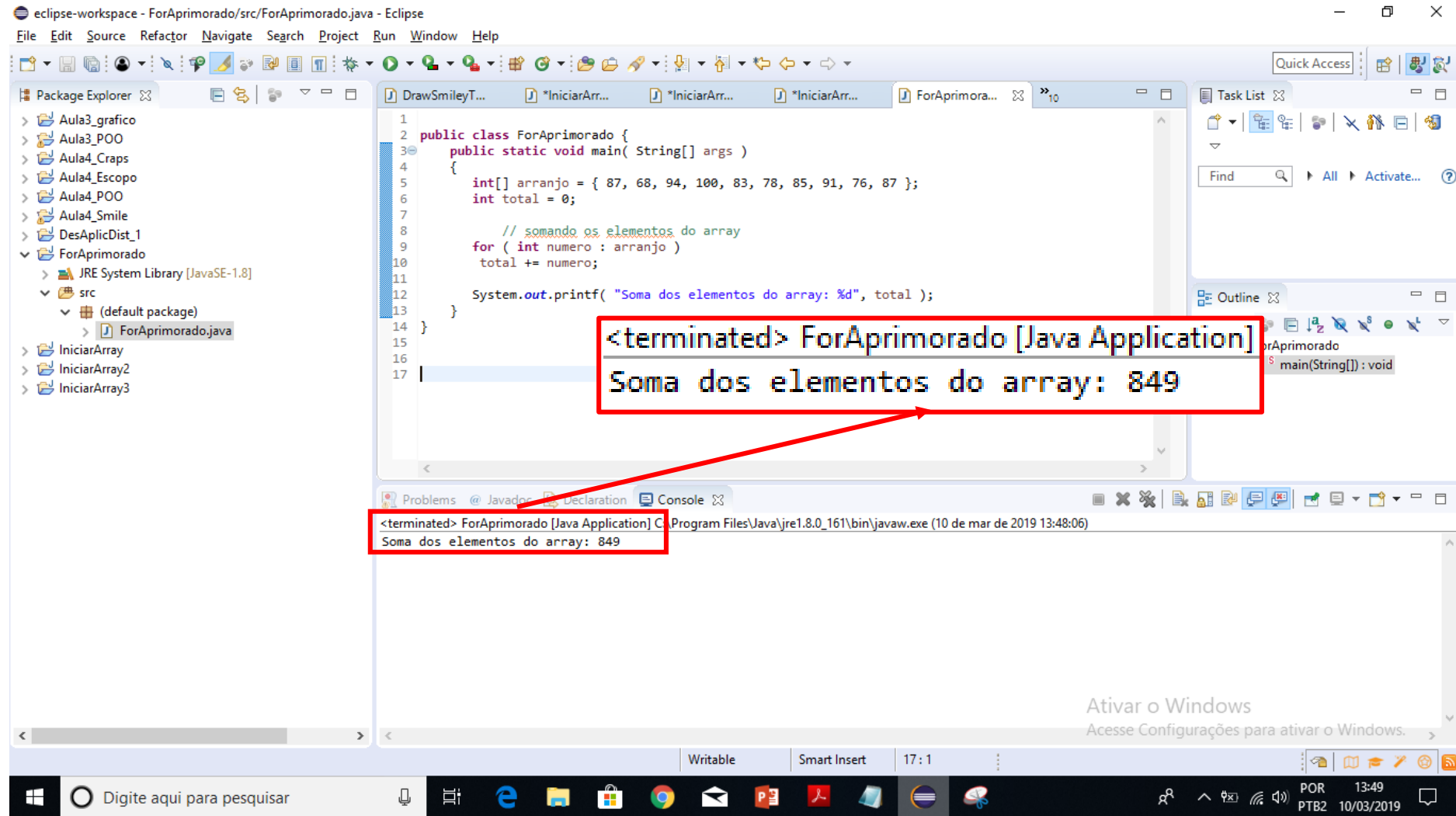
13:32 10/03/2019

Arrays

- Usando **for** aprimorado:

```
public class ForAprimorado {  
    public static void main( String[] args )  
    {  
        int[] arranjo = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };  
        int total = 0;  
  
        // somando os elementos do array  
        for ( int numero : arranjo )  
            total += numero;  
  
        System.out.printf( "Soma dos elementos do array: %d", total );  
    }  
}
```

Arrays



Arrays

- Passando arrays e elementos do arrays individuais aos métodos.
- Vamos escrever um programa com dois métodos (além do main):
 - Um que recebe o array inteiro como referência, e
 - Um outro que recebe apenas um elemento do array (passagem por valor).

Arrays

```
public class PasssandoArray {  
    public static void main( String[] args )  
    {  
        int[] arranjo = { 1, 2, 3, 4, 5 };  
  
        System.out.println(  
            "Efeitos de se passar por referência todo o array:\n" +  
            "Os valores originais do array são: " );  
  
        for ( int valor : arranjo )  
            System.out.printf( "    %d", valor );  
  
        modificandoArray( arranjo ); // array passado por referência!  
        System.out.println( "\n\nOs valores do array modificado são: " );  
  
        for ( int valor : arranjo )  
            System.out.printf( "    %d", valor );  
  
        System.out.printf(  
            "\n\nEfeitos da passagem de um elemento do array:\n" +  
            "arranjo[3] antes de chamar modificarElemento(): %d\n", arranjo[ 3 ] );  
  
        modificarElemento( arranjo[ 3 ] ); // tentativa de modificar arranjo[ 3 ]  
        System.out.printf(  
            "arranjo[3] depois da chamada a modificarElemento(): %d\n", arranjo[ 3 ] );  
    }  
}
```

Chamadas
aos métodos
da classe

Arrays

Acrescente os métodos:

```
public static void modificandoArray( int algum_array[] )
{
    for ( int contador = 0; contador < algum_array.length; contador++ )
        algum_array[ contador ] *= 2; // multiplica os elementos por 2
}

public static void modificarElemento( int elemento )
{
    elemento *= 3; // multiplica o elemento por 3
    System.out.printf(
        "Valor do elemento em modificarElemento(): %d\n", elemento );
}
}
```

Arrays

The screenshot shows the Eclipse IDE with a Java project named 'PassandoArray'. The source code in 'PassandoArray.java' includes a main method 'IniciarArray' and a static method 'modificandoArray'. The console output, which is highlighted with a red box, shows the results of running the program. It displays the original array values (1, 2, 3, 4, 5), the modified array values (2, 4, 6, 8, 10), and the effects of passing an array element by reference, specifically showing that 'arranjo[3]' changes from 8 to 24 when the 'modificarElemento()' method is called. A red arrow points from the console output to the corresponding code line in the source file.

```
<terminated> PassandoArray [Java Application] C:\Program Files\Java\j
Efeitos de se passar por referência todo o array:
Os valores originais do array são:
1 2 3 4 5

Os valores do array modificado são:
2 4 6 8 10

Efeitos da passagem de um elemento do array:
arranjo[3] antes de chamar modificarElemento(): 8
Valor do elemento em modificarElemento(): 24
arranjo[3] depois da chamada a modificarElemento(): 8
```

```
10 "Os valores originais
11
12 for ( int valor : arran
13     System.out.printf( "
14
15 modificandoArray( arran
16     System.out.println( "\n
17
18 for ( int valor : arran
19     System.out.printf( "
20
21 System.out.printf(
22     "\n\nEfeitos da passa
23     "arranjo[3] antes de
24
25 |
26 modificandoArray( arran
27     System.out.printf(
28     "arranjo[3] depois da chamada a modificarElemento(): %d\n", arranjo[3] );
29
30 public static void modificandoArray( int algum_array[] )
31 {
```

Ativar o Windows
Acesse Configurações para ativar o Windows.

Arrays

- Argumentos com
comprimentos
variáveis:

```
public class ComprimentosVariaveis {  
  
    // calculando a média dos elementos  
    public static double media( double... valores )  
    {  
        double total = 0.0; // inicializando a variável total  
  
        for ( double d : valores )  
            total += d;  
  
        return total / valores.length;  
    } // end method average  
  
    public static void main( String[] args )  
    {  
        double d1 = 10.0;  
        double d2 = 20.0;  
        double d3 = 30.0;  
        double d4 = 40.0;  
  
        System.out.printf( "d1 = %.1f\nd2 = %.1f\nd3 = %.1f\nd4 = %.1f\n\n",  
            d1, d2, d3, d4 );  
  
        System.out.printf( "A média entre d1 e d2 é %.1f\n",  
            media( d1, d2 ) );  
        System.out.printf( "A média entre d1, d2 e d3 é %.1f\n",  
            media( d1, d2, d3 ) );  
        System.out.printf( "A média entre d1, d2, d3 e d4 é %.1f\n",  
            media( d1, d2, d3, d4 ) );  
    }  
}
```

Argumentos
variáveis!

Chamadas
com listas
variáveis de
argumentos

Arrays

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows a project named 'ComprimentosVariaveis' with a source folder 'src' containing the file 'ComprimentosVariaveis.java'.
- Editor:** Displays the code for 'ComprimentosVariaveis.java'. The code includes a static method 'media' that calculates the average of an array of doubles, and a 'main' method that initializes an array 'valores' with values 10.0, 20.0, 30.0, and 40.0, and prints the averages for different subsets of the array.
- Console:** Shows the output of the application, which matches the code's output statements.

```
public class ComprimentosVariaveis {  
    // calculando a média dos elementos  
    public static double media( double[] valores )  
    {  
        double total = 0.0; // inicializa  
        for ( double d : valores )  
            total += d;  
        return total / valores.length;  
    } // end method average  
  
    public static void main( String[] args )  
    {  
        double d1 = 10.0;  
        double d2 = 20.0;  
        double d3 = 30.0;  
        double d4 = 40.0;  
  
        System.out.printf( "d1 = %.1f\n", d1 );  
        System.out.printf( "d2 = %.1f\n", d2 );  
        System.out.printf( "d3 = %.1f\n", d3 );  
        System.out.printf( "d4 = %.1f\n", d4 );  
  
        System.out.printf( "A média entre d1 e d2 é %.1f\n",  
            media( new double[] { d1, d2 } ) );  
        System.out.printf( "A média entre d1, d2 e d3 é %.1f\n",  
            media( new double[] { d1, d2, d3 } ) );  
        System.out.printf( "A média entre d1, d2, d3 e d4 é %.1f\n",  
            media( new double[] { d1, d2, d3, d4 } ) );  
    }  
}
```

Output in Console:

```
<terminated> ComprimentosVariaveis [Java Application]  
d1 = 10,0  
d2 = 20,0  
d3 = 30,0  
d4 = 40,0  
  
A média entre d1 e d2 é 15,0  
A média entre d1, d2 e d3 é 20,0  
A média entre d1, d2, d3 e d4 é 25,0
```

A Classe Array

- Operações:
 - Ordenar
 - Preencher
 - Copiar
 - Comparar
 - Buscar elemento

A Classe Array

- Exemplo

Estrutura básica

```
import java.util.Arrays;

public class Manipulacao_De_Arrays {
    public static void main( String[] args )
    {
```

Ordenação

```
        // ordenação
        double[] arranjoDouble = { 8.4, 9.3, 0.2, 7.9, 3.4 };
        Arrays.sort( arranjoDouble );
        System.out.printf( "\nArray de doubles: " );

        for ( double valor : arranjoDouble )
            System.out.printf( "%.1f ", valor );
```

Preenchendo um array

```
        // preenchendo um array com o valor 7
        int[] arrayDeInteirosPreenchido = new int[ 10 ];
        Arrays.fill( arrayDeInteirosPreenchido, 7 );
        displayArray( arrayDeInteirosPreenchido, "arrayDeInteiroPreenchido" );
```

A Classe Array

- Exemplo

Copiando arrays

```
// Copiando arrays
int[] arrayInt = { 1, 2, 3, 4, 5, 6 };
int[] arrayIntCopia = new int[ arrayInt.length ];
System.arraycopy( arrayInt, 0, arrayIntCopia, 0, arrayInt.length );
displayArray( arrayInt, "arrayInt" );
displayArray( arrayIntCopia, "arrayIntCopia" );
```

Comparando arrays

```
// comparando arrays
boolean b = Arrays.equals( arrayInt, arrayIntCopia );
System.out.printf( "\n\narrayInt %s arrayIntCopia\n",
    ( b ? "==" : "!=" ) );

// outra comparação
b = Arrays.equals( arrayIntCopia, arrayDeInteirosPreenchido );
System.out.printf( "intArray %s filledIntArray\n",
    ( b ? "==" : "!=" ) );
```

A Classe Array

- Exemplo

Buscando valores
específicos em um array

```
// buscando elemento de valor 5
int localizacao = Arrays.binarySearch( arrayInt, 5 );

if ( localizacao >= 0 )
    System.out.printf(
        "O elemento 5 foi encontrado na posição %d do arrayInt\n", localizacao );
else
    System.out.println( "O valor 5 não foi encontrado em intArray" );

// buscando o elemento 8763
localizacao = Arrays.binarySearch( arrayInt, 8763 );

if ( localizacao >= 0 )
    System.out.printf(
        "O elemento 8763 foi encontrado na posição %d do arrayInt\n", localizacao );
else
    System.out.println( "O valor 8763 não foi encontrado em intArray" );
}
```

A Classe Array

- Exemplo

Método para
exibir o array

```
public static void displayArray( int[] array, String descricao )  
{  
    System.out.printf( "\n%s: ", descricao );  
  
    for ( int valor : array )  
        System.out.printf( "%d ", valor );  
}
```

A Classe Array

The screenshot shows the Eclipse IDE with a Java project named 'Manipulacao_De_Arrays'. The source code in 'Manipulacao_De_Arrays.java' includes imports for 'java.util.Arrays' and a 'main' method. The code creates a double array, sorts it, prints it, and then creates and compares two integer arrays. The console output, shown in two windows, displays the results of these operations. A red box highlights the console output in the top window, and a red arrow points from this box to the console output in the bottom window.

```
1 import java.util.Arrays;
2
3
4 public class Manipulacao_De_Arrays {
5     public static void main( String[] args)
6     {
7         // ordenação
8         double[] arranjoDouble = { 8.4,
9         Arrays.sort( arranjoDouble );
10        System.out.printf( "\nArray de d
11
12        for ( double valor : arranjoDoub
13            System.out.printf( "%.1f ", v
14
15        // preenchendo um array com o va
16        int[] arrayDeInteirosPreenchido
17        Arrays.fill( arrayDeInteirosPree
18        displayArray( arrayDeInteirosPre
19
20        // Copiando arrays
21        int[] arrayInt = { 1, 2, 3, 4, 5
22        int[] arrayIntCopia = new int[ a
23        System.arraycopy( arrayInt, 0, a
```

<terminated> Manipulacao_De_Arrays [Java Application] C:\Program
Array de doubles: 0,2 3,4 7,9 8,4 9,3
arrayDeInteiroPreenchido: 7 7 7 7 7 7 7 7 7 7
arrayInt: 1 2 3 4 5 6
arrayIntCopia: 1 2 3 4 5 6
arrayInt == arrayIntCopia
intArray != filledIntArray
O elemento 5 foi encontrado na posição 4 do arrayInt
O valor 8763 não foi encontrado em intArray

<terminated> Manipulacao_De_Arrays [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (10 de mar de 2019 14:54:42)
Array de doubles: 0,2 3,4 7,9 8,4 9,3
arrayDeInteiroPreenchido: 7 7 7 7 7 7 7 7 7 7
arrayInt: 1 2 3 4 5 6
arrayIntCopia: 1 2 3 4 5 6
arrayInt == arrayIntCopia
intArray != filledIntArray
O elemento 5 foi encontrado na posição 4 do arrayInt
O valor 8763 não foi encontrado em intArray

Classe ArrayList

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.

Classe ArrayList

```
import java.util.ArrayList;

public class Lista {

    public static void main( String[] args )
    {
        ArrayList< String > cores = new ArrayList< String >();

        cores.add( "vermelho" );
        cores.add( 0, "amarelo" );

        System.out.print(
            "Mostrando a lista de cores:" );

        for ( int i = 0; i < cores.size(); i++ )
            System.out.printf( " %5s", cores.get( i ) );

        // mostrando a lista chamando o método display
        display( cores,
            "\nMostrando a lista com outro comando: " );

        cores.add( "verde" ); // acrescenta no fim da lista
        cores.add( "amarelo" );
        display( cores, "Lista com dois novos elementos:" );
    }
}
```

Acrescenta "vermelho" no final da lista

Acrescenta "amarelo" no começo da lista

Retorna elemento da lista

Classe ArrayList

```
cores.remove( "amarelo" ); // remove o primeiro amarelo
display( cores, "Lista com a primeira instância de amarelo removida:" );

cores.remove( 1 ); // remove o item de índice 1
display( cores, "Removido o segundo elemento da lista (no caso, o verde):" );

// verificando se a lista contém um determinado valor
System.out.printf( "\"vermelho\" %s na lista\n",
    cores.contains( "vermelho" ) ? "está": "não está " );

// mostra o número de elementos da lista
System.out.printf( "Tamanho da lista: %s\n", cores.size() );
}

// método display
public static void display( ArrayList< String > cores, String inicio )
{
    System.out.print( inicio );

    for ( String item : cores )
        System.out.printf( " %s", item );

    System.out.println();
}
}
```

Classe ArrayList

The screenshot shows the Eclipse IDE with a Java project named 'Lista'. The Package Explorer on the left shows the project structure, including a 'Lista' package with a 'Lista.java' file. The main editor displays the source code of 'Lista.java', which uses the 'ArrayList' class from 'java.util'. The code includes imports, class declarations, and methods for adding, removing, and displaying elements. The console at the bottom shows the output of the program, which is highlighted by a red box. The output text is as follows:

```
<terminated> Lista [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (10 de mar d
Mostrando a lista de cores: amarelo vermelho
Mostrando a lista com outro comando: amarelo vermelho
Lista com dois novos elementos: amarelo vermelho verde amarelo
Lista com a primeira instância de amarelo removida: vermelho verde amarelo
Removido o segundo elemento da lista (no caso, o verde): vermelho amarelo
"vermelho" está na lista
Tamanho da lista: 2
```

The console output is repeated in a smaller window below it. A red arrow points from the first instance of the output to the second. The status bar at the bottom shows the file is 'Writable', 'Smart Insert' is enabled, and the cursor is at line 42, column 1. The system tray at the bottom right shows the date and time as '10/03/2019 15:32'.

Dúvidas?