Arranjos

TIPO DE DADOS
ARRANJO EM JAVA

Concluído anteriormente...

Nos vídeos anteriores, foram apresentados os seguintes Tipos Abstratos de Dados:

- Arranjo
- Listas
- Pilhas
- Filas

Objetivo deste tutorial:

Apresentar a implementação de Arranjos 1D em JAVA (Tipo de dados Arranjo 1D).

Utilizar algumas particularidades da Linguagem JAVA na criação e manipulação de Arranjos.

Apresentar diversos exemplos sobre criação e manipulação de Arranjos 1D em Java.

Deixar alguns exercícios práticos para serem resolvidos.

TD Arranjos em JAVA

Na Linguagem de programação JAVA, assim como em outras LP´s, os arranjos são modos tradicionais de armazenamento de dados.

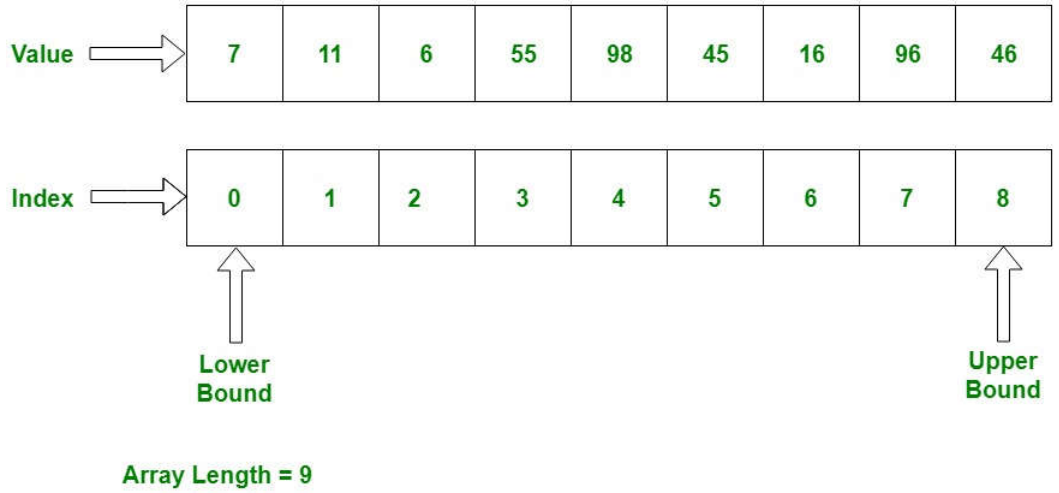
Basicamente, são estruturas de armazenamento homogêneas (armazenam um conjunto de dados do mesmo tipo)

Podem ser lineares (vetores) ou não (matrizes).

Em sua forma básica, são estáticas (tamanho definido no momento da sua criação).

TD Arranjos em JAVA

Para implementar o mapeamento índice x posição, o Java adota o índice [0] para indicar a posição do primeiro elemento armazenado em um arranjo unidimensional.



TD Arranjo em JAVA

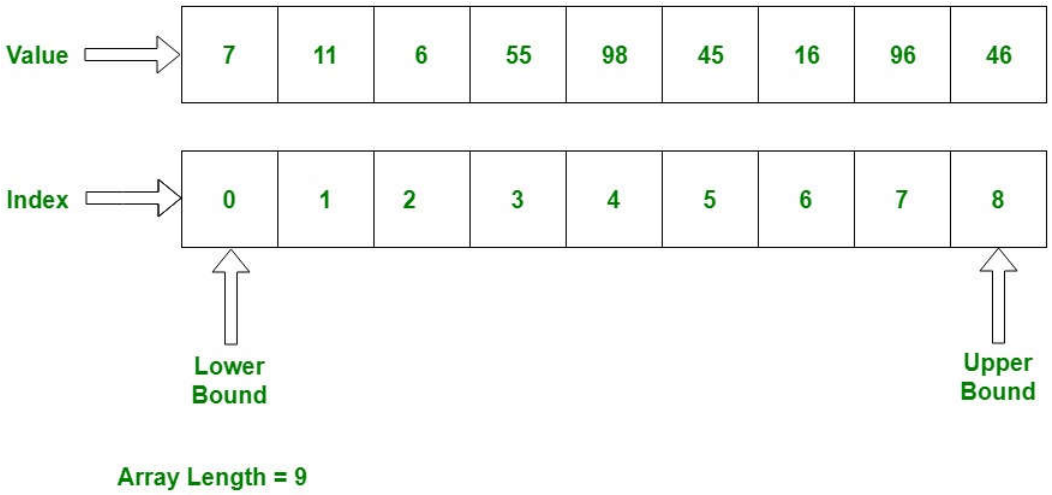
Java possui uma forma bem simples de criar arranjos:

```
tipo_do_elemento[] nome_do_arranjo = {val_0, val_1, val_2, val_N-1}
```

- ***tipo_do_elemento*** pode ser qualquer tipo base de JAVA ou um nome de classe.
- ***nome_do_arranjo*** pode ser qualquer identificador válido
- ***val_0*** ... ***val_N-1*** devem ser do mesmo tipo que o tipo_do_elemento.

TD Arranjos em JAVA

Vamos criar um arranjo unidimensional para armazenar o nosso primeiro exemplo de arranjo:



tipo_do_elemento: ***int***
nome_do_arranjo: ***a***

TD Arranjos em Java

```java
class Main {
  public static void main(String[] args) {
    int[] a = {7,11,6,55,98,45,16,96,46};
    System.out.println(a[0]);
    System.out.println(a[1]);
    System.out.println(a[2]);
    System.out.println(a[3]);
    System.out.println(a[4]);
    System.out.println(a[5]);
    System.out.println(a[6]);
    System.out.println(a[7]);
    System.out.println(a[8]);
  }
}
```

SAÍDA:

7

11

6

55

98

45

16

96

46

TD Arranjos em JAVA

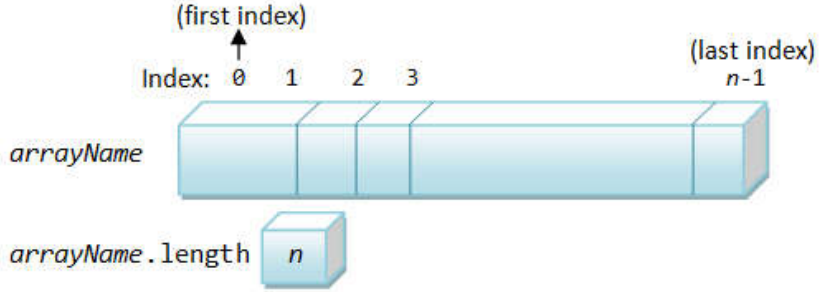
```java
class Main {
  public static void main(String[] args) {
    int[] a = {7,11,6,55,98,45,16,96,46};
    System.out.println(a[0]);
    System.out.println(a[1]);
    System.out.println(a[2]);
    System.out.println(a[3]);
    System.out.println(a[4]);
    System.out.println(a[5]);
    System.out.println(a[6]);
    System.out.println(a[7]);
    System.out.println(a[8]);
    System.out.println(a[9]);
  }
}
```

SAÍDA:

```
7
11
6
55
98
45
16
96
46
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException
: Index 9 out of bounds for length 9 at
Main.main(Main.java:13)
```

TD Arranjos em JAVA

Em JAVA, os arranjos são considerados objetos e por isso possuem métodos associados.



Os arranjos em JAVA possuem um método que retorna o tamanho do arranjo.

```
nome_do_arranjo.length
```

Com isso, as posições válidas em um arranjo unidimensional estão entre

```
0 e nome_do_arranjo.length-1
```

TD Arranjos em Java

```java
class Main {
  public static void main(String[] args) {
    int[] a = {7,11,6,55,98,45,16,96,46};
    for(int i=0; i<a.length; i++)
      System.out.println(a[i]);
  }
}
```

SAÍDA:

7

11

6

55

98

45

16

96

46

TD Arranjos em Java

Em Java, os Arranjos são objetos e por isso precisam ser instanciados

```
1  public class Prog005{
2    public static void main(String args[]){
3      int a[];
4      System.out.println(a);
5    }
6  }
```

TD Arranjos em Java

Em Java, os Arranjos são objetos e por isso precisam ser instanciados

```
1  public class Prog005{
2    public static void main(String args[]){
3      int a[];
4      System.out.println(a);
5    }
6  }
```

```
C:\Users\eduar\Desktop\ED>javac Prog005.java
Prog005.java:4: error: variable a might not have been initialized
    System.out.println(a);
                       ^
1 error
```

TD Arranjos em JAVA

```
1  public class Prog006{
2    public static void main(String args[]){
3      int a[];
4      a = new int[5];
5      System.out.println(a);
6    }
7  }
```

TD Arranjos em JAVA

```java
public class Prog006{
  public static void main(String args[]){
    int a[];
    a = new int[5];
    System.out.println(a);
  }
}
```

```
C:\Users\eduar\Desktop\ED>java Prog006
[I@15db9742
```

TD Arranjos em JAVA

```java
public class Prog006{
  public static void main(String args[]){
    int a[];
    a = new int[5];
    System.out.println(a);
  }
}
```

```
C:\Users\eduar\Desktop\ED>java Prog006
[I@15db9742
```

O que é [I@15db9742 ?

TD Arranjos em JAVA

```java
public class Prog006{
  public static void main(String args[]){
    int a[];
    a = new int[5];
    System.out.println(a);
  }
}
```

```
C:\Users\eduar\Desktop\ED>java Prog006
[I@15db9742
```

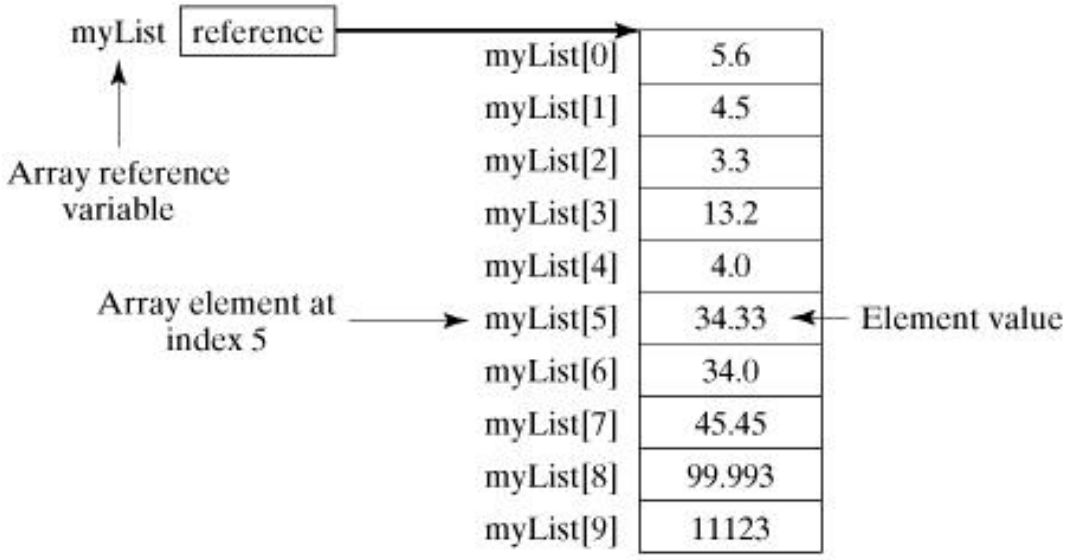
O que é [I@15db9742 ?

[I indica Array de Inteiros

@15db9742 é um endereço de memória

TD Arranjos em Java

```
double[] myList = new double[10];
```



TD Arranjo em Java

Arrays em JAVA

- Ocupam posições contínuas de memória RAM
- Mapeamento índice x posição de memória

TD Arranjo em Java

Arrays em JAVA

- Ocupam posições contínuas de memória RAM
- Mapeamento índice x posição de memória

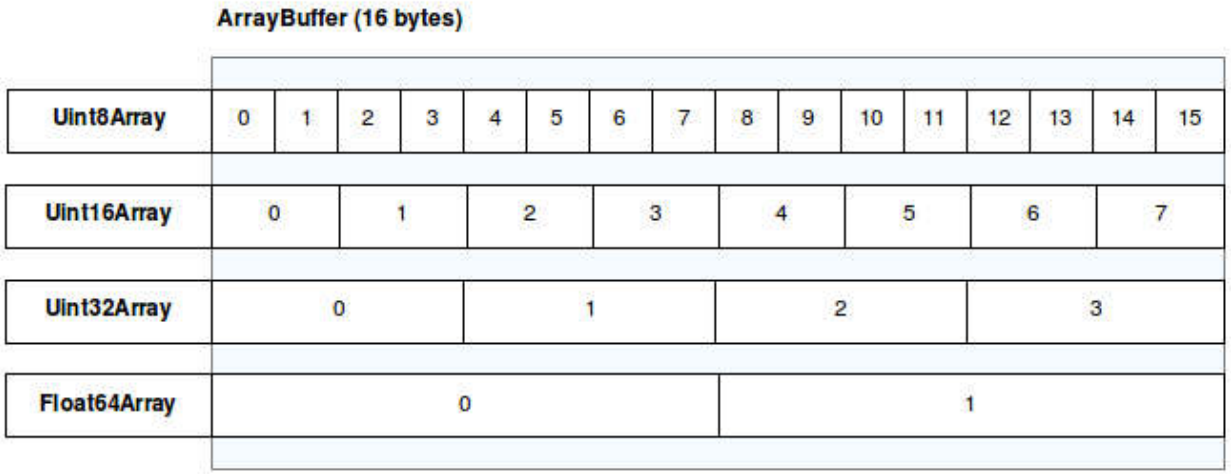
```
tipo[] arranjo = new tipo[N];
arranjo[indice] => arranjo + (indice * sizeof(tipo))
```

TD Arranjo em Java

Arrays em JAVA

- Ocupam posições contínuas de memória RAM
- Mapeamento índice x posição de memória

```
tipo[] arranjo = new tipo[N];
arranjo[indice] => arranjo + (indice * sizeof(tipo))
```



TD Arranjo em Java

Algumas informações importantes sobre arranjos em JAVA:

- São considerados objetos => variáveis referência e métodos.
- Variável referência => guarda o tipo e endereço de memória do objeto

  [I@15db9742

  [I indica Array de Inteiros

  @15db9742 é um endereço de memória
- Mapeamento índice => posição

```
tipo[] arranjo = new tipo[N];
arranjo[indice] => arranjo + (indice * sizeof(tipo))
```

TD Arranjos em JAVA

IMPORTANTE:

- Variáveis referência armazenam o endereço de memória em que os dados do objeto estão armazenados.
- Copiar uma variável referência, significa copiar endereço de um objeto e não os dados do objeto!

TD Arranjos em JAVA

```java
public class Prog007{
  public static void main(String args[]){
    int[] a = {2, 4, 6};
    int[] b = a;
    System.out.println(b[0]);
    b[1] = 8;
    for(int i=0; i<a.length; i++)
      System.out.print(a[i] + "  ");
  }
}
```

TD Arranjos em JAVA

```java
public class Prog007{
  public static void main(String args[]){
    int[] a = {2, 4, 6};
    int[] b = a;
    System.out.println(b[0]);
    b[1] = 8;
    for(int i=0; i<a.length; i++)
      System.out.print(a[i] + "  ");
  }
}
```

```
C:\Users\eduar\Desktop\ED>java Prog007
2
2  8  6
```

TD Arranjos em JAVA

```java
public class Prog007{
  public static void main(String args[]){
    int[] a = {2, 4, 6};
    int[] b = a;
    System.out.println(b[0]);
    b[1] = 8;
    for(int i=0; i<a.length; i++)
      System.out.print(a[i] + "  ");
  }
}
```

```
C:\Users\eduar\Desktop\ED>java Prog007
2
2  8  6
```

Cuidado com as cópias!

Copia de Arranjos em JAVA

Em JAVA existem várias formas de copiar o conteúdo de um arranjo em outro arranjo.

Neste tutorial, iremos utilizar:

- Um loop usando um comando for.
- O método clone().

Cópia de Arranjos em JAVA usando loop.

Maneira mais tradicional e geral para copiar Arranjos.

- Pode ser implementada em qualquer linguagem de programação.

Cópia de Arranjos em JAVA usando loop.

Maneira mais tradicional e geral para copiar Arranjos.

- Pode ser implementada em qualquer linguagem de programação.

Procedimento:

- Inicialmente, reservamos espaço para o Arranjo destino
- Em seguida, um laço varre o Arranjo original e, a cada posição acessada, copia o conteúdo para a mesma posição do Arranjo destino.

Cópia de Arranjos em JAVA usando loop.

```
1  public class Prog008{
2    public static void main(String args[]){
3      int[] a = {2, 4, 6};
4      // reserva de memória para b (arranjo destino)
5      int[] b = new int[a.length];
6      // loop para copiar os valores de cada posição
7      for(int i=0; i<b.length; i++)
8        b[i] = a[i];
9      System.out.print(a + " " + b + "\n");
10     for(int i=0; i<b.length; i++)
11       System.out.print(b[i] + "  ");
12   }
13 }
```

Cópia de Arranjos em JAVA usando loop.

```java
public class Prog008{
  public static void main(String args[]){
    int[] a = {2, 4, 6};
    // reserva de memória para b (arranjo destino)
    int[] b = new int[a.length];
    // loop para copiar os valores de cada posição
    for(int i=0; i<b.length; i++)
      b[i] = a[i];
    System.out.print(a + " " + b + "\n");
    for(int i=0; i<b.length; i++)
      System.out.print(b[i] + "  ");
  }
}
```

```
C:\Users\eduar\Desktop\ED>java Prog008
[I@15db9742 [I@6d06d69c
2  4  6
```

Cópia de Arranjos em JAVA com método clone().

Na linguagem JAVA, o método clone() é utilizado para clonar um objeto.

- Não é necessário reservar espaço para o objeto destino.
- Objeto destino deve ser do mesmo tipo que o objeto clonado.

Cópia de Arranjos em JAVA com método clone().

Na linguagem JAVA, o método clone() é utilizado para clonar um objeto.

- Não é necessário reservar espaço para o objeto destino.
- Objeto destino deve ser do mesmo tipo que o objeto clonado.

Procedimento:

- Variável do objeto destino recebe o retorno do método clone() do objeto origem.

Cópia de Arranjos em JAVA com método clone().

```
1  public class Prog009{
2    public static void main(String args[]){
3      int[] a = {2, 4, 6};
4      // b recebe uma cópia de a
5      int[] b = a.clone();
6      System.out.print(a + " " + b + "\n");
7      for(int i=0; i<b.length; i++)
8        System.out.print(b[i] + "  ");
9    }
10 }
```

Cópia de Arranjos em JAVA com método clone().

```java
public class Prog009{
  public static void main(String args[]){
    int[] a = {2, 4, 6};
    // b recebe uma cópia de a
    int[] b = a.clone();
    System.out.print(a + " " + b + "\n");
    for(int i=0; i<b.length; i++)
      System.out.print(b[i] + "  ");
  }
}
```

```
C:\Users\eduar\Desktop\ED>java Prog009
[I@15db9742 [I@6d06d69c
2  4  6
```

Exercício 01

Escreva um programa em JAVA que crie um vetor de inteiros com 10 elementos.

Esse vetor deve ser preenchido com números pares começando em 2.

Imprima na tela os valores armazenados nesse vetor.

Exercício 02

Escreva um programa em JAVA que crie um vetor de inteiros com 10 elementos.

Esse vetor deve ser preenchido com números primos.

Imprima na tela os valores armazenados no vetor.

Exercício 03

Escreva um programa em JAVA que crie um vetor de inteiros com 10 elementos.

Esse vetor deve ser preenchido com números pares começando em 2.

Crie uma cópia do vetor.

Adicione 1 para cada elemento do vetor original.

Apresente os valores dos dois vetores.

Código Base para exercícios 04 a 07

```
1  public class Exercicios04_a_07{
2    public static void main(String args[]){
3      float[] a = {-1.7,3.0,0.0,1.5,0.0,-1.7,2.3,-1,7};
4
5    }
6  }
```

Exercício 04

Modifique o código base para criar um novo arranjo b, onde os valores são os mesmos de a, porém em ordem inversa.

Exercício 05

Modifique o código base para criar um novo arranjo b, onde os valores são os mesmos de a, porém em ordem crescente.

Exercício 06

Modifique o código base para criar um novo arranjo b, onde os valores são definidos pelas seguintes regras:

- Para as posições de b em que em a esteja um valor negativo, deve-se armazenar -1.
- Para as posições de b em que em a esteja um valor positivo, deve-se armazenar 1.
- Para as posições de b em que em a esteja um valor 0, deve-se armazenar 0.

Exercício 07

Modifique o código base para que a saída seja dada pelas seguintes mensagens e valores correspondentes:

A) quantidade de valores: ___

B) maior valor: _____

C) menor valor: _____

D) valor médio: _____

Exercício 08

Explique as duas linhas de código abaixo

```
int a[], b[], c[];

int[] a, b, c;
```

Conclusões:

Foi apresentado a criação e uso de Arranjos em JAVA.

Restringiu-se a Arranjos unidimensionais.

Verificou-se que em Java os arranjos são objetos.

- Precisam ser instanciados e quando são instanciados a variável de referência do Arranjo guarda o endereço de memória da primeira posição do Arranjo
- Cópia simples de um arranjo na verdade copia o endereço do Arranjo.

Foram apresentadas duas formas de realizar uma cópia de Arranjos.

- A primeira por meio de um loop e que pode ser utilizada em qualquer linguagem.
- A segunda usando o método clone() da linguagem Java.

Foram propostos exercícios práticos.