

Programação OO em Java

Profa Andréa Schwertner Charão
DLSC/CT/UFSM

Sumário

- Arrays em Java
 - Declaração
 - Alocação
 - Acesso
 - Percurso
- Pacotes em Java
 - Uso de import
 - Declarando pacotes

Arrays em Java

- Arrays representam um conjunto homogêneo de dados, de tamanho fixo, alocados de forma contígua na memória
- Ao contrário de C, arrays em Java são **objetos**
- Assim como em C, os elementos de um array são numerados a partir de **zero**

Declarando e alocando arrays

- Arrays podem ser de tipos primitivos (int, double, etc.) ou de tipos abstratos de dados (classes, como por exemplo String, Date, etc.).
- Em qualquer caso, a forma de declarar é a mesma, usando colchetes:

```
int[] a1;  
String[] s1;
```

- A declaração cria apenas uma **referência** para um objeto array (não cria o objeto). O valor desta referência é **null** por padrão.

Declarando e alocando arrays

- Para criar o objeto array (e alocá-lo na memória), geralmente se usa "new"

```
int[] a1 = new int[5];
```

```
int[] a2;
```

```
int tam = 10;
```

```
a2 = new int[tam];
```

```
String[] s1;
```

```
s1 = new String[2];
```

Declarando e alocando arrays

- Quando se usa `new`, cada elemento do array é inicializado com um valor default:

- **zero** para tipos numéricos primitivos

```
int[] v = new int[2];  
// v[0] e v[1] valem 0
```

- **null** quando os elementos são referências para objetos

```
String[] s = new String[2];  
// s[0] e s[1] valem null
```

Declarando e alocando arrays

- Quando array é de objetos
 - cada elemento é uma referência para um objeto
 - é preciso **criar cada um dos objetos** do array

```
class MyClass {
    public MyClass() {
        System.out.println("Objeto criado!");
    }
}

class ArrayApp {
    public static void main(String[] args) {
        MyClass[] objs = new MyClass[2];
        objs[0] = new MyClass(); // objeto criado
        objs[1] = new MyClass(); // objeto criado
    }
}
```

Declarando e alocando arrays

- Arrays também podem ser criados usando inicializadores (neste caso não se usa new):

```
String[] s2 = {"Ola", "Mundo"};
```

```
double[] ds = {0.1,  
               Math.sqrt(2),  
               Math.sqrt(3)};
```


Acessando elementos do array

- Acesso como em C:

```
int[] v = {1, 2};  
System.out.println(v[0]);  
v[1] = 3;
```

- Atributo length contém o tamanho do array (esse atributo não pode ser alterado)
- Índices de 0 a length-1
- Índices fora dos limites produzem exceção do tipo `ArrayIndexOutOfBoundsException`

```
System.out.println(arr[arr.length+2]);
```

Percorrendo um array

- Laço **for** tradicional, com índice

```
int[] numbers = {1,2,3,4};  
  
for (int i = 0; i < numbers.length; i++) {  
    System.out.println(numbers[i]);  
}
```

Percorrendo um array: outra alternativa

- Laço **for** melhorado
- Também conhecido como **for-each**
- Inspirado na programação funcional (map)

```
for (int n : numbers) {  
    System.out.println(n);  
}  
// int[] numbers = {1,2,3,4};
```

tipo dos elementos do array

referência para elemento do array

referência para o array

Mais sobre arrays em...

Arrays (The Java™ Tutorials > Learning the Java Language > Language Basics) - Mozilla Firefox

docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html

The Java™ Tutorials

Download Ebooks
Download JDK
Search Java Tutorials
Hide TOC

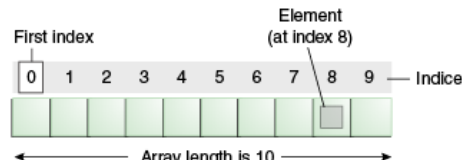
Language Basics
Variables
Primitive Data Types
Arrays
Summary of Variables
Questions and Exercises
Operators
Assignment, Arithmetic, and Unary Operators
Equality, Relational, and Conditional Operators
Bitwise and Bit Shift Operators
Summary of Operators
Questions and Exercises
Expressions, Statements, and

Home Page > Learning the Java Language > Language Basics

« Previous • Trail • Next »

Arrays

An *array* is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed. You've seen an example of arrays already, in the `main` method of the "Hello World!" application. This section discusses arrays in greater detail.



An array of ten elements

Each item in an array is called an *element*, and each element is accessed by its numerical *index*. As shown in the above illustration, numbering begins with 0. The 9th

<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>

Pacotes em Java

- Pacotes são conjuntos de classes em Java
- Os nomes de classes e pacotes seguem uma estrutura hierárquica, como se fossem arquivos organizados dentro de pastas
- Exemplos:
 - `java.lang` é o pacote das classes básicas da linguagem Java
 - `java.lang.String` é o nome completo da classe `String`, que está dentro do pacote `java.lang`)
 - `java.util` é um pacote com classes utilitárias
 - `java.util.Arrays` é o nome completo de uma classe com métodos utilitários para manipulação de arrays

Uso de import

- A palavra 'import', no início de um programa Java, serve para indicar um nome completo de classe a ser usada.
- Classes importadas podem ser chamadas pelos seus nomes curtos (ex.: Arrays em vez de `java.util.Arrays`).
- O uso de import é apenas uma conveniência para que os programas fiquem mais fáceis de ler.

Uso de import

- Exemplos:
- `'import java.util.Arrays;'` torna a classe `Arrays` acessível pelo seu nome curto
- `'import java.util.*;'` torna todas as classes desse pacote acessíveis pelos seus nomes curtos

Import versus classpath

- Atenção: o 'import' não é responsável por indicar a localização de uma classe ou pacote para a JVM
- O recurso em java responsável por isso é o 'classpath', que é uma opção de configuração da JVM
- Exemplo com linha de comando (-cp):

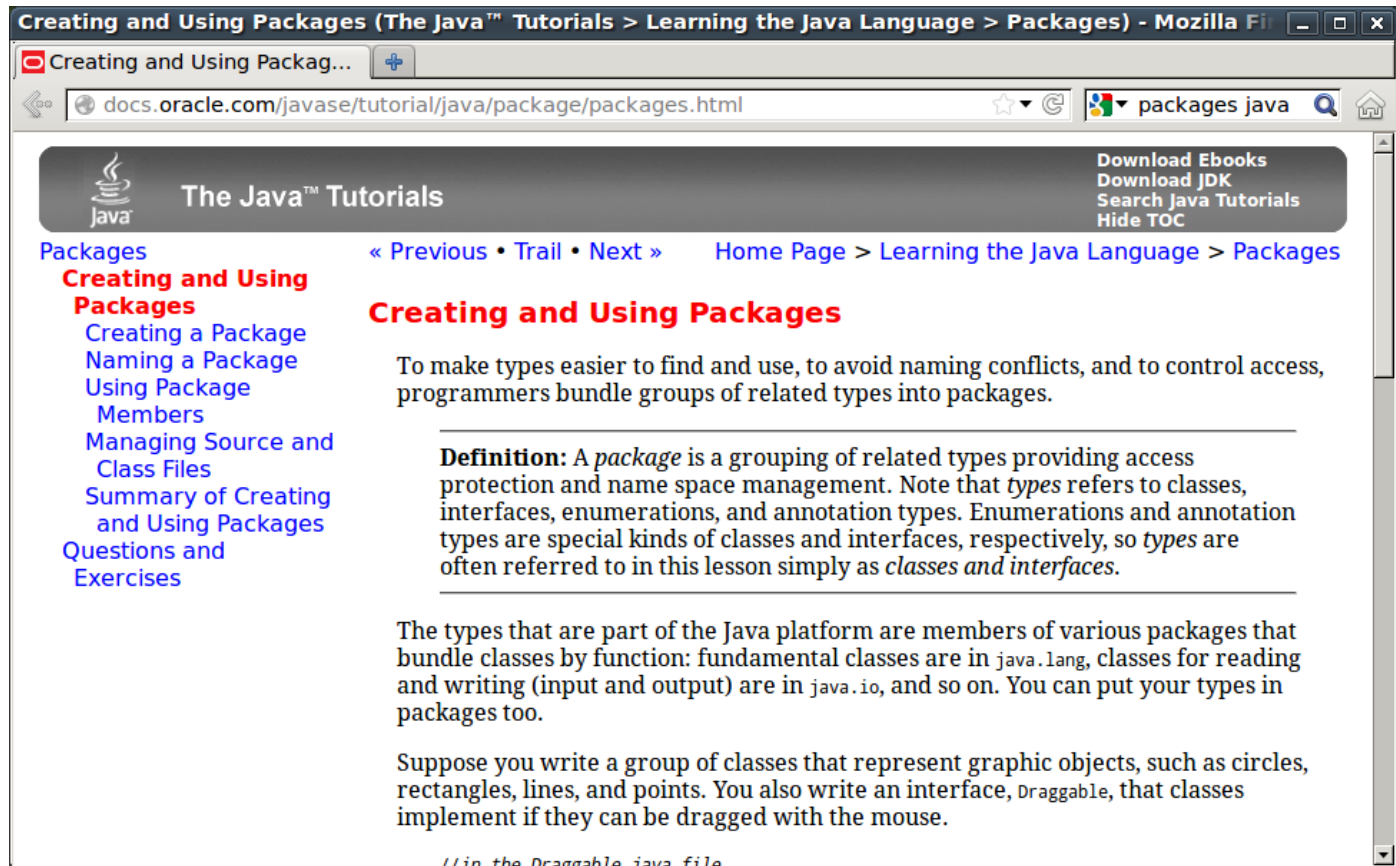
```
java -cp . SpaceShipTest
```

(indica que o classpath deve incluir a pasta '.', isto é, a pasta corrente)

Declarando pacotes

- Pode-se usar a palavra 'package' no início de um arquivo .java para indicar que a(s) classe(s) nele contida(s) farão parte de um pacote.
- Exemplos:
 - 'package t4;': pacote de classes do trabalho 4
 - 'package br.ufsm.inf;': pacote de classes de uma organização
 - etc.

Mais sobre pacotes em...



<http://docs.oracle.com/javase/tutorial/java/package/packages.html>