

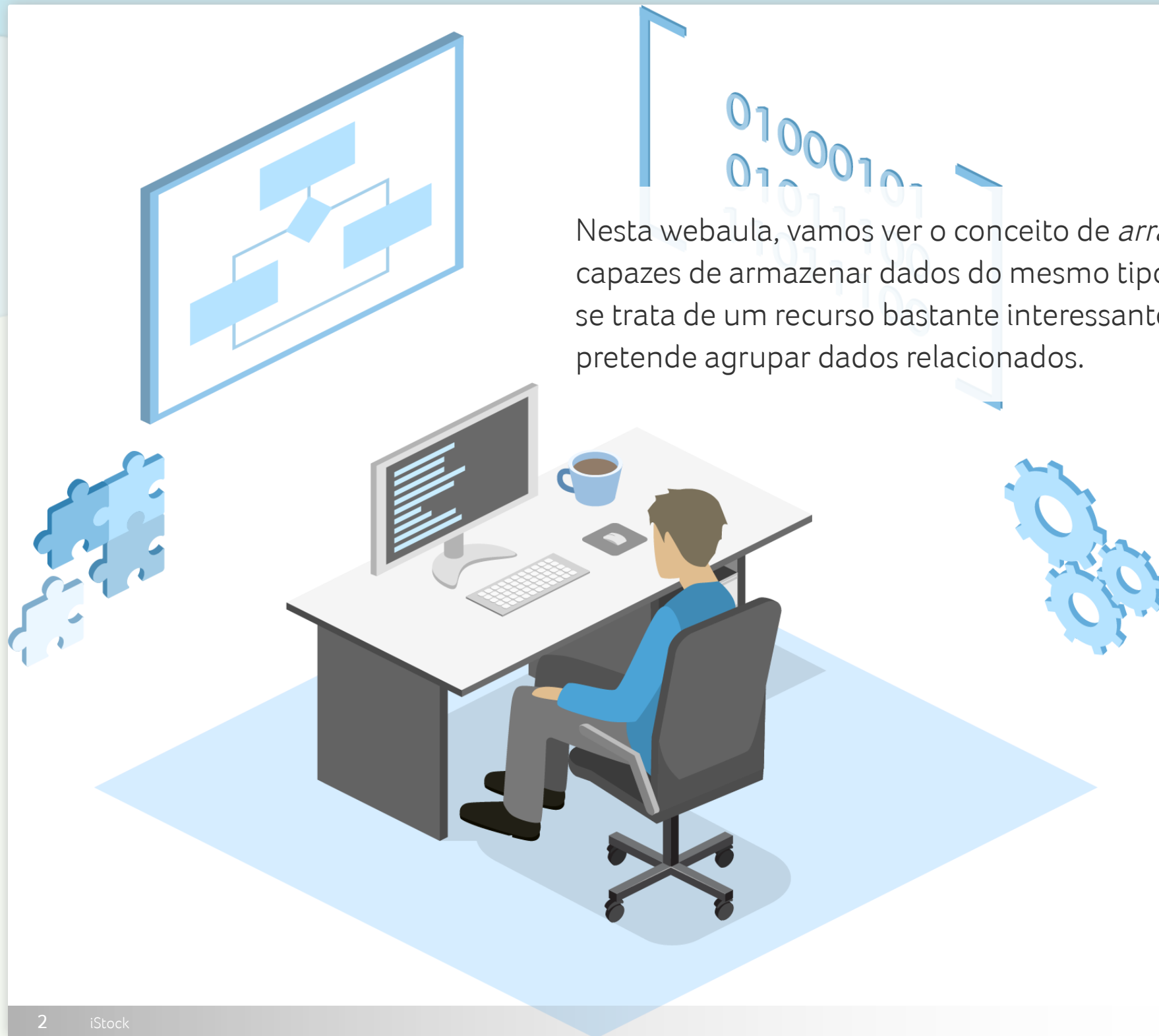
## Unidade 4

### Seção 1

# Programação Orientada a Objetos

# Webaula 1

## *Arrays em Java*



Nesta webaula, vamos ver o conceito de *arrays*, estruturas capazes de armazenar dados do mesmo tipo. Veremos que se trata de um recurso bastante interessante quando se pretende agrupar dados relacionados.

## *Array*

Um *array* é um grupo de variáveis (chamados elementos ou componentes) que contém valores do mesmo tipo. Eles superam a limitação das variáveis de armazenar um só valor numa mesma unidade de tempo. É um meio para armazenar mais do que um dado sem precisar declarar mais do que uma variável.



### Array de 10 elementos

b (nome do array)

elementos do array

7	2	12	0	6	3	2	10	4	3
b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]	b[7]	b[8]	b[9]

índices

Fonte: adaptado do autor

- O vetor de 10 elementos foi chamado de **b** e cada elemento do *array* representa um valor.
- Para referenciar um elemento na estrutura, utiliza-se o nome do *array* e índice (inteiro, não negativo), que é o número de posição do elemento no *array*.
- O primeiro elemento do *array* tem índice zero.
- A posição indicada por b[2] contém o elemento 12.

Os elementos devem ser de um único tipo, definido na declaração do *array*.

Os tipos primitivos já conhecidos são:

`byte`, `short`, `int`, `long`,  
`float`, `double`, `boolean` e `char`.

[Clique aqui para saber mais](#)

Para a criação da estrutura, é necessário criar uma instância de *array* e atribuí-la à variável que lhe dá o nome.

```
b = new int[10]
```

A variável de *array* `b` recebe a referência para um novo vetor de inteiros de 10 elementos.

No entanto, é comum declarar e incluir a expressão de criação do *array* numa mesma linha, como segue:

```
int b[ ] = new int[10]
```

Quando um *array* é criado, cada elemento seu recebe o valor zero para tipos numéricos, `false` para o tipo `boolean` e `null` para referências.

Para incluir elementos em um *array*, utilizam-se comandos individuais de atribuição ou de leitura pelo teclado.

Inicializar o *array* com os valores:

```
int[] valoresIniciais = {7, 12,  
58, 4, 15, 0}
```

O tamanho do *array* é determinado pela quantidade de elementos nesta lista.

Clique na imagem para visualizar a atribuição de valores ao *array* por meio da leitura pelo teclado.

```
public class Array1 {  
    public static void main (String[] args) {  
        int[] b = new int[4]; //declaração do vetor  
        /* Criação dos elementos do vetor */  
        /* por meio de comandos de atribuição */  
        /* estas atribuições poderiam ser substituídas */  
        /* por comandos de leitura pelo teclado. */  
        b[0] = 7;  
        b[1] = 2;  
        b[2] = 12;  
        b[3] = 0;  
        // Impressão individual de cada elemento do vetor  
        System.out.println (b[0]);  
        System.out.println (b[1]);  
        System.out.println (b[2]);  
        System.out.println (b[3]);  
    }  
}
```

Fonte: autor

```
import java.util.Scanner;
public class Array2 {
    public static void main (String[] args) {
        int i;
        int[] b = new int[10];
        Scanner entrada = new Scanner(System.in);
        //Entrada dos elementos do vetor.
        for (i=0;i<10;i++) {
            System.out.printf("Informe o %d elemento do
array: ", i+1);
            b[i] = entrada.nextInt();
        }
        // Impressão de cada elemento do vetor.
        for (i=0;i<10;i++) {
            System.out.printf("\nElemento %d: %d ",
i+1,b[i]);
        }
    }
}
```

Fonte: autor

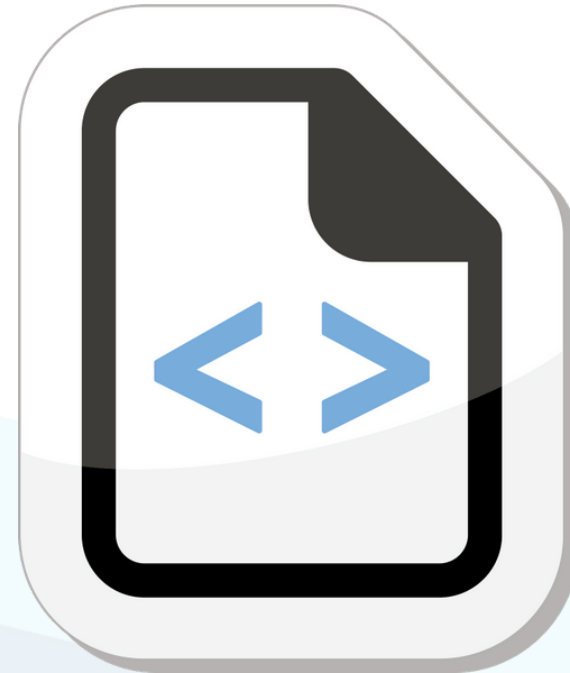
Exemplo de *array* para atribuir e imprimir valores em muitas posições.

- O índice do *array* é implementado pela variável *i*.
- Por meio do comando **for**, o índice varia de 0 a 9, permitindo percorrer as dez posições da estrutura.
- A cada iteração, um valor de elemento é obtido pelo comando de leitura do teclado.
- No segundo comando **for** é feita a impressão dos elementos.



## Classe Arrays

A classe `java.util.Arrays` oferece métodos e atributos para manipulação de arrays. Por exemplo, o atributo `length` retorna o tamanho (ou dimensão) do array que está em uso, o método `sort` retorna os valores em ordem crescente, entre outros.



## Arrays de instâncias de classes

Sabemos que um *array* suporta tipos primitivos e, no item dos *arrays* com mais de uma dimensão, trataremos de array que armazena outro *array*, tornando possível contar com a possibilidade de armazenarmos instâncias de classes nos vetores. Exemplo:

```
Automovel[] auto = new
Automovel[4];
auto[0] = new Automovel();
auto[1] = new Automovel();
auto[2] = new Automovel();
auto[3] = new Automovel();
```

Foi declarado um *array* de 4 elementos normalmente. No entanto, os elementos que compõem este *array* são instâncias da classe Automovel. O mesmo efeito seria obtido com a linha que segue:

```
Automovel[] auto = new
Automovel[] {new
Automovel(), new
Automovel(), new Automovel(),
new
Automovel()};
```

## Array unidimensional

Um *array* unidimensional possui apenas um índice para que seja possível o acesso a seus elementos, assim sua única dimensão é a largura.

```
import java.util.Scanner;
import java.util.Arrays;
public class SomaArray {
    public static void main (String[] args) {
        Scanner entrada = new Scanner(System.in);
        //uma constante é criada para conter a dimensão do
        vetor.
        final int tamanhoVetor = 8;
        int soma = 0; //variável que conterá a soma dos
        elementos.
        float media = 0; //variável que conterá a média dos
        elementos.
        int i;
        int a[] = new int[tamanhoVetor]; //declaração e
        criação do vetor.
        for (i=0; i < a.length; i++) {
            System.out.printf("Informe o %d elemento do
            array: ", i+1);
            a[i] = entrada.nextInt();
            soma = soma + a[i];
        }
        media = soma/8;
        System.out.printf("\nA soma dos elementos é %d ",
        soma);
        System.out.printf("\nO valor médio dos elementos é
        %.2f ", media);
    }
}
```

Fonte: autor

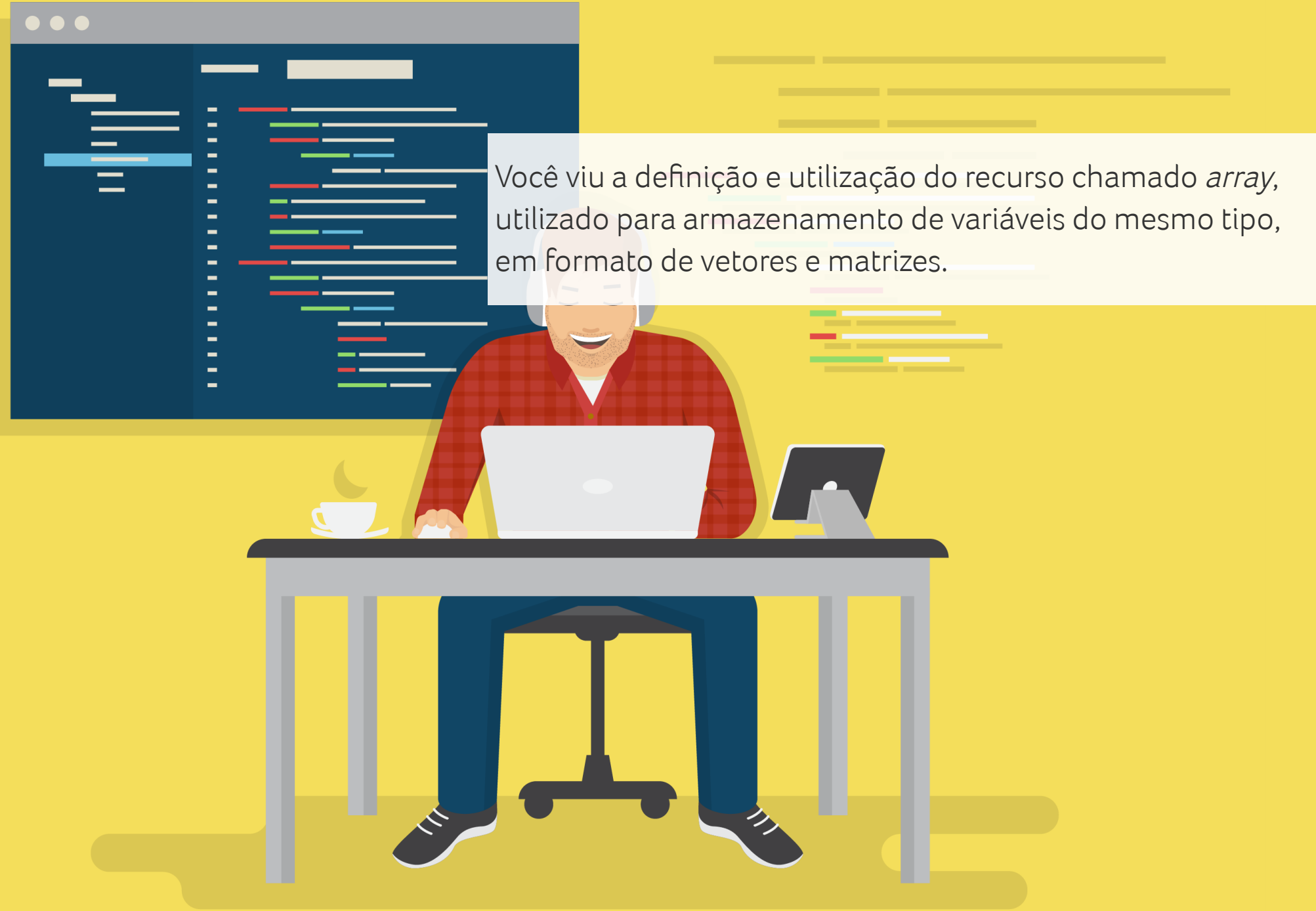
## Arrays bidimensionais

Os arrays bidimensionais, por sua vez, são representados por quadros com altura e largura, ou seja, possuem dois índices e possibilitam que os valores sejam armazenados na forma de matriz. Embora o Java não suporte *arrays* bidimensionais, é possível obter a mesma funcionalidade criando um *array* de *arrays*.

No exemplo, temos a **matriz** com dois vetores e a aplicação de dois **laços de repetição**.

```
public class ArrayBi {  
    public static void main (String[] args) {  
        int matriz [][] = new int [2][3];  
        int soma = 0;  
        int linha, coluna;  
        // inicialização da matriz por atribuição.  
        matriz [0][0] = 1; matriz [0][1] = 2; matriz [0][2]  
= 3;  
        matriz [1][0] = 4; matriz [1][1] = 5; matriz [1][2]  
= 6;  
        for (linha = 0; linha < 2; linha++)  
            for (coluna = 0; coluna < 3; coluna++) {  
                soma = soma + matriz [linha][coluna];  
            }  
        System.out.printf("\nA soma dos elementos é %d ",  
soma);  
    }  
}
```

Fonte: autor

An illustration of a person with grey hair and a beard, wearing a red plaid shirt and blue pants, sitting at a grey desk and smiling while working on a silver laptop. To the left of the laptop is a white coffee cup with a crescent moon on it. To the right is a small black monitor. In the background, there is a large dark blue window displaying a code editor with various colored lines of code. A white text box with a light blue border is overlaid on the right side of the code editor. The entire scene is set against a bright yellow background. There are blue navigation arrows in the top right and bottom right corners, and a grey menu button on the right edge.

Você viu a definição e utilização do recurso chamado *array*, utilizado para armazenamento de variáveis do mesmo tipo, em formato de vetores e matrizes.

## Você já conhece o Saber?

Aqui você tem na palma da sua mão a **biblioteca digital** para sua **formação profissional**.

Estude no celular, tablet ou PC em qualquer hora e lugar sem pagar mais nada por isso.

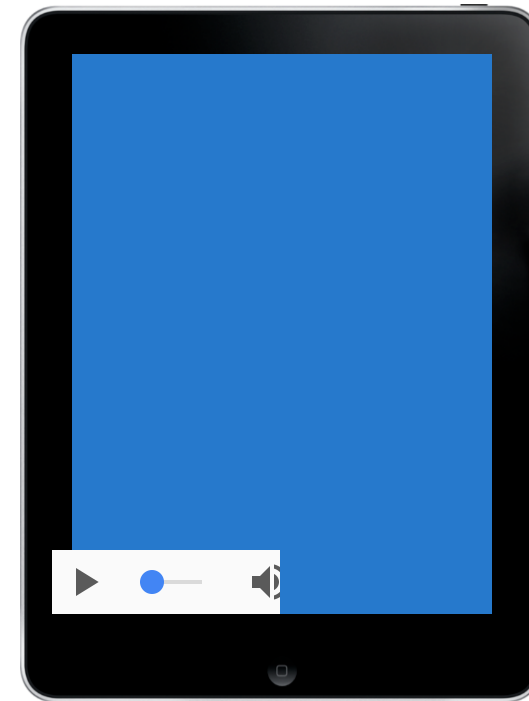
Mais de 450 livros com interatividade, vídeos, animações e jogos para você.



Android:  
<https://goo.gl/yAL2Mv>



iPhone e iPad - IOS:  
<https://goo.gl/OFWqcq>





Bons estudos!