

## Lab 4 – Vetores e Matrizes em Java

Neste laboratório construiremos e executaremos exemplos de programas que usam arrays em Java, também conhecidos como matrizes e vetores.

Duração prevista: 90 minutos

### Exercícios

**Exercício 1:** Construindo arrays de primitivos inteiros (40 minutos)

**Exercício 2:** Construindo arrays de objetos String (50 minutos)

### Exercício 1 – Construindo Arrays de Primitivos Inteiros

1. Usando arrays unidimensionais de inteiros
2. Usando arrays bidimensionais de inteiros
3. Busca maior valor

#### 1.1. Usando arrays unidimensionais de inteiros

1. Escreva, compile e execute a **Listagem-4.1, UniArray.java**, preste bastante atenção aos comentários.

```
public class UniArray {  
  
    public static void main(String[] args) {  
  
        // Declara e cria um array de int com tamanho 10  
        int[] idades = new int[10];  
  
        // Imprime o valor de cada elemento do array  
        for (int i = 0; i < idades.length; i++) {  
            System.out.print(idades[i]);  
        }  
    }  
}
```

Listagem 4.1 – UniArray.java

2. Compile e execute o programa, veja o resultados

0000000000

3. Antes do laço **for** que imprime os valores modifique o programa **UniArray.java**, crie outro laço **for** para inicializar os elementos do array **idades** de modo que cada entrada fique como na sequência: **idades[0]=31, idades[1]=32 ... idades[n]=N** e assim por diante.

## 1.2. Usando arrays bidimensionais de inteiros

1. Escreva, compile e execute a **Listagem-4.2, BiDArray.java**, preste bastante atenção aos comentários.

```
public class BiDArray {  
  
    public static void main(String[] args) {  
  
        // Declara e cria um array de duas dimensões de int com tamanho 10x5  
        int[][] idades = new int[10][5];  
  
        // Mostra numero de linhas e colunas  
        System.out.println("idades.length =" + idades.length); //linhas  
        System.out.println("idades.length[1] =" + idades[1].length); //colunas  
  
        // Imprime o valor de cada elemento do array  
        for (int i = 0; i < idades.length; i++) {  
            System.out.println("\n Linha inicio " + i);  
  
            for (int j = 0; j < idades[i].length; j++) {  
                idades[i][j] = i * j;  
                System.out.print(idades[i][j] + " ");  
            }  
        }  
    }  
}
```

Listagem 4.2 – BiDArray.java

2. Compile e execute o programa, veja o resultado:

```
idades.length=10  
idades.length[1]=5
```

```
Linha 0  
0 0 0 0 0  
Linha 1  
0 1 2 3 4  
Linha 2  
0 2 4 6 8  
Linha 3  
0 3 6 9 12  
Linha 4  
0 4 8 12 16  
Linha 5  
0 5 10 15 20  
Linha 6  
0 6 12 18 24  
Linha 7  
0 7 14 21 28  
Linha 8  
0 8 16 24 32  
Linha 9  
0 9 18 27 36
```

3. Crie um programa **TriDArray.java** que declara um **array tridimensional** de inteiros e inicialize cada célula com um número

crescente começando de **1000, 1001, 1002 ... N**, assim por diante.

### 1.3. Busca maior valor

1. Escreva, compile e execute a **Listagem-4.3, MaiorNumero.java**, preste bastante atenção aos comentários.

```
import javax.swing.JOptionPane;

public class MaiorNumero {

    public static void main(String[] args) {

        int[] num = new int[10];
        int contador;
        int max = 0;
        int numerostotal = 3;

        // Pede ao usuário para digitar números
        for (contador = 0; contador < numerostotal; contador++) {

            num[contador] = Integer.parseInt(JOptionPane.showInputDialog("Entre com números
até " + numerostotal + " no total"));

            // verifica se o número digitado é maior que max
            if (( contador == 0 ) || ( num[contador] > max ))
                max = num[contador];

        }

        // Mostra o maior número.
        JOptionPane.showMessageDialog(null, "O maior número é " + max);

    }

}
```

Listagem 4.3 – MaiorNumero.java

2. Compile e execute o programa e veja o resultado.

3. Modifique o programa para achar o menor número e mostrá-lo.

## Exercício 2 – Construindo Arrays de Objetos String

1. Usando **arrays unidimensionais** de String

### 1.1. Usando arrays unidimensionais de String

1. Escreva, compile e execute a **Listagem-4.4, DiasDaSemana.java**, preste bastante atenção aos comentários.

```
public class DiasDaSemana {

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

```
// Declara e inicializa um array de String contendo os dias da semana
String[] dias = { "Domingo", "Segunda", "Terca", "Quarta", "Quinta", "Sexta", "Sabado" };

// Mostra dias da semana com laço forEach/Enhenced for Java 5.0
System.out.println("Mostra dias da Semana usando laço For-Each");
for (String str : dias)
    System.out.print(str + " ");

// Declara e inicializa um array de String contendo os dias da semana em Inglês
// veja que para cada elemento você cria um objeto do tipo usado pelo array
String[] days = new String[7];
days[0] = new String("Sunday");
days[1] = new String("Monday");
days[2] = new String("Tuesday");
days[3] = new String("Wednesday");
days[4] = new String("Thursday");
days[5] = new String("Friday");
days[6] = new String("Saturday");

// imprime a traducao de portugues-ingles dos dias da semana
System.out.println("\nPORTUGUES - INGLES");
for (int i = 0; i < dias.length; i++) {
    System.out.println(dias[i] + " - " + days[i]);
}
}
```

Listagem 4.4 – DiasDaSemana.java

3. Crie um programa como o anterior, que contenha um único **array bidimensional de String**, onde cada linha contenha a String em Português e em Inglês do dia da semana, como exemplo **dias[0][0]="Segunda", dias[0][1]=Monday** ..., e assim por diante.