

## JAVA - AULA 6

= A declaração de vetores deve ser: `int v[] = { 87, 45, 68, 12, 20, 37 };`  
= ...ou então: `int v[] = new int[10];`  
= Em Java, a declaração `int v[12];` gera um erro de sintaxe

### Iniciando um vetor (InitArray1.java)

```
public class InitArray1
{
    public static void main( String args[] )
    {
        int array[] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };

        System.out.printf( "%s%8s\n", "Indice", "Valor" );

        for ( int counter = 0; counter < array.length; counter++ )
            System.out.printf( "%6d%8d\n", counter, array[ counter ] );
    }
}
```

### Usando vetores como contadores (RollDie.java)

```
import java.util.Random;

public class RollDie
{
    public static void main( String args[] )
    {
        Random randomNumbers = new Random();
        int frequency[] = new int[ 7 ];

        for ( int roll = 1; roll <= 6000; roll++ )
            ++frequency[ 1 + randomNumbers.nextInt( 6 ) ];

        System.out.printf( "%s%11s\n", "Face", "Frequencia" );

        for ( int face = 1; face < frequency.length; face++ )
            System.out.printf( "%4d%11d\n", face, frequency[ face ] );
    }
}
```

### Passando vetores para métodos (PassArray.java)

```
public class PassArray
{
    public static void main( String args[] )
    {
        int array[] = { 1, 2, 3, 4, 5 };

        System.out.println(
            "Efeito de passar por referencia todo vetor:\n" +
            "Os valores na ordem original sao:" );

        for ( int value : array )
            System.out.printf( "    %d", value );
    }
}
```

```

    modifyArray( array );
    System.out.println( "\n\nOs valores do vetor modificado sao:" );

    for ( int value : array )
        System.out.printf( "    %d", value );

    System.out.printf(
        "\n\nEfeito de passar o valor de um elemento do vetor:\n" +
        "array[3] antes de modifyElement: %d\n", array[ 3 ] );

    modifyElement( array[ 3 ] );
    System.out.printf(
        "array[3] depois de modifyElement: %d\n", array[ 3 ] );
}

public static void modifyArray( int array2[] )
{
    for ( int counter = 0; counter < array2.length; counter++ )
        array2[ counter ] *= 2;
}

public static void modifyElement( int element )
{
    element *= 2;
    System.out.printf(
        "Valor do elemento em modifyElement: %d\n", element );
}
}

```

- = A passagem por referência altera diretamente a variável
- = A passagem por valor altera apenas a cópia da variável
- = Em vetores muito grandes a passagem por valor gasta tempo e memória

### Vetores multidimensionais (GradeBook.java)

```

public class GradeBook
{
    private String courseName;
    private int grades[][];

    public GradeBook( String name, int gradesArray[][])
    {
        courseName = name;
        grades = gradesArray;
    }

    public void setCourseName( String name )
    {
        courseName = name;
    }

    public String getCourseName()
    {
        return courseName;
    }

    public void displayMessage()

```

```

{
    System.out.printf( "Bemvindo ao curso\n%s!\n\n",
        getCourseName() );
}

public void processGrades()
{
    outputGrades();

    System.out.printf( "\n%s %d\n%s %d\n\n",
        "Menor nota no curso e", getMinimum(),
        "Maior nota no curso e", getMaximum() );

    outputBarChart();
}

public int getMinimum()
{
    int lowGrade = grades[ 0 ][ 0 ];

    for ( int studentGrades[] : grades )
    {
        for ( int grade : studentGrades )
        {
            if ( grade < lowGrade )
                lowGrade = grade;
        }
    }

    return lowGrade;
}

public int getMaximum()
{
    int highGrade = grades[ 0 ][ 0 ];

    for ( int studentGrades[] : grades )
    {
        for ( int grade : studentGrades )
        {
            if ( grade > highGrade )
                highGrade = grade;
        }
    }

    return highGrade;
}

public double getAverage( int setOfGrades[] )
{
    int total = 0;

    for ( int grade : setOfGrades )
        total += grade;

    return (double) total / setOfGrades.length;
}

```

```

public void outputBarChart()
{
    System.out.println( "Distribuicao da notas da turma:" );

    int frequency[] = new int[ 11 ];

    for ( int studentGrades[] : grades )
    {
        for ( int grade : studentGrades )
            ++frequency[ grade / 10 ];
    }

    for ( int count = 0; count < frequency.length; count++ )
    {
        if ( count == 10 )
            System.out.printf( "%5d: ", 100 );
        else
            System.out.printf( "%02d-%02d: ",
                                count * 10, count * 10 + 9 );

        for ( int stars = 0; stars < frequency[ count ]; stars++ )
            System.out.print( "*" );

        System.out.println();
    }
}

public void outputGrades()
{
    System.out.println( "As notas sao:\n" );
    System.out.print( "          " );

    for ( int test = 0; test < grades[ 0 ].length; test++ )
        System.out.printf( "Prova %d ", test + 1 );

    System.out.println( "    Media" );

    for ( int student = 0; student < grades.length; student++ )
    {
        System.out.printf( "Estudante %2d", student + 1 );

        for ( int test : grades[ student ] )
            System.out.printf( "%8d", test );

        double average = getAverage( grades[ student ] );
        System.out.printf( "%9.2f\n", average );
    }
}
}

```

### ...e chamando seus métodos (GradeBookTest.java)

```
public class GradeBookTest
{
    public static void main( String args[] )
    {
        int gradesArray[][] = { { 87, 96, 70 }, { 68, 87, 90 },
                                { 94, 100, 90 }, { 100, 81, 82 },
                                { 83, 65, 85 }, { 78, 87, 65 },
                                { 85, 75, 83 }, { 91, 94, 100 },
                                { 76, 72, 84 }, { 87, 93, 73 } };

        GradeBook myGradeBook = new GradeBook(
            "Programacao Java", gradesArray);
        myGradeBook.displayMessage();
        myGradeBook.processGrades();
    }
}
```

### Lista de argumentos de comprimento variável (VarargsTest.java)

```
public class VarargsTest
{
    public static double average(double... numbers)
    {
        double total = 0.0;

        // calcula total utilizando a instrução for aprimorada
        for ( double d : numbers )
            total += d;

        return total / numbers.length;
    }

    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( "Media de d1 e d2          = %.1f\n",
            average( d1, d2 ) );
        System.out.printf( "Media de d1, d2 e d3       = %.1f\n",
            average( d1, d2, d3 ) );
        System.out.printf( "Media de d1, d2, d3 e d4 = %.1f\n",
            average( d1, d2, d3, d4 ) );
    }
}
```

### Utilizando argumentos de linha de comando (InitArray2.java)

```
public class InitArray2
{
    public static void main(String args[])
    {
        // verifica número de argumentos de linha de comando
        if (args.length != 3)
            System.out.println(
                "Erro: Favor redigitar todo o comando, incluindo\n" +
                "o tamanho do vetor, valor inicial e incremento." );
        else
        {
            // obtém o tamanho do array a partir do primeiro
            // argumento de linha de comando
            int arrayLength = Integer.parseInt( args[ 0 ] );
            int array[] = new int[ arrayLength ]; // cria o array

            // obtém o valor inicial e o incremento a partir
            // do argumento de linha de comando
            int initialValue = Integer.parseInt( args[ 1 ] );
            int increment = Integer.parseInt( args[ 2 ] );

            // calcula valor de cada elemento do array
            for ( int counter = 0; counter < array.length; counter++ )
                array[ counter ] = initialValue + increment * counter;

            System.out.printf( "%s%8s\n", "Indice", "Valor" );

            // exibe o valor e o índice de array
            for ( int counter = 0; counter < array.length; counter++ )
                System.out.printf( "%5d%8d\n", counter, array[ counter ] );
        }
    }
}
```

= Depois de compilar, rodar em linha de comando

= Exemplo: java InitArray2 10 0 4 // lista 10 números de 4 em 4, início 0

### EXERCÍCIOS

1) Determine se cada um dos seguintes exemplos é falso ou verdadeiro. Se falso, explique por quê.

- a) Um array pode armazenar muitos tipos de valores diferentes.
- b) Um índice de array deve ser normalmente do tipo float.
- c) Argumentos de linha de comando são separados por vírgulas.

2) Escreva um aplicativo que calcule o produto de uma série de inteiros que são passados para o método product utilizando uma lista de argumentos de comprimento variável. Teste seu método com várias chamadas, cada uma com um número diferente de argumentos.