

Qualificação Profissional de Assistente de
Desenvolvimento de Sistemas

LÓGICA DE PROGRAMAÇÃO

GEEaD - Grupo de Estudos de Educação a Distância

Centro de Educação Tecnológica Paula Souza

Autores:

Eliana Cristina Nogueira Barion

Marcelo Fernando Iguchi

Paulo Henrique Mendes Carvalho

Rute Akie Utida

Revisão Técnica: Sandra

Maria Leandro

Revisão Gramatical:

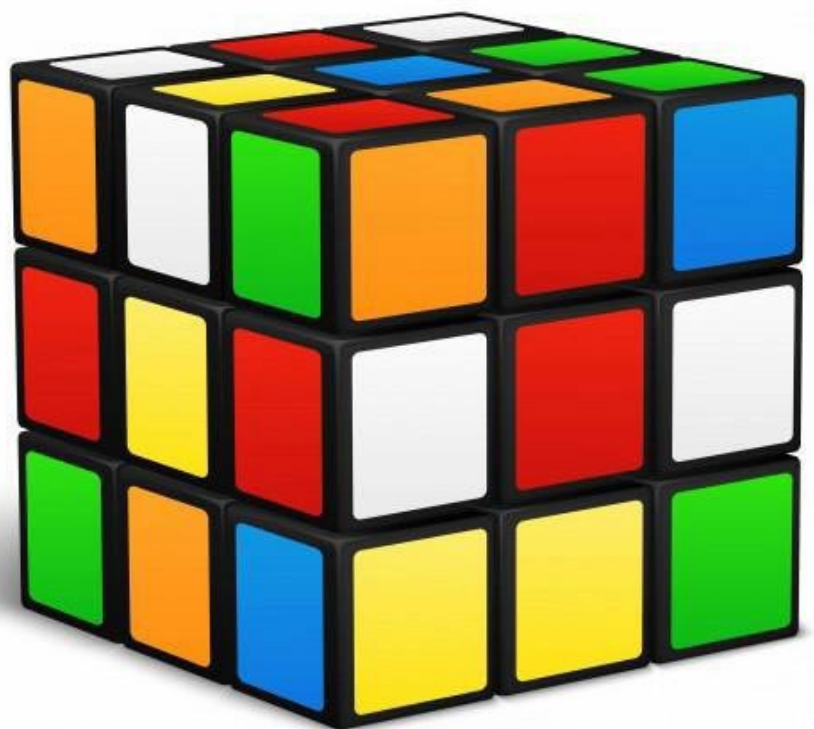
Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:

Flávio Biazim

AGENDA10

VETORES



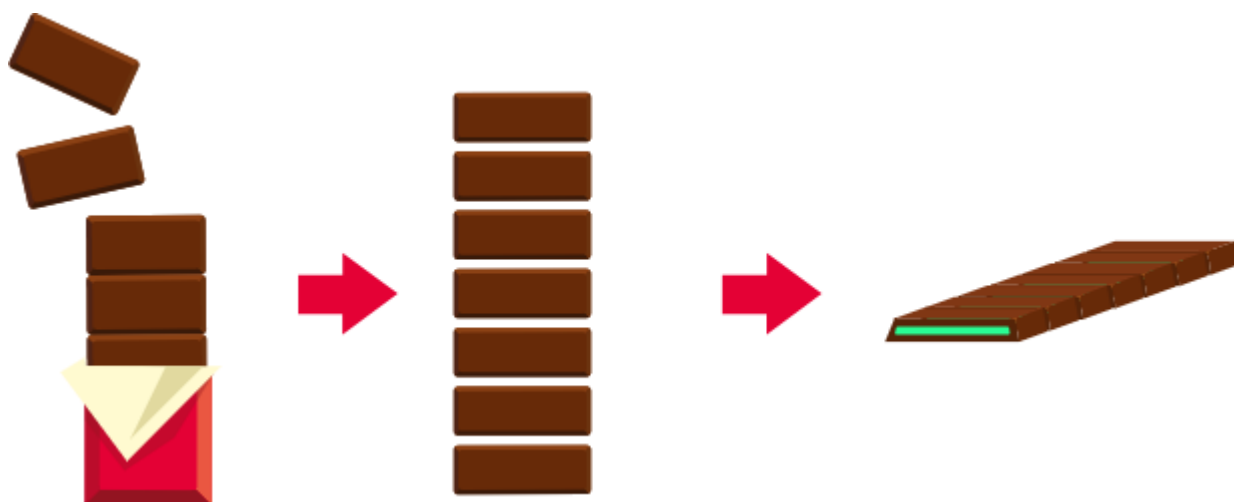


Durante seus estudos de Lógica de Programação, você já passou por alguma situação que necessite a declaração de muitas variáveis para um mesmo propósito?

Alguma vez você já se perguntou se para calcular a média entre 10 notas de um aluno seria necessário declarar 10 variáveis?

Ao longo das agendas anteriores, você trabalhou apenas com variáveis simples, que só recebem um dado por vez. Mas há uma técnica de programação que permite o **agrupamento de várias informações** dentro de uma mesma variável, fazendo com que a programação seja mais rápida e eficiente, que são chamados de **vetores**.

Para fazermos uma analogia gastronômica, vetores são como barras de chocolate: uma barra é composta por diversos quadradinhos, que podem armazenar recheios, desde que sejam todos do mesmo sabor.



Assim como uma única barra de chocolate armazena quantidades de recheio em unidades menores, **um único vetor é capaz de armazenar diversos valores em suas casas (espaços)**. Compramos barras de chocolate em vez de bombons individuais pela praticidade: adquirimos uma quantidade maior do produto levando em um único item (uma barra). Além disso, a barra costuma ser mais econômica que o bombom individual. Na programação, utilizamos vetores em vez de variáveis convencionais também pela praticidade: **isso facilita o acesso aos dados e economiza espaço na memória do computador**.

Vetores são variáveis especiais que têm a capacidade de receberem diversos valores, desde que todos sejam do mesmo tipo (int, double, String etc). São também conhecidos como **matrizes unidimensionais** (apresenta uma única dimensão, uma única coluna ou uma única linha depende da forma que você olha o vetor).

Na videoaula a seguir, o professor Sandro Valérius apresenta um exemplo de programa com vetores que também utiliza o laço de repetição for e faz outras analogias bastante interessantes para te ajudar a entender melhor os conceitos de vetores. **Não deixe de assistir!**



Vetores foram desenvolvidos para que leituras e gravações de dados repetitivos sejam simplificados por meio das **estruturas de repetição**, uma vez que valores individuais devem sempre ser lidos individualmente, não sendo possível ler N variáveis dentro de uma única estrutura de repetição.

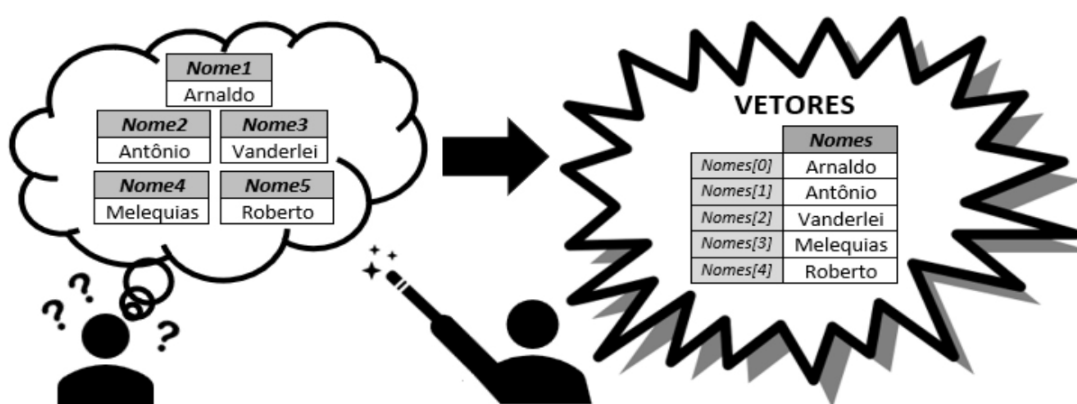


Imagem 01: GEEaD – O primeiro balão representa 5 variáveis que recebem valores com nomes de pessoas individualmente e o segundo balão representa uma variável do tipo vetor com contendo os 5 nomes nas posições de 0 à 4.



Agora que já vimos que vetores são variáveis que podem receber mais de um valor, em quais casos já estudados podemos substituir variáveis repetidas por vetores? Reflita sobre este novo conceito antes de prosseguir com a leitura.

Conhecendo a estrutura do vetor

Para entender melhor a importância deste tipo de variável, vamos resolver a seguinte situação problema:



Jennifer, uma professora do Curso Técnico em Eventos e formada em Programação de Computadores, decidiu criar um programa que confeccione automaticamente uma lista de convidados para o próximo evento da turma do 1º Módulo do Curso. Para testar se o programa funcionará corretamente, Jennifer deseja primeiramente cadastrar apenas 10 convidados e exibí-los na mesma ordem na qual foram cadastrados.

Para isso, foi criado o seguinte programa em Java:


```

1  import javax.swing.JOptionPane;
2  public class Principal {
3      public static void main(String[] args) {
4          String convidado1;
5          String convidado2;
6          String convidado3;
7          String convidado4;
8          String convidado5;
9          String convidado6;
10         String convidado7;
11         String convidado8;
12         String convidado9;
13         String convidado10;
14         String listafinal;
15
16         convidado1 = JOptionPane.showInputDialog("Digite o nome do 1º Convidado");
17         convidado2 = JOptionPane.showInputDialog("Digite o nome do 2º Convidado");
18         convidado3 = JOptionPane.showInputDialog("Digite o nome do 3º Convidado");
19         convidado4 = JOptionPane.showInputDialog("Digite o nome do 4º Convidado");
20         convidado5 = JOptionPane.showInputDialog("Digite o nome do 5º Convidado");
21         convidado6 = JOptionPane.showInputDialog("Digite o nome do 6º Convidado");
22         convidado7 = JOptionPane.showInputDialog("Digite o nome do 7º Convidado");
23         convidado8 = JOptionPane.showInputDialog("Digite o nome do 8º Convidado");
24         convidado9 = JOptionPane.showInputDialog("Digite o nome do 9º Convidado");
25         convidado10 = JOptionPane.showInputDialog("Digite o nome do 10º Convidado");
26         listafinal = "Os convidados da festa são: " + convidado1 + ", " + convidado2 + ", " + convidado3 + ", "
27                     + convidado4 + ", " + convidado5 + ", " + convidado6 + ", " + convidado7 + ", " + convidado8 + ", "
28                     + convidado9 + ", " + convidado10;
29
30         JOptionPane.showMessageDialog(null, listafinal);
31     }
32 }

```



Apesar do programa funcionar sem problemas, a codificação ficou muito repetitiva, não acha?


Isto ocorre porque o programa está fazendo apenas uma função básica, sem muitos incrementos.

Com vetores, podemos reorganizar este código para utilizar **menos da metade** dos comandos, e com uma vantagem: não é necessário acrescentar nenhuma linha de código extra, caso seja necessário cadastrar 10, 100 ou até 1000 convidados.

Como trabalhar com um Vetor

Em primeiro lugar, para trabalhar com um vetor, devemos **declará-lo**, assim como fazemos com uma variável. A diferença é que na declaração devemos especificar a quantidade de “casas” (espaços para armazenamento das variáveis) o vetor terá.

Para o computador, cada casa do vetor é um espaço onde poderá ser escrito um valor. A primeira casa do vetor, sempre será a casa de número zero devido à numeração binária utilizada em qualquer meio computacional, onde o **primeiro número é zero**.



Vale lembrar que assim como uma variável, você nunca deve declarar um vetor sem necessidade. Um problema comum para quem está iniciando os estudos com vetores é a declaração de mais casas no vetor do que realmente necessita, ou o estouro do vetor, que é o fato de tentar alocar mais valores no vetor do que ele suporta. Tome cuidado!

Declarando um vetor

Para declarar um vetor é muito simples, basta seguir o seu padrão conforme a tabela abaixo:

PORTUGOL	JAVA
tipo nome[capacidade]	tipo[] nomevariavel = new tipo[10]

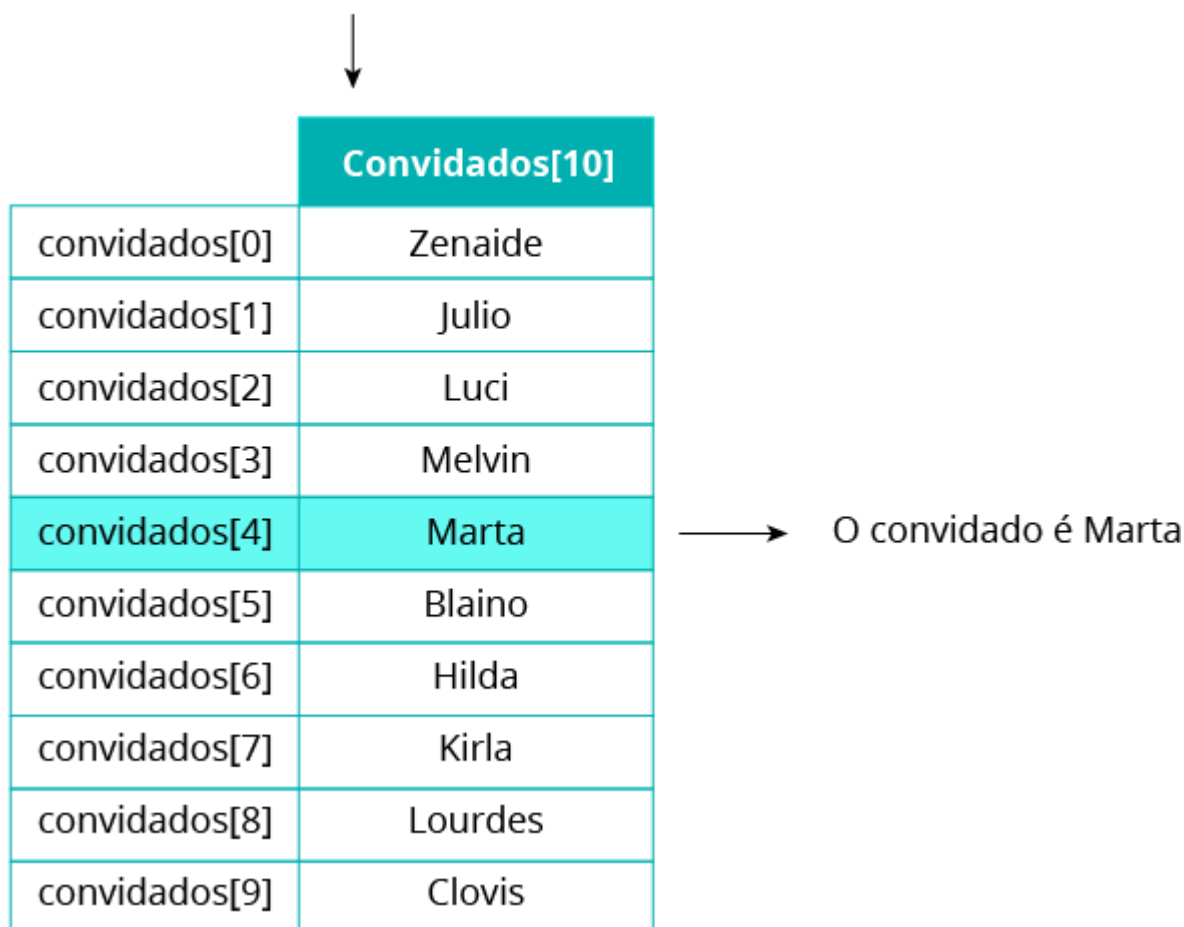
Praticando a Lógica com Vetores

As diferenças entre uma variável comum e um vetor são:

- O vetor sempre deve estar acompanhado da “casa” em que o dado deve ser lido/gravado;
- A primeira casa do vetor sempre será nomeada como casa número 0.

Na prática, se declaramos um vetor de 10 casas em Java, ele alocará na memória as casas numeradas de 0 a 9. Veja o exemplo abaixo, parte da solução do programa de Jennifer:


```
System.out.println( "O convidado é " + convidados [4]);
```



Convidados[10]	
convidados[0]	Zenaide
convidados[1]	Julio
convidados[2]	Luci
convidados[3]	Melvin
convidados[4]	Marta
convidados[5]	Blaino
convidados[6]	Hilda
convidados[7]	Kirla
convidados[8]	Lourdes
convidados[9]	Clovis

→ O convidado é Marta

Imagem 02: Tabela mostrando o armazenamento de 10 convidados, por meio da variável convidado, iniciando pelo índice 0 até o índice 9. O nome Zenaide está armazenado na variável convidados[0], o nome Júlio está armazenado na variável convidados[1] e assim sucessivamente até o décimo nome que está armazenado na variável convidados[9]. A linha de código: `system.out.println("O convidado é " + convidados[4]);` retorna ao valor Marta, que pela tabela está armazenado em convidados[4].

Agora assista à videoaula do Professor Sandro Valérius, explicando o uso do vetor para separar os convidados de uma festa entre os lados direito e esquerdo do salão.



Na videoaula a seguir, o professor Rogério Silva apresenta a estrutura de vetores aplicada em Java.



Conhecendo alguns exemplos...

Voltando ao exemplo, que tal verificar como ficará o problema da Jennifer utilizando vetores?

Seguindo as particularidades do algoritmo e da Linguagem de Programação Java, veja, a seguir, a solução para o problema da professora Jennifer utilizando vetor:

PORTUGOL

```
programa
{
    funcao inicio()
    {
        cadeia convidados[10]
        inteiro contador
        cadeia listafinal

        listafinal = "Os convidados são: "

        // Entrada dos nomes dos convidados
        para (contador = 0; contador < 10; contador++)
        {
            escreva("Digite o convidado número ", contador + 1, ": ")
            leia(convidados[contador])
        }

        // Montagem da lista final
        para (contador = 0; contador < 10; contador++)
        {
            listafinal = listafinal + convidados[contador] + ", "
        }

        escreva(listafinal)
    }
}
```

Em Java:

```

1  import javax.swing.JOptionPane;
2
3  public class PrincipalVetor {
4
5      public static void main(String[] args) {
6          String[] convidados = new String[10];
7          int contador;
8          //Definir a frase que antecede o nome dos convidados
9          String listafinal = "Os convidados da festa são: ";
10         //Laço de repetição para ler cada um dos convidados.
11         for (contador = 0; contador < 10; contador++) {
12             convidados[contador] = JOptionPane.showInputDialog("Digite o " + (contador + 1) + " convidado");
13             //Como o vetor inicia-se com zero, utilizamos contador+1 para identificar ao usuário qual
14             //é o número do convidado no qual ele está digitando corretamente.
15         }
16         //laço de repetição para alocar em uma variável o nome de cada um dos candidatos
17         for (contador = 0; contador < 10; contador++) {
18             //Alocação dos dados na variável, adicionando novos valores a cada passagem do laço.
19             listafinal = listafinal + convidados[contador] + " ";
20         }
21         //exibição dos dados armazenados em listafinal.
22         JOptionPane.showMessageDialog(null, listafinal);
23     }
24 }
25

```



- Você percebeu o quanto o código fonte diminuiu de tamanho após a aplicação do vetor? Um vetor pode ser um pouquinho mais trabalhoso para ser aplicado no início, mas ele consegue fazer o trabalho de muitas variáveis utilizando apenas uma! Este é um ótimo aliado para todo o bom programador.*



VOCÊ NO COMANDO

1. Observe o exemplo de declaração de vetor a seguir e, em seguida, declare o vetor para as situações solicitadas:

a) Conforme a lista de convidados de Jennifer, vamos declarar o vetor correspondente?

Portugol	Java
Cadeia convidados [10]	String[] convidados = new String[10];

b) declarar um vetor para possibilitar o cálculo de uma média aritmética simples com base em 15 notas.

Portugol	Java

c) declarar um vetor com 4 senhas diferentes para ser lidas e comparadas posteriormente.

Portugol	Java

2. Resolva as seguintes situações problemas com auxílio de Vetores:

a) Luana necessita de um programa para o cálculo das médias dos alunos de sua escola e música. Para cada módulo do curso, o aluno efetua 5 atividades. Caso a média destas atividades seja maior que 5, o aluno poderá passar para o próximo módulo. desenvolva um programa em Java utilizando Vetores que calcule a média para os alunos de Luana corretamente.

b) Desenvolva um programa que leia 20 valores inteiros. Após a leitura, o programa deverá exibir os números armazenados no vetor, porém, os números que estiverem em casas pares do vetor, o valor deverá ser multiplicado por 2.

c) Desenvolva um programa que leia 5 valores de um vetor e os exiba na ordem inversa da qual foram cadastrados.

Confira as respostas:

1.

b)

Portugol	Java
Inteiro media [15]	int[] media = new int[15];

c)

Portugol	Java
cadeia convidados [4]	String[] convidados = new String[4];

2.

a) Portugol

```

programa
{
    funcao inicio()
    {
        real nota[5]
        real soma = 0
        real media
        inteiro contador

        para (contador = 0; contador < 5; contador++)
        {
            escreva("Digite a nota ", contador + 1, ": ")
            leia(nota[contador])
        }

        // Soma das notas
        para (contador = 0; contador < 5; contador++)
        {
            soma = soma + nota[contador]
        }
        media = soma / 5

        se (media >= 5)
        {
            escreva("O aluno tirou ", media, ", portanto está
APROVADO")
        }
        senao
        {
            escreva("O aluno tirou ", media, ", portanto está
REPROVADO")
        }
    }
}

```


No Java:

```

1
2 import javax.swing.JOptionPane;
3
4 public class Principal {
5     public static void main(String args[]) {
6         double nota[] = new double[5];
7         double soma = 0;
8         double media;
9         int contador;
10        //laço para ler as notas do aluno
11        for (contador = 0; contador < 5; contador++) {
12            nota[contador] = Double.parseDouble(JOptionPane.showInputDialog("Digite a " + (contador+1) + " nota do aluno.));
13        }
14        //laço para somar as notas dos alunos
15        for(contador = 0; contador <5; contador++) {
16            soma = soma + nota[contador];
17        }
18        media = soma / 5;
19        if (media >= 5 ) {
20            JOptionPane.showMessageDialog(null, "O aluno tirou " + media + ", portando está APROVADO");
21        }
22        else{
23            JOptionPane.showMessageDialog(null, "O aluno tirou " + media + ", portando está REPROVADO");
24        }
25    }
26 }
27

```

Assista à videoaula do professor Sandro Valérius para complementar seus estudos:



b) Portugal

```

programa
{
    funcao inicio()
    {
        inteiro valor[20]
        inteiro contador, mult
        cadeia numeros
        numeros = "Os números digitados foram: "

        para (contador = 0; contador < 20; contador++){
            escreva("Digite o número ", contador + 1, ": ")
            leia(valor[contador])
        }
        para (contador = 0; contador < 20; contador++){
            se (valor[contador] % 2 == 0){
                mult = valor[contador] * 2
                numeros = numeros + mult + " "
            }senao{
                numeros = numeros + valor[contador] + " "
            }
        }

        escreva(numeros)
    }
}

```

No Java:

```

1
2 ☐ import javax.swing.JOptionPane;
3
4 public class Principal {
5     ☐ public static void main(String args[]) {
6         int valor[] = new int[20];
7         int contador;
8         String numeros;
9         int mult;
10
11         for (contador = 0; contador < 20; contador++) {
12             valor[contador] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + (contador + 1) + " valor"));
13         }
14         numeros = "Os números digitados foram: ";
15         for (contador = 0; contador < 20; contador++) {
16             if(valor[contador] % 2 == 0) {
17                 mult = valor[contador] * 2;
18                 numeros = numeros + String.valueOf(mult) + " ";
19             }
20             else {
21                 numeros = numeros + String.valueOf(valor[contador]) + " ";
22             }
23         }
24         JOptionPane.showMessageDialog(null, numeros);
25     }
26 }

```

c) Portugal

```

programa
{
    funcao inicio()
    {
        inteiro numeros[5]
        inteiro contador
        cadeia resultado

        resultado = "Os números em ordem inversa são: "

        para (contador = 0; contador < 5; contador++){
            escreva("Digite o ", contador + 1, "º número: ")
            leia(numeros[contador])
        }

        para (contador = 4; contador >= 0; contador--){
            resultado = resultado + numeros[contador] + " "
        }

        escreva(resultado)
    }
}

```

No Java:

```

1
2 import javax.swing.JOptionPane;
3
4 public class Principal {
5     public static void main(String[] args) {
6         int[] numeros = new int[5];
7         int contador;
8         String resultado;
9         resultado = "Os números em ordem inversa da digitada são: ";
10        for (contador = 0; contador < 5; contador++) {
11            numeros[contador] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + (contador + 1) + "º número"));
12        }
13
14        for (contador = 4; contador >= 0; contador--) {
15            resultado = resultado + String.valueOf(numeros[contador]) + " ";
16        }
17        JOptionPane.showMessageDialog(null, resultado);
18    }
19 }
20
21
22
23
24

```



Como você já sabe, para cada programa e para cada código fonte gerado, sempre há mais de uma forma de se estruturar o programa para receber os mesmos resultados ao final. Ao se tratar de Vetores, não é diferente. Nesta aula, vimos a sua forma básica, abordando apenas uma dimensão. Dependendo da circunstância na qual se encontra o problema, podemos utilizar também as Matrizes Bidimensionais e Heterogêneas para resolver nossos problemas computacionais, basta saber utilizá-los adequadamente.