

Laboratório de Desenvolvimento de Algoritmos



- ✓ Vetores e Matrizes

Desafio!

1- Faça um programa que escreva “Parabéns!” nas melhores provas de uma disciplina com 3 alunos. O programa deve:

- ✓ Ler os nomes e as notas de 3 alunos
- ✓ Calcular a média da turma
- ✓ Listar os alunos tiveram nota acima da média



Desafio!

```
import javax.swing.JOptionPane;

public class ExemploAlunos {
    public static void main(String[] args) {
        float nota1, nota2, nota3, media;
        String nome1, nome2, nome3;
        nome1 = JOptionPane.showInputDialog(null, "Digite o nome do 1º aluno:");
        nota1 = Float.parseFloat(JOptionPane.showInputDialog(null, "Digite a nota do aluno " + nome1));

        nome2 = JOptionPane.showInputDialog(null, "Digite o nome do 2º aluno:");
        nota2 = Float.parseFloat(JOptionPane.showInputDialog(null, "Digite a nota do aluno " + nome2));

        nome3 = JOptionPane.showInputDialog(null, "Digite o nome do 3º aluno:");
        nota3 = Float.parseFloat(JOptionPane.showInputDialog(null, "Digite a nota do aluno " + nome3));

        media = (nota1+nota2+nota3)/3;
        System.out.println("A média da turma foi: " + media);
        if(nota1>media)
            System.out.println("Parabéns " + nome1);
        if(nota2>media)
            System.out.println("Parabéns " + nome2);
        if(nota3>media)
            System.out.println("Parabéns " + nome3);
    }
}
```

Desafio!

```
import javax.swing.JOptionPane;

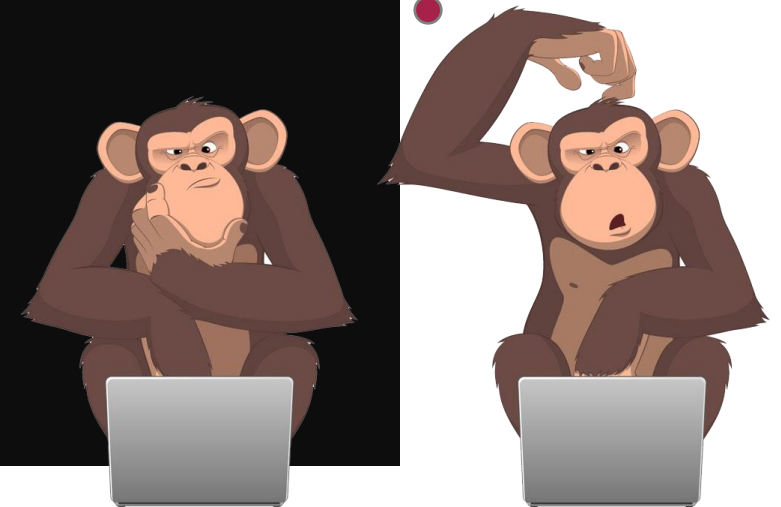
public class ExemploAlunos {
    public static void main(String[] args) {
        float nota1, nota2, nota3, media;
        String nome1, nome2, nome3;
        nome1 = JOptionPane.showInputDialog(null, "Digite o nome do 1º aluno:");
        nota1 = Float.parseFloat(JOptionPane.showInputDialog(null, "Digite a nota do aluno " + nome1));

        nome2 = JOptionPane.showInputDialog(null, "Digite o nome do 2º aluno:");
        nota2 = Float.parseFloat(JOptionPane.showInputDialog(null, "Digite a nota do aluno " + nome2));

        nome3 = JOptionPane.showInputDialog(null, "Digite o nome do 3º aluno:");
        nota3 = Float.parseFloat(JOptionPane.showInputDialog(null, "Digite a nota do aluno " + nome3));

        media = (nota1+nota2+nota3)/3;
        System.out.println("A média da turma foi: " + media);
        if(nota1>media)
            System.out.println("Parabéns " + nome1);
        if(nota2>media)
            System.out.println("Parabéns " + nome2);
        if(nota3>media)
            System.out.println("Parabéns " + nome3);
    }
}
```

Mas, e se
fossem 40
alunos???



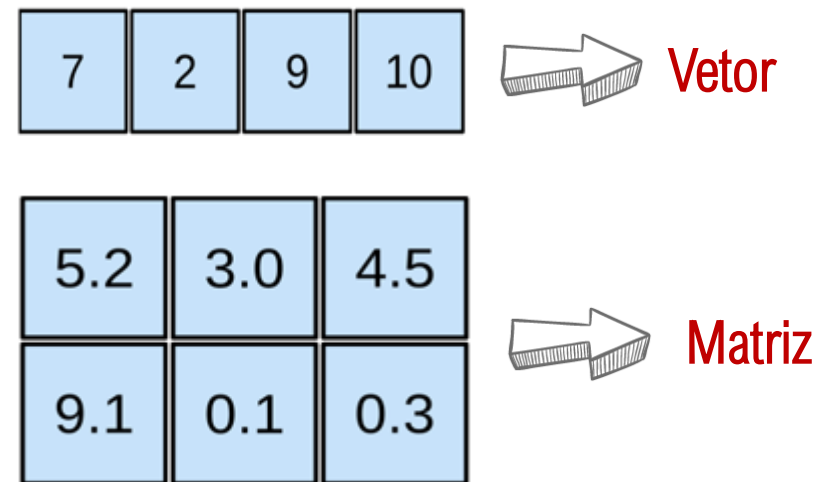
Estruturas de dados

- ✔ Uma variável é capaz de armazenar apenas um valor de cada vez. Existem situações em que há necessidade de armazenar uma grande quantidade de valores e para isso não iremos declarar várias variáveis.
- ✔ Para resolver esse problema construímos novos tipos que têm um formato denominado estrutura de dados, que define como os tipos primitivos estão organizados.



Arranjos

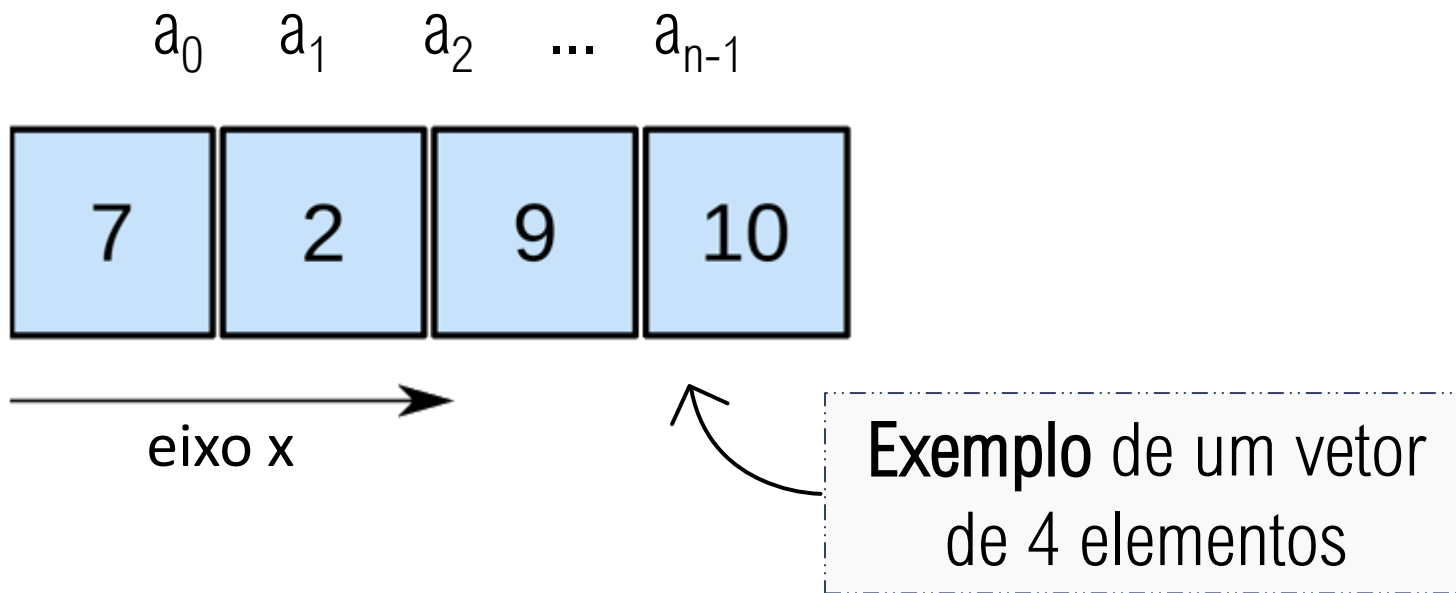
- ✓ Um arranjo é um grupo/conjunto de elementos ou itens que respondem a um mesmo nome e que podem ser acessados segundo a posição (índice) que ocupam dentro do arranjo.
- ✓ Cada elemento ou item de um arranjo está em determinada posição e armazena um valor. Este valor poderá ser real, inteiro, lógico, literal (float, int, boolean, char, String etc. no caso da linguagem Java).
- ✓ Os arranjos podem ser:
 - **Unidimensionais:** conhecidos como **vetores**, possuem somente um índice.
 - **Bidimensionais:** conhecidos como **matrizes**, possuem dois ou mais índices.



Arranjos

- ✓ Os arranjos podem ser:
 - **Unidimensionais:** conhecidos como **vetores**, possuem somente um índice.
 - **Bidimensionais:** conhecidos como **matrizes**, possuem dois ou mais índices.

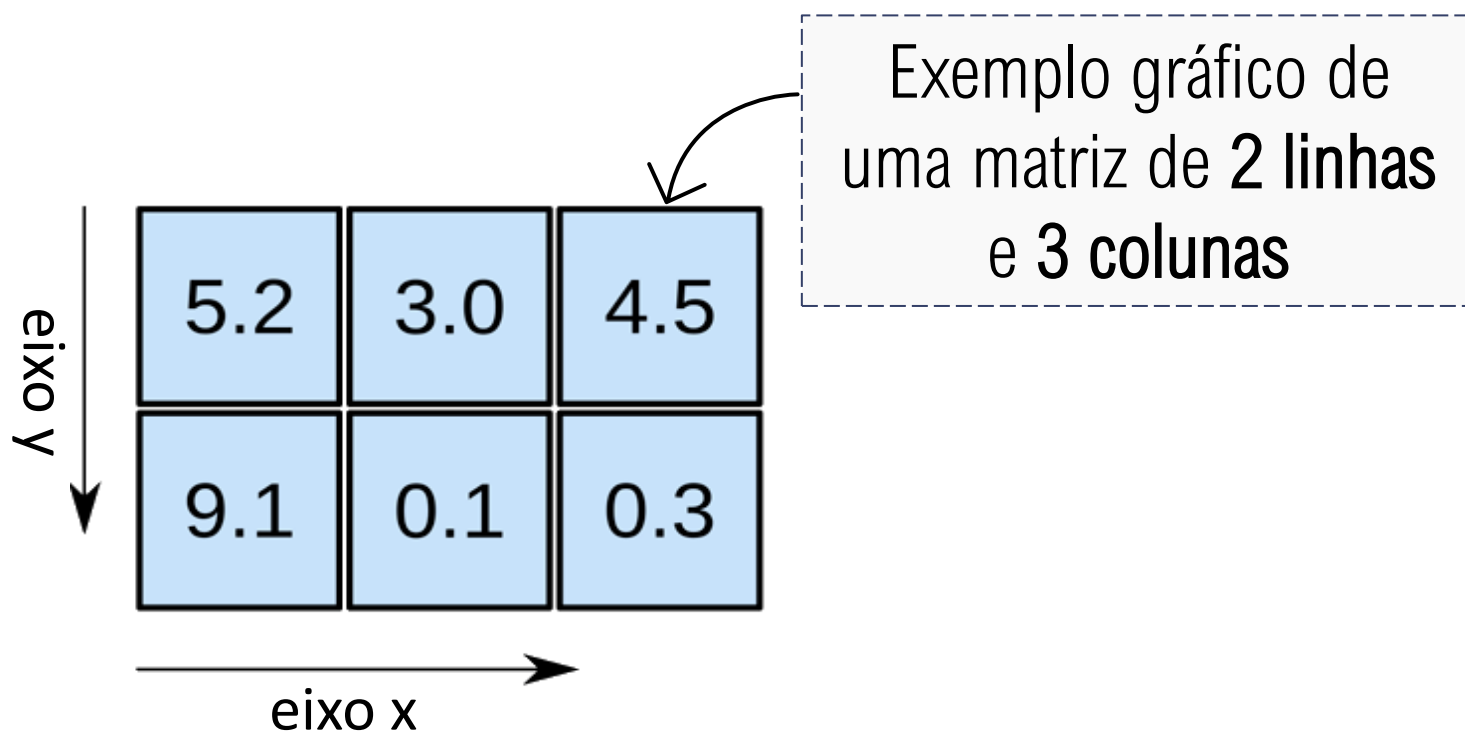
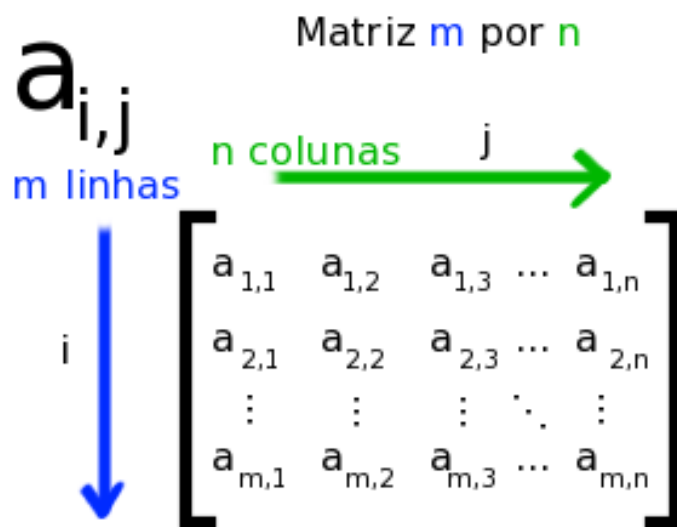
Vetor (de n elementos):



Arranjos

- ✔ Os arranjos podem ser:
 - **Unidimensionais:** conhecidos como **vetores**, possuem somente um índice.
 - **Bidimensionais:** conhecidos como **matrizes**, possuem dois ou mais índices.

Matriz (de m linhas e n colunas):






Arranjos unidimensionais: vetores

- ✔ São utilizados para armazenar um conjunto de dados cujos elementos podem ser endereçados por um único índice.
- ✔ Os vetores são coleções de objetos ou tipos de dados primitivos, que têm como características:
 - Tamanho Fixo: Vetores não podem ser redimensionados após sua construção, seria necessário criar um novo vetor e copiar os valores do antigo.
 - Verificados em tempo de execução: uma tentativa de acessar índices inexistentes provoca, na execução, um erro do tipo ArrayIndexOutOfBoundsException.
 - Tipo Definido: deve-se restringir o tipo dos elementos que podem ser armazenados.

Arranjos unidimensionais: vetores

- ✓ O acesso às posições é feito colocando o nome de identificação e o número da posição (índice);
- ✓ A posição do primeiro elemento de um vetor é sempre 0 (zero);
- ✓ O exemplo abaixo ilustra um vetor de nome "c", com nove elementos:

Posição do elemento no arranjo						Nome do arranjo		
								
c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]
10	20	30	40	50	60	70	80	90
								
Conteúdo de uma posição (um elemento) do arranjo								



Arranjos unidimensionais: vetores

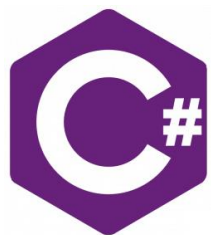
A posição do primeiro elemento de um vetor é sempre 0 (zero)



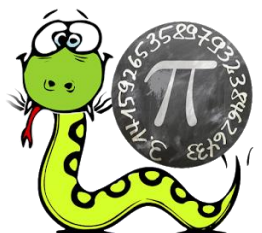
Declaração de vetores



```
tipo nome[quantidade]
```



```
tipo nome[] = new tipo[quantidade]
```



```
nome = []
```

Declaração de vetores em Java

Não podemos utilizar um vetor antes da sua declaração e inicialização:

- ✓ Declaração de um vetor:

```
tipo[] nomeVar;    ou    tipo nomeVar[]; //tanto faz
```

- ✓ Inicialização de um vetor:

```
nomeVar = new mesmotipo[quantidade];
```

- ✓ Pode-se criar diretamente, na mesma linha:

```
tipo[] nomeVar = new mesmotipo[quantidade]; //mais prático
```

- ✓ Ou ainda, podemos criar um vetor já com elementos, utilizando chaves:

```
int[] numeros = {1, 2, 3, 4, 5};  
String[] nomes = {"Ana Paula", "Cristiane", "Leonardo", "Alcides"};
```



Propriedade

- ✓ Os elementos de um vetor podem ser recuperados/modificados a partir do índice:

```
public static void main(String[] args) {  
    int idade[] = {18, 24, 36, 48};  
    System.out.println("Idade na posição 0 = " + idade[0]);  
    idade[1] = 22;  
    System.out.println("Idade na posição 1 = " + idade[1]);  
}
```

- ✓ Cada elemento do vetor é inicializado com um valor padrão, depende do tipo de dados:
 - **null** para Objetos (como Strings, etc.)
 - **0** para números (int, float, double, byte, etc.)
 - **false** para booleanos (boolean)



Vetores: atribuição de valores

- ✓ A utilização de um vetor está associado a uma estrutura de repetição.
- ✓ Com isso podemos facilmente percorrer um vetor para consultas ou atualizações.



```
algoritmo exemplo
    inteiro c[1000]
    início
        //Preenchendo o vetor c
        para(i=0; i<1000; i++){
            escreva("Digite um número")
            leia(c[i])
        }
    fim
```

Vetores: atribuição de valores

- ✓ A utilização de um vetor está associado a uma estrutura de repetição.
- ✓ Com isso podemos facilmente percorrer um vetor para consultas ou atualizações.



```
import javax.swing.JOptionPane;

public class Exemplo{
    public static void main(String[] args) {
        int c[] = new int[1000];
        //Preenchendo o vetor c
        for (int i=0; i<1000; i++){
            c[i]=Integer.parseInt(JOptionPane.showInputDialog(null,
                                                                "Digite um número: "));
        }
    }
}
```


Exemplos

1- Faça um programa que escreva “Parabéns!” nas melhores provas de uma disciplina com 3 alunos. O programa deve:

- ✓ Ler os nomes e as notas de 3 alunos
- ✓ Calcular a média da turma
- ✓ Listar os alunos tiveram nota acima da média



Desafio!

```
import javax.swing.JOptionPane;
public class ExemploAlunos {
    public static void main(String[] args) {
        float nota[] = new float[3];
        String nome[] = new String[3];
        float media, soma = 0;
        for(int i=0; i<3; i++){
            nome[i] = JOptionPane.showInputDialog(null, "Digite o nome do " + (i+1) + "º aluno:");
            nota[i] = Float.parseFloat(JOptionPane.showInputDialog(null,
                "Digite a nota do aluno " + nome[i]));
            soma += nota[i];
        }
        media = soma/3;
        System.out.println("A média da turma foi: " + media);
        for(int i=0; i<3; i++){
            if(nota[i]>media)
                System.out.println("Parabéns " + nome[i]);
        }
    }
}
```



A propriedade length

- ✓ Todo vetor em Java possui esta propriedade, que informa o número de elementos que possui. **Exemplo:**

```
public class Exemplo{  
    public static void main(String[] args) {  
        int idade[] = {18, 24, 36, 13};  
        //Utilização da propriedade length  
        for (int i=0; i<idade.length; i++){  
            System.out.println(idade[i]);  
        }  
    }  
}
```



Exemplo da propriedade length

```
public class LengthDemo {  
    public static void main(String[] args) {  
        int[] lista = new int[10];  
        int[] numeros = {1, 2, 3, 4};  
        int[][] tabela = { //tabela de tamanho variável  
            {1, 2, 3}, {4, 5}, {6, 7, 8, 9}};  
        System.out.println("Tamanho da lista: " + lista.length);  
        System.out.println("Tamanho de números: " + numeros.length);  
        //Tamanho da tabela nas linhas 0, 1 e 2  
        System.out.println("Tamanho da tabela[0]: " + tabela[0].length);  
        System.out.println("Tamanho da tabela[1]: " + tabela[1].length);  
        System.out.println("Tamanho da tabela[2]: " + tabela[2].length);  
    }  
}
```



```
TERMINAL  CONSOLE DE DEPURACÃO  PROBLEMAS  3  SAÍDA  
  
Tamanho da lista: 10  
Tamanho de números: 4  
Tamanho da tabela[0]: 3  
Tamanho da tabela[1]: 2  
Tamanho da tabela[2]: 4
```

Exemplos

2- Elabore um programa em Java que leia os salários de 10 trabalhadores de uma empresa e os armazene dentro de um vetor. Depois, em ciclos independentes:

- Calcule a média desses salários.
- Determine o maior dos salários desta empresa.
- Conte os salários menores que R\$850,00.

Exemplos

```
import javax.swing.JOptionPane;

public class ExemploEmpresa {
    public static void main(String[] args) {
        float[] salario = new float[10];
        float soma=0, media, maiorSal;
        int i, qte=0;
        for(i=0; i<10; i++){
            salario[i] = Float.parseFloat(JOptionPane.showInputDialog(null,
                                                                    "Digite o salário do " + (i+1) + "º funcionário"));
        }
        for(i=0; i<10; i++){
            soma += salario[i];
        }
    }
}
```

Exemplos

```
    for(i=0; i<10; i++){
        soma += salario[i];
    }
    media = soma/10;
    System.out.println("A média dos salários é: " + media);
    maiorSal = salario[0];
    for(i=0; i<10; i++){
        if(salario[i]>maiorSal)
            maiorSal = salario[i];
    }
    System.out.println("O maior salário é: " + maiorSal);
    for(i=0; i<10; i++){
        if(salario[i]<850)
            qte++;
    }
    System.out.println("A quantidade de salários menores que 850 é: " + qte);
}
}
```

Arranjos bidimensionais: Matrizes

- ✔ Seguindo o conceito de vetores, matrizes também têm uma declaração e uma inicialização (um pouco mais complexa).
- ✔ Quando inicializada, uma matriz também tem seus elementos inicializados com valores padrão.

		Posição do livro				
		0	1	2	...	n-1
Prateleira	0	788	598	265	...	156
	1	145	258	369	...	196
	2	989	565	345	...	526
	⋮	⋮	⋮	⋮	⋮	⋮
	m-1	845	153	564	892	210



Arranjos bidimensionais: Matrizes

- ✓ A diferença na declaração é que utilizamos mais um par de colchetes:

```
int[][] matriz ou int matriz[][]; //tanto faz
```

- ✓ Na inicialização, podemos colocar as duas dimensões:

```
matriz = new int[2][3]; //matriz com 2 linhas e 3 colunas
```

- ✓ Mas, da mesma forma que vetores, também temos “elementos” de matrizes para já inicializá-la com os valores pré-definidos, como se fosse um “vetor de vetores”:

```
int[][] matriz = {{1, 2, 3}, {4, 5, 6}};
```



Arranjos bidimensionais: Matrizes

```
public class Matrizes {  
    public static void main(String[] args) {  
        //Declaração da matriz  
        String[][] nome;  
        //Declaração do tamanho: 5 linhas x 2 colunas  
        nome = new String[5][2];  
        //Preenchimento da matriz  
        nome[0][0] = "Leonardo";  
        nome[0][1] = "Pitágoras";  
        nome[1][0] = "Ana";  
        nome[1][1] = "Paula";  
    }  
}
```



	0	1
0	"Leonardo"	"Pitágoras"
1	"Ana"	"Paula"
2	null	null
3	null	null
4	null	null

Arranjos bidimensionais: Matrizes

- ✓ No preenchimento e processamento de uma **matriz** (todos seus elementos ou alguma parte bidimensional da matriz) é frequente utilizar um ciclo duplo, que permita percorrer as duas dimensões.

```
algoritmo matriz
    inteiro num[3][3]
    início
        para(inteiro i=0; i<3; i++){
            para(inteiro j=0; j<3; j++){
                escreva("Digite um número: ")
                //faz leitura e armazenamento na posição i, j
                leia(num[i][j])
            }
        }
    fim
```



Arranjos bidimensionais: Matrizes

- ✓ No preenchimento e processamento de uma **matriz** (todos seus elementos ou alguma parte bidimensional da matriz) é frequente utilizar um ciclo duplo, que permita percorrer as duas dimensões.

```
import javax.swing.JOptionPane;

public class ExemploMatriz {
    public static void main(String[] args) {
        int[][] num = new int[3][3];
        for(int i=0; i<3; i++){
            for(int j=0; j<3; j++){
                num[i][j]= Integer.parseInt(
                    JOptionPane.showInputDialog(null, "Digite um número: "));
            }
        }
    }
}
```



Parâmetros por valor e por referência



Parâmetro por valor: um valor será enviado para o método. O valor poderá ser alterado, mas não afetará a variável utilizada na chamada.



Parâmetro por referência: neste caso, será enviada para o método uma referência (o endereço) da variável utilizada na chamada. Se o método alterar o valor deste parâmetro, o valor da variável utilizada na chamada também será modificado.

- ✓ Cada linguagem de programação estabelece como considerará os parâmetros por valor e por referência. Exemplo: Visual Basic utiliza ByVal e ByRef.
- ✓ Em Java e outras linguagens de programação, os vetores e matrizes são considerados sempre como parâmetros por referência.

Parâmetros por referência

```
import javax.swing.JOptionPane;

public class ParametroPorReferencia {
    public static void main(String[] args) {
        int[] num = new int[3];
        for(int i=0; i<3; i++){
            num[i] = Integer.parseInt(JOptionPane.showInputDialog(null,
                                                                    "Digite o " + (i+1) + "º número"));
        }
        multiplica(num);
        System.out.println("\nValores após a chamada do método:");
        for(int i=0; i<3; i++)
            System.out.println("Numero: " + num[i]);
    }
    public static void multiplica(int num[]){
        for(int i=0; i<3; i++)
            num[i] = num[i] * 2;
    }
}
```

O método recebe a referência do vetor

Os elementos do vetor foram alterados

... e multiplica cada elemento por 2

Parâmetros por referência

```
import javax.swing.JOptionPane;

public class ParametroPorReferencia {
    public static void main(String[] args) {
        int[] num = new int[3];
        for(int i=0; i<3; i++){
            num[i] = Integer.parseInt(JOptionPane.showInputDialog(null,
                "Digite o " + (i+1) + "º número"));
        }
        multiplica(num);
        System.out.println("\nValores após a chamada do método:");
        for(int i=0; i<3; i++)
            System.out.println("Numero: " + num[i]);
    }
    public static void multiplica(int num[]){
        for(int i=0; i<3; i++)
            num[i] = num[i] * 2;
    }
}
```



```
TERMINAL  CONSOLE DE DEPURACÃO  PROBLEMAS  4  SAÍDA

Digite o 1º número: 5
Digite o 2º número: 7
Digite o 3º número: 9

Valores após a chamada do método
Numero: 10
Numero: 14
Numero: 18
```

Exemplos

3- Faça um programa em Java que obtenha as notas de 10 alunos em 3 provas diferentes e mostre um relatório com o número do aluno (linha + 1) e a sua média; no final exiba a quantidade de alunos aprovados ($\text{media} \geq 6$). Utilize uma matriz 10 x 3 para armazenar as notas.

Exemplos

```
import javax.swing.JOptionPane;
public class ExemploNotas {
    public static void main(String[] args) {
        float[][] nota = new float[10][3];
        float[] media = new float[10];
        int i, j, qte=0;
        for(i=0; i<10; i++){
            float soma = 0;
            for(j=0; j<3; j++){
                nota[i][j] = Float.parseFloat(JOptionPane.showInputDialog(null,
                    "Digite a " + (j+1) + "ª nota do " + (i+1) + "º aluno"));
                soma += nota[i][j];
            }
            media[i] = soma/3;
            if(media[i]>6)
                qte++;
        }
    }
}
```

Exemplos

```
System.out.println("Nº\tNota1\tNota2\tNota3\tMedia");
for(i=0; i<10; i++){
    System.out.print((i+1)+"\t");
    for(j=0; j<3; j++){
        System.out.print(nota[i][j)+"\t");
    }
    System.out.print(media[i)+"\n");
}
System.out.println("A quantidade de alunos aprovados é: " + qte);
}
}
```

Exemplos

4- Faça um algoritmo que armazene em um vetor a idade de 20 alunos de uma academia de ginástica. O algoritmo deve calcular e mostrar na tela a idade média dos alunos e o número de alunos com idade acima da média.

Exemplos

```
import javax.swing.JOptionPane;

public class Exemplo3 {
    public static void main(String[] args) {
        int idade[] = new int[20];
        int i, acimaM=0;
        float media, soma=0;
        for(i=0; i<20;i++){
            idade[i] = Integer.parseInt(JOptionPane.showInputDialog(null,
                                                                    "Digite a idade do " +(i+1)+ "º aluno"));

            soma += idade[i];
        }
        media = soma/20;
        System.out.println("A idade média dos alunos é: " + media);
        for(i=0; i<20; i++){
            if(idade[i]>media)
                acimaM++;
        }
        System.out.println("A quantidade de alunos com idade acima da média é: " + acimaM);
    }
}
```

Exemplos

5- Altere o **exemplo 3** para que seja feita a leitura do nome dos alunos (utilize um vetor de 10 posições) e exiba o relatório com o nome e o número do alunos. No final seja exibida também a média geral em cada prova.

Exemplos

```
import javax.swing.JOptionPane;
public class ExemploNotas2 {
    public static void main(String[] args) {
        float[][] nota = new float[10][3];
        float[] media = new float[10];
        String[] nome = new String[10];

        int i, j, qte=0;
        for(i=0; i<10; i++){
            float soma = 0;
            nome[i] = JOptionPane.showInputDialog(null, "Digite o nome do aluno: ");
            for(j=0; j<3; j++){
                nota[i][j] = Float.parseFloat(JOptionPane.showInputDialog(null,
                    "Digite a " + (j+1) + "ª nota do aluno " + nome[i]));

                soma += nota[i][j];
            }
            media[i] = soma/3;
            if(media[i]>6)
                qte++;
        }
    }
}
```

Exemplos

```
System.out.println("Nº\tNome\tNota1\tNota2\tNota3\tMedia");
for(i=0; i<10; i++){
    System.out.print((i+1)+"\t"+nome[i+"\t"]);
    for(j=0; j<3; j++){
        System.out.print(nota[i][j)+"\t");
    }
    System.out.print(media[i)+"\n");
}
System.out.println("A quantidade de alunos aprovados é: " + qte);
for(j=0; j<3; j++){
    float soma = 0;
    for(i=0; i<10; i++){
        soma += nota[i][j];
    }
    System.out.println("Média da "+(j+1)+"ª prova: " + (soma/10));
}
}
```

Alguma dúvida????



Então, agora é pra valer... Exercícios de aplicação



Observações sobre exercícios

- ✓ Todos os exercícios devem ser resolvidos em Java.
- ✓ Os códigos podem ser feito no NetBeans ou no Repl.it.
- ✓ Após finalizar todos os exercícios da aula, compacte todos os arquivos e pastas e envie no Blackboard.



Exercícios

1- Um famoso e-commerce de livros deseja fazer uma estatística das vendas do livro **Java: Como Programar**, durante o ano. Para tanto foram catalogados os seguintes dados:

Qte	10	15	8	13	7	2	23	4	17	7	9	20
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez

Desenvolver um programa em Java que seja capaz de:

- a) Encontrar os meses com menor e maior quantidade de vendas.
- b) Encontrar a média de livros vendida no período.
- c) Determinar quantos meses a média vendida foi superada.

OBS: Inicialize o vetor na sua declaração com os valores acima, não é necessário fazer a leitura.

Exercícios

- 2- Faça um algoritmo que leia um vetor de 10 posições de números inteiros e divida todos os seus elementos pelo maior valor do vetor e armazene o resultado no mesmo vetor. Mostre o vetor após os cálculos.
- 3- Faça um programa algoritmo que solicite ao usuário, enquanto o mesmo desejar, números e armazene-os em um vetor. Após a entrada de dados, somar os valores do vetor, calcular e mostrar a média. Calcule e mostre quantos números armazenados no vetor estão acima da média.

Exercício

4- Faça um programa em Java que obtenha os salários de 10 funcionários em 6 meses e armazene-os em uma matriz, obtenha também um vetor contendo os nomes de cada um, calcule e mostre a média salarial semestral de cada funcionário, juntamente com o seu nome.

Créditos

Esta aula foi elaborada com base no material produzido e cedido gentilmente pelos **Professores Alcides, Lédon, Ana e Cristiane.**





That's all Folks!