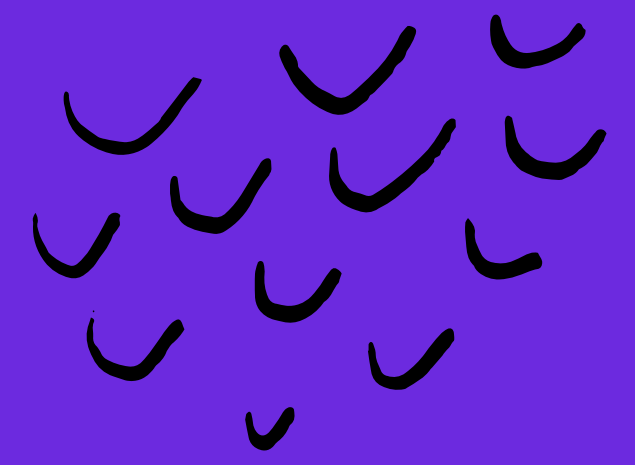
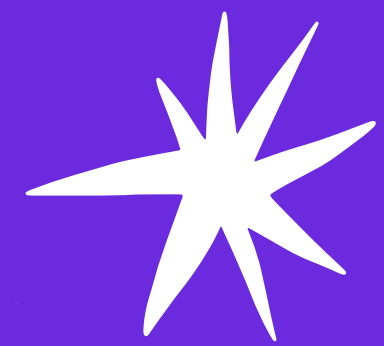


# VARIÁVEIS COMPOSTAS HOMOGÊNEAS



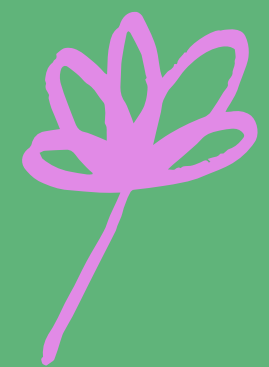
Vetores

unidimensional





NÃO PODEMOS RESOLVER  
PROBLEMAS USANDO O  
MESMO TIPO DE  
PENSAMENTO QUE USAMOS  
QUANDO OS CRIAMOS.           



ALBERT EINSTEIN



# ESTRUTURA DE DADOS

coleção de dados relacionados



- Coleções dos itens de dados relacionados.
- Arrays são estruturas de dados com dados do mesmo tipo relacionados



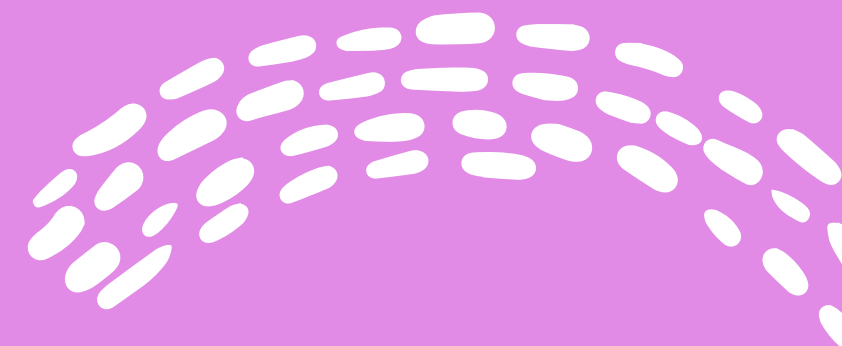
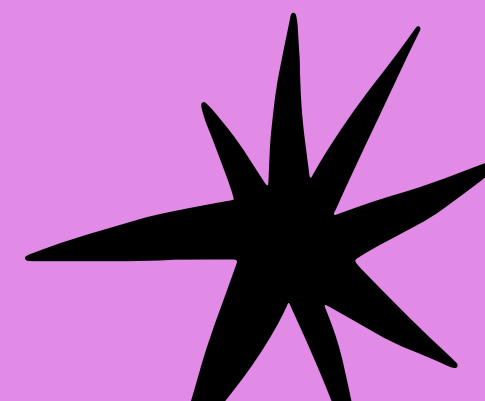
# NECESSIDADE



Armazenamento de dados semelhantes

Exemplo:

Crie um programa que leia a nota e o nome de três alunos e mostre quais alunos ficaram acima da média



```
Scanner dado = new Scanner(System.in);
float nota1, nota2, nota3, media;
String nome1, nome2, nome3;
System.out.print("Digite o nome do aluno 1: ");
nome1 = dado.nextLine();
System.out.print("Digite a nota do aluno 1: ");
nota1 = dado.nextFloat();
dado.nextLine();
System.out.print("Digite o nome do aluno 2: ");
nome2 = dado.nextLine();
System.out.print("Digite a nota do aluno 2: ");
nota2 = dado.nextFloat();
dado.nextLine();
System.out.print("Digite o nome do aluno 3: ");
nome3 = dado.nextLine();
System.out.print("Digite a nota do aluno 3: ");
nota3 = dado.nextFloat();
media = (nota1 + nota2 + nota3) / 3;
System.out.println("A media foi " + media);
if (nota1 > media) {
    System.out.println(nome1 + " obteve nota acima da media");
}
if (nota2 > media) {
    System.out.println(nome2 + " obteve nota acima da media");
}
if (nota3 > media) {
    System.out.println(nome3 + " obteve nota acima da media");
}
```

*UWU*

```
Digite o nome do aluno 1: lucilia
Digite a nota do aluno 1: 10
Digite o nome do aluno 2: bruna
Digite a nota do aluno 2: 7
Digite o nome do aluno 3: andre
Digite a nota do aluno 3: 5
A media foi 7.3333335
lucilia obteve nota acima da media
```

*✓ ✓ ✓ ✓ ✓*

E SE FOSSEM **100**  
ALUNOS???

**CRIAR 100 VARIÁVEIS**

**100 LINHAS PARA  
ENTRADA DE DADOS**

**100 COMPARAÇÕES  
COM A MÉDIA!!!!**





A SOLUÇÃO É  
ARMAZENAR  
EM UM ARRAY

```
Scanner dado = new Scanner(System.in);
final int TAMANHO = 3;
float[] nota = new float[TAMANHO];
String[] nome = new String[TAMANHO];
float somaNotas = 0, media;
for (int i = 0; i < TAMANHO; i++) {
    System.out.println("DADOS DO ALUNO " + (i + 1));
    System.out.print("Nome: ");
    nome[i] = dado.nextLine();
    System.out.print("Nota: ");
    nota[i] = dado.nextFloat();
    somaNotas += nota[i];
    dado.nextLine();
}
media = somaNotas / TAMANHO;
System.out.printf("\nA media foi %.1f\n", media);
System.out.println("ALUNOS ACIMA DA MEDIA");
System.out.printf("\n%-30s ... %5s\n", "NOME", "NOTA");
for (int i = 0; i < TAMANHO; i++) {
    if (nota[i] > media) {
        System.out.printf("%-30s ... %5.1f\n", nome[i], nota[i]);
    }
}
}
```



DADOS DO ALUNO 1

Nome: lucilia

Nota: 10

DADOS DO ALUNO 2

Nome: bruna

Nota: 8

DADOS DO ALUNO 3

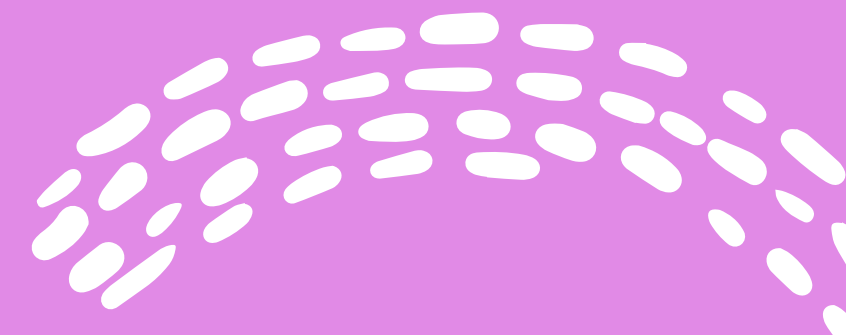
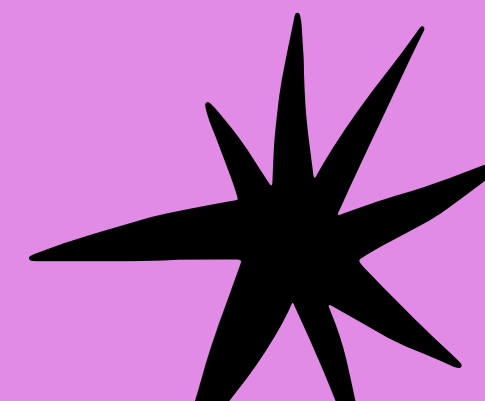
Nome: andre

Nota: 5

A media foi 7,7

ALUNOS ACIMA DA MEDIA

NOME	...	NOTA
lucilia	...	10,0
bruna	...	8,0

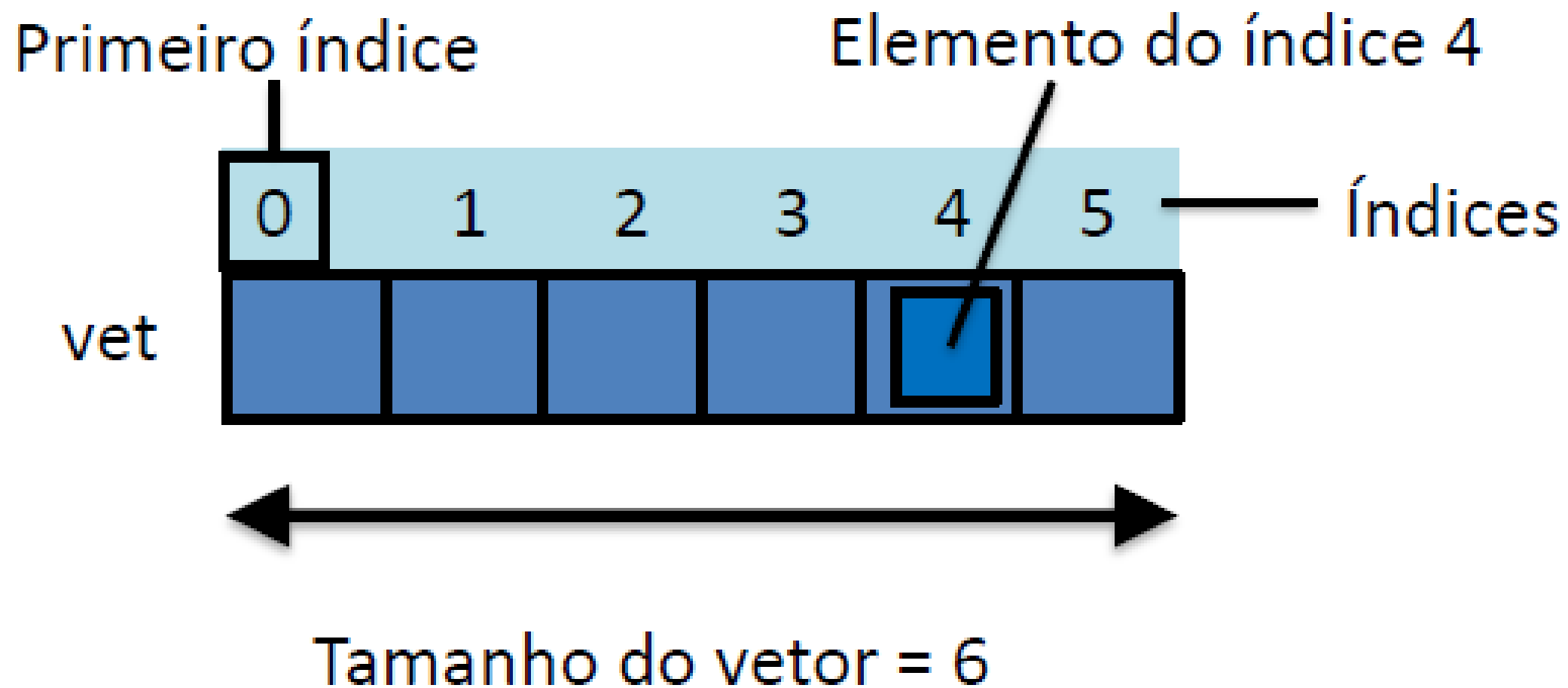




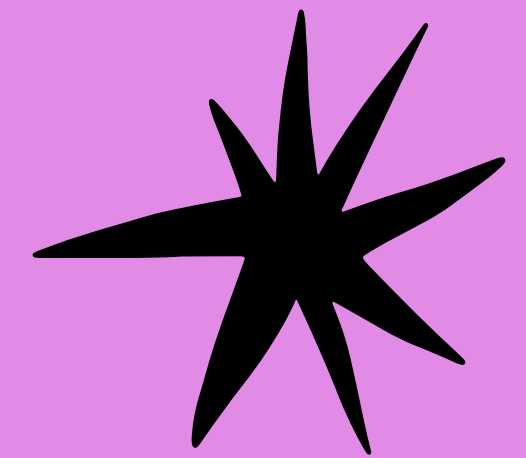
# VETOR

0 que é

Variável composta de vários valores do mesmo tipo, acessíveis por um índice



# VETOR



- Sequência de vários valores do mesmo tipo, armazenadas sequencialmente na memória, e fazendo uso de um mesmo nome de variável para acessar valores
- Cada elemento desta sequência pode ser acessado individualmente através de um índice (número inteiro)
- Os elementos são indexados de 0 até  $n - 1$ , onde  $n$  é a quantidade de elementos do vetor, também chamado de dimensão ou tamanho do vetor
- O vetor tem tamanho fixo (definido na declaração)





# EXEMPLOS DE VETORES

```
final int TAMANHO = 3;  
float[] nota = new float[TAMANHO];  
int matricula[] = new int[TAMANHO];  
String[] nome = new String[TAMANHO];  
int[] num = { 1, 7, 13 };
```



# ACESSO AO CONTEÚDO DO VETOR



Realizado por meio do operador de índice

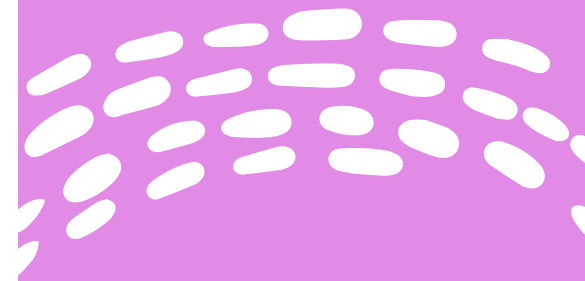
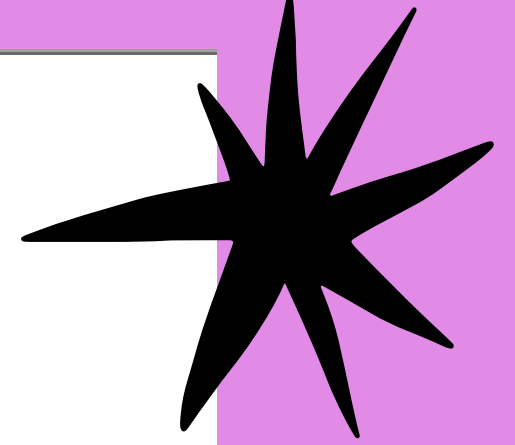
- Utiliza uma referência de memória (variável do tipo vetor)
- Utiliza um índice (número inteiro)
- Retorna uma referência para o elemento correspondente ao índice (valor)



Faça um programa que leia dez números, os imprima em ordem inversa e mostre o maior e o menor número. Para isso, é necessário armazenar os dez números para poder imprimi-los.



```
import java.util.Scanner;
public class ExemploArray2 {
    public static void main(String[] args) {
        Scanner dado = new Scanner(System.in);
        final int TAMANHO = 10;
        int maior = 0, menor = 0;
        int[] numero = new int[TAMANHO];
        System.out.println("Digite 10 números inteiros: ");
        for (int i = 0; i < TAMANHO; i++) {
            System.out.print("[ " + (i + 1) + " ] :");
            numero[i] = dado.nextInt();
            if (i == 0) {
                maior = numero[i];
                menor = numero[i];
            } else {
                if (numero[i] > maior) {
                    maior = numero[i];
                }
                if (numero[i] < menor) {
                    menor = numero[i];
                }
            }
        }
        System.out.println("\nValores em ordem reversa: ");
        for (int i = (TAMANHO - 1); i >= 0; i--) {
            System.out.print(numero[i] + " - ");
        }
        System.out.println("\nMaior = " + maior + " e Menor = " + menor);
    }
}
```



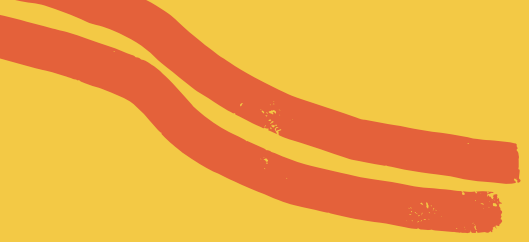


Digite 10 números inteiros:

```
[1] :1  
[2] :2  
[3] :3  
[4] :4  
[5] :5  
[6] :6  
[7] :10  
[8] :18  
[9] :20  
[10] :21
```

Valores em ordem reversa:

```
21 - 20 - 18 - 10 - 6 - 5 - 4 - 3 - 2 - 1  
Maior = 21 e Menor = 1
```



# CLASSE ARRAYS





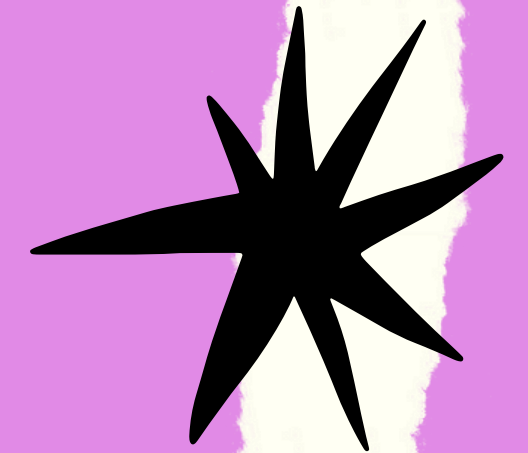
```
import java.util.Arrays;
public class ClasseArray {
    public static void main(String[] args) {

        double[] vetorDouble = { 8.4, 9.3, 0.2, 7.9, 3.4 };
        Arrays.sort(vetorDouble); // classifica em ordem crescente
        System.out.print("\nVETOR ORDENADO: ");
        //instrução "for" aprimorada - (estilo for .. each)
        for (double valor : vetorDouble) {
            System.out.printf("%.1f - ", valor);
        }

        int[] vetorPreenchido = new int[10];
        Arrays.fill(vetorPreenchido, 7); // preenche o vetor com 7s
        mostraVetor(vetorPreenchido, "VETOR PREENCHIDO");

        int[] vetorInt = { 1, 2, 3, 4, 5, 6, 7 };
        int[] copiaVetorInt = new int[vetorInt.length];
        // copia vetorInt em copiaVetorInt
        //(origem, posição inicial, destino, posição inicial, quantos elementos)
        System.arraycopy(vetorInt, 3, copiaVetorInt, 0, vetorInt.length - 3);
        mostraVetor(vetorInt, "VETOR INTEIRO");
        mostraVetor(copiaVetorInt, "COPIA DO VETOR");

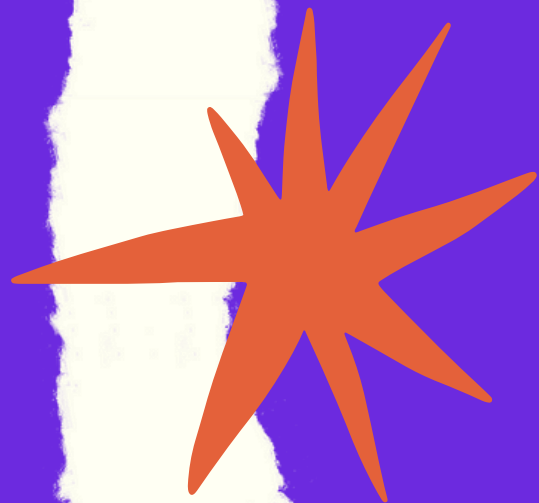
        // verifica a igualdade entre dois vetores
        boolean b = Arrays.equals(vetorInt, copiaVetorInt);
        // if ternário: condição ? verdadeiro : falso
        System.out.printf("\nvetorInt %s copiaVetorInt\n", (b ? "==" : "!="))
    }
}
```



```
// pesquisa o valor 5 em vetorInt
int local = Arrays.binarySearch(vetorInt, 5);
// if ternário
String msg = local >= 0 ?
    "\nEncontrado 5 na posicao " + local :
    "\n5 nao encontrado no vetor ";
System.out.println(msg);
//usa o método procuraVetor
procuraVetor(vetorInt, 42);
}
```

```
//gera saída de dados para vetor inteiro
public static void mostraVetor(int[] vetor, String descricao) {
    System.out.printf("\n%s: ", descricao);
    for (int valor : vetor) {
        System.out.printf("%d - ", valor);
    }
}
```

```
public static void procuraVetor(int[] vetor, int valor) {
    int local = Arrays.binarySearch(vetor, valor);
    String msg = local >= 0 ?
        ("\nEncontrado " + valor + " na posicao " + local) :
        ("\n" + valor + " nao encontrado no vetor ");
    System.out.println(msg);
}
```



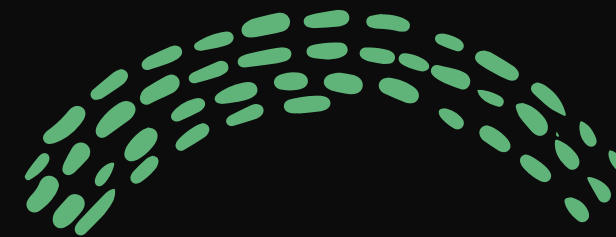


```
VETOR ORDENADO: 0,2 - 3,4 - 7,9 - 8,4 - 9,3 -  
VETOR PREENCHIDO: 7 - 7 - 7 - 7 - 7 - 7 - 7 - 7 - 7 - 7 - 7  
VETOR INTEIRO: 1 - 2 - 3 - 4 - 5 - 6 - 7 -  
COPIA DO VETOR: 4 - 5 - 6 - 7 - 0 - 0 - 0 -
```

```
vetorInt != copiaVetorInt
```

```
Encontrado 5 na posicao 4
```

```
42 nao encontrado no vetor
```





# CLASSE STRING



```
String s0 = "como vai";
char[] vetorChar = new char[5];
System.out.printf("%ns0: %s", s0);

//tamanho da string: length
System.out.printf("%nTamanho de s0: %d", s0.length());
System.out.printf("%nORDEM REVERSA: ");
for (int cont = s0.length() - 1; cont >= 0; cont--) {
    System.out.printf("%c ", s0.charAt(cont));
}

// copia caracteres de string para vetor de char: getChars
s0.getChars(0, 5, vetorChar, 0);
System.out.printf("%nVETOR DE CARACTERES: ");
for (char caractere : vetorChar) {
    System.out.print(caractere);
}
System.out.println();
```



```
s0: como vai
Tamanho de s0: 8
ORDEM REVERSA: i a v o m o c
VETOR DE CARACTERES: como
```

```
// comparação de strings
String s1 = new String("Que o dia"), s2 = "seja feliz";
String s3 = "Feliz Aniversario", s4 = "feliz aniversario";

// usando equals (testa quanto à igualdade)
if (s1.equals("Que o dia"))
    System.out.println("s1 igual a \"Que o dia\" ");

// usando equals (ignorando maiúsculas e minúsculas)
if (s3.equalsIgnoreCase(s4))
    System.out.printf("%s igual a %s %n", s3, s4);

// usando o == (compara objetos)
if (s1 == "Que o dia")
    System.out.println("s1 eh o mesmo objeto que \"Que o dia\" ");
else
    System.out.println("s1 nao eh o mesmo objeto que \"Que o dia\" ");
```



```
s1 igual a "Que o dia"
Feliz Aniversario igual a feliz aniversario
s1 nao eh o mesmo objeto que "Que o dia"
```

```
String s1 = new String("Que o dia"), s2 = "seja feliz";  
String s3 = "Feliz Aniversario", s4 = "feliz aniversario";
```

```
//usando o compareTo
```

```
System.out.printf("%n s1.compareTo(s2)= %d", s1.compareTo(s2));  
System.out.printf("%n s2.compareTo(s2)= %d", s2.compareTo(s1));  
System.out.printf("%n s1.compareTo(s1)= %d", s1.compareTo(s1));  
System.out.printf("%n s4.compareTo(s3)= %d", s4.compareTo(s3));  
System.out.printf("%n s3.compareTo(s4)= %d", s3.compareTo(s4));
```

```
//usando regionMatches
```

```
if (s3.regionMatches(0, s4, 0, 5))
```

```
    System.out.println
```

```
    ("\n5 primeiros caracteres de s3 e s4 coincidem");
```

```
else
```

```
    System.out.println
```

```
    ("\n5 primeiros caracteres de s3 e s4 nao coincidem");
```

```
if (s3.regionMatches(true, 0, s2, 5, 5)) // sem case sensitive
```

```
    System.out.println
```

```
    ("\n5 primeiros caracteres de s3 coincidem com os de s2");
```



```
s1.compareTo(s2)= -34
```

```
s2.compareTo(s2)= 34
```

```
s1.compareTo(s1)= 0
```

```
s4.compareTo(s3)= 32
```

```
s3.compareTo(s4)= -32
```

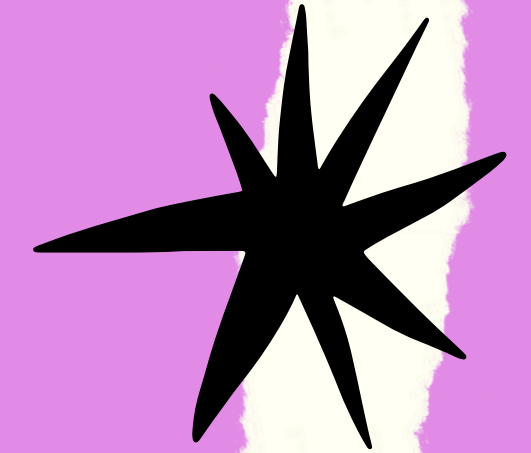
```
Os 5 primeiros caracteres de s3 e s4 nao coincidem
```

```
Os 5 primeiros caracteres de s3 coincidem com os de s2
```

```
//métodos startswith e endswith
String[] strings = {"inicio", "iniciando", "final", "finalizando"};
for (String string : strings) {
    if (string.startsWith("in"))
        System.out.printf("\'%s\' começa com \'in\' %n", string);

    if (string.startsWith("ci", 3)) //índice que a comparação inicia
        System.out.printf
            ("\'%s\' começa com \'ci\' na posicao 4%n", string);

    if (string.endsWith("ando"))
        System.out.printf
            ("\'%s\' termina com \'ando\' %n", string);
}
```

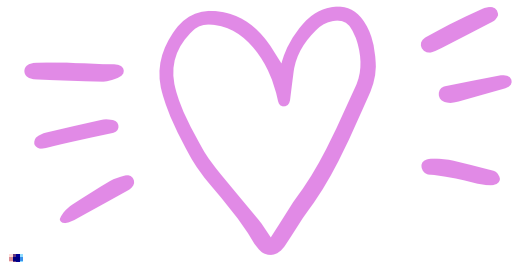


```
"inicio" começa com "in"
"inicio" começa com "ci" na posicao 4
"iniciando" começa com "in"
"iniciando" começa com "ci" na posicao 4
"iniciando" termina com "ando"
"finalizando" termina com "ando"
```





```
//métodos de pesquisa
String letras = "abcdefghijklmabcdefghijklm";
System.out.printf
    ("'k' está localizado em %d %n", letras.indexOf('k'));
System.out.printf
    ("'a' está localizado em %d %n", letras.indexOf('a', 1));
System.out.printf
    ("'$' está localizado em %d %n", letras.indexOf('$'));
System.out.printf
    ("'def' está localizado em %d %n", letras.indexOf("def"));
System.out.printf
    ("o ultimo 'k' está localizado em %d %n", letras.lastIndexOf('k'));
```



```
'k' está localizado em 10
'a' está localizado em 13
'$' está localizado em -1
'def' está localizado em 3
o ultimo 'k' está localizado em 23
```

```
//substring
System.out.printf
("Substring do 20 ate o final: %s %n", letras.substring(20));
System.out.printf
("A partir do 3 até o 6 (não incluso): %s %n", letras.substring(3, 6));

s1 = "feliz"; s2 = "ANIVERSARIO"; s3 = "    ESPACOS    ";
System.out.printf("s1 = %s%s2 = %s%s3 = %s%n", s1, s2, s3);

//replace
System.out.printf
("Troca o 'l' por 'L' em s1: %s%n", s1.replace('l', 'L'));
```



```
Substring do 20 ate o final: hijklm
A partir do 3 até o 6 (não incluso): def
s1 = feliz
s2 = ANIVERSARIO
s3 =    ESPACOS
Troca o 'l' por 'L' em s1: feLiz
```



```
//toLowerCase e toUpperCase
System.out.printf
("s1.toUpperCase(): %s\n", s1.toUpperCase());
System.out.printf
("s2.toLowerCase(): %s\n", s2.toLowerCase());

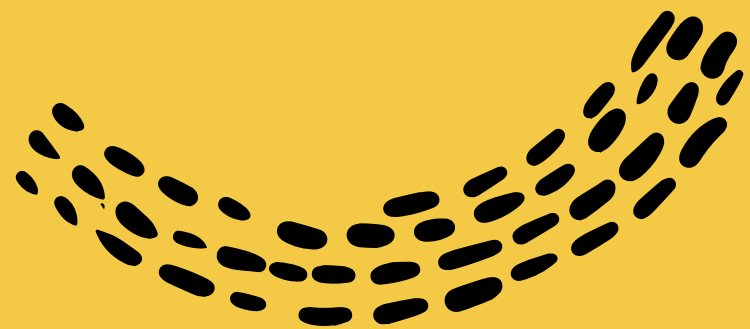
//trim: tira os espaços antes e depois
System.out.printf
("tam s3 antes do trim: %d e depois: %d\n", s3.length(), s3.trim().length());

//transforma string em vetor de char
vetorChar = s1.toCharArray();
for (char caractere : vetorChar)
    System.out.print(caractere + " - ");
System.out.println();
```

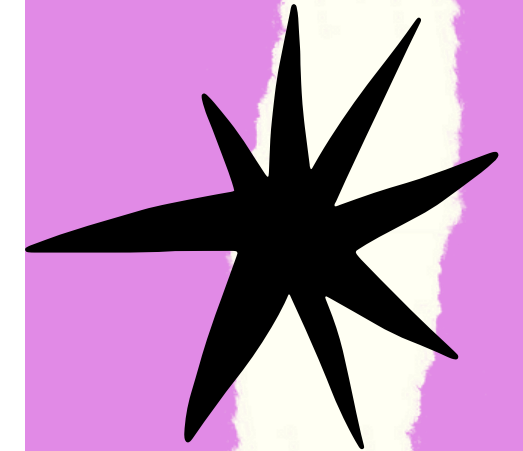


```
s1.toUpperCase(): FELIZ
s2.toLowerCase(): aniversario
tam s3 antes do trim: 17 e depois: 7
f - e - l - i - z -
```

# CLASSE ARRAYLIST

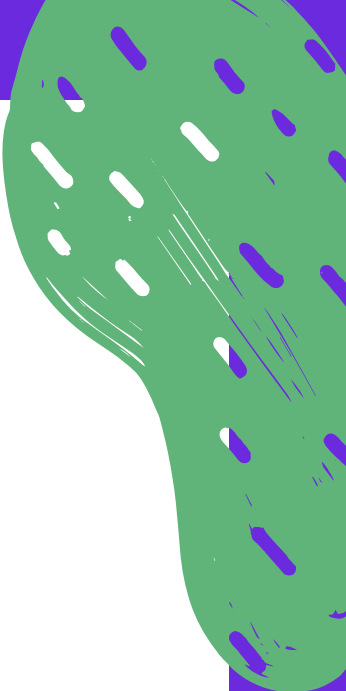
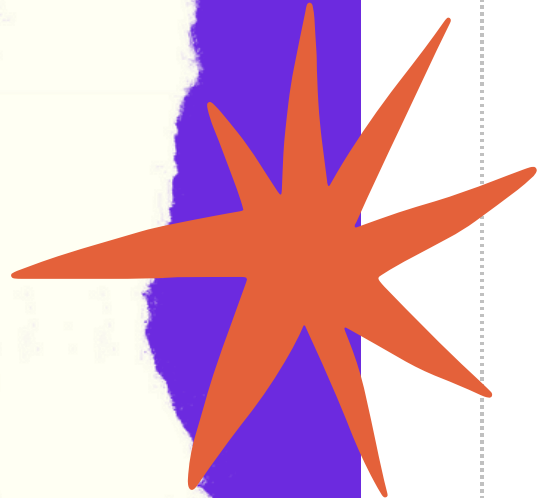
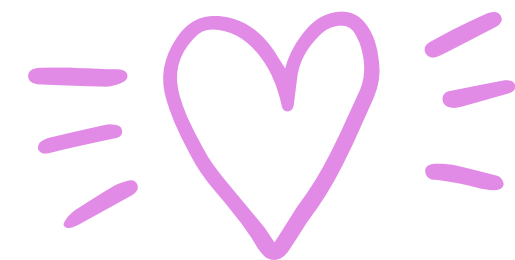


```
import java.util.ArrayList;
public class ClasseArrayList {
    public static void main(String[] args) {
        // por padrão, a quantidade de elementos é 10
        ArrayList<String> cores = new ArrayList<String>();
        ArrayList<Integer> num = new ArrayList<Integer>(7);
        cores.add("vermelho");
        display(cores, "CORES");
        cores.add(1, "amarelo");
        display(cores, "INSERIU O AMARELO NA POSIÇÃO 1");
        cores.add(0, "azul");
        display(cores, "INSERIU O AZUL NA POSIÇÃO 0");
        cores.add("verde"); // insere no final
        cores.add("branco");
        cores.add("preto");
        cores.add("laranja");
        cores.add("marron");
        cores.add("roxo");
        cores.add("rosa");
        cores.add("cinza"); //observe que ultrapassou 10 elementos
        cores.add("branco");
        System.out.println("LISTA DE CORES");
        for (int i = 0; i < cores.size(); i++) {
            System.out.printf("[%02d] %s\n", i + 1, cores.get(i));
        }
    }
}
```



```
System.out.println("LISTA DE CORES");  
for (int i = 0; i < cores.size(); i++) {  
    System.out.printf("[%02d] %s\n", i + 1, cores.get(i));  
}  
cores.remove("branco"); //remove o primeiro "branco"  
cores.remove(7); //remove o 6º elemento  
display(cores, "LISTA COM ITENS REMOVIDOS");  
//verifica se um valor está na lista  
System.out.printf("vermelho %s na lista\n",  
    cores.contains("vermelho") ? "esta" : "nao esta");  
System.out.printf("roxo %s na lista\n",  
    cores.contains("roxo") ? "esta" : "nao esta");  
//mostra o número de elementos da lista  
System.out.printf("Tamanho = %d\n", cores.size());  
// mostra o índice de determinado elemento  
System.out.printf("Índice do laranja: %d\n", cores.indexOf("laranja"))  
}
```

```
public static void display(ArrayList<String> items, String header) {  
    System.out.println(header);  
    for (String item : items) {  
        System.out.printf("  %s\n", item);  
    }  
    System.out.println();  
}
```



CORES

vermelho

INSERIU O AMARELO NA POSIÇÃO 1

vermelho amarelo

INSERIU O AZUL NA POSIÇÃO 0

azul vermelho amarelo

LISTA DE CORES

[01] azul

[02] vermelho

[03] amarelo

[04] verde

[05] branco

[06] preto

[07] laranja

[08] marron

[09] roxo

[10] rosa

[11] cinza

[12] branco



LISTA COM ITENS REMOVIDOS

azul

vermelho

amarelo

verde

preto

laranja

marron

rosa

cinza

branco

vermelho esta na lista

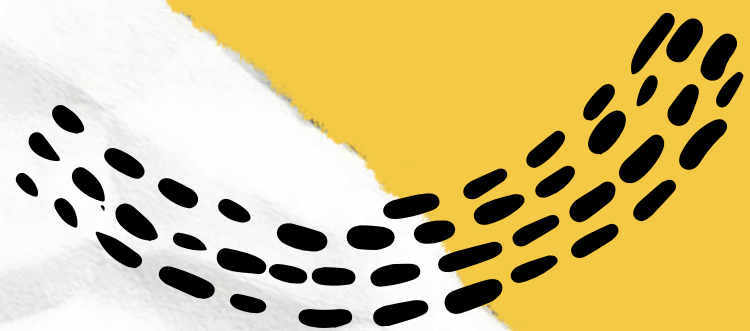
roxo nao esta na lista

Tamanho = 10



Indice do laranja: 5



# UTILIZANDO ARGUMENTOS DA LINHA DE COMANDO







Argumentos de linha de comando:  
Crie um programa que utilize três  
argumentos: tamanho do vetor, valor  
inicial e incremento. Seu programa  
deve ter o nome Args:

```
>> java Args 5 0 4
```



```
public class Args {
    public static void main(String[] args) {
        // verifica o número de argumentos da linha de comando
        if (args.length != 3) {
            System.out.println("Erro! Digite novamente o comando: ");
            System.out.println("< Args <tamanho do vetor> <valor inicial> <incremento> >");
        } else {
            int dimensao = Integer.parseInt(args[0]);
            int[] vetor = new int[dimensao];
            int inicio = Integer.parseInt(args[1]);
            int incremento = Integer.parseInt(args[2]);
            for (int i = 0; i < dimensao; i++) {
                vetor[i] = inicio + (incremento * i);
            }
            System.out.printf("%s%8s%n", "Indice", "Valor");
            for (int i = 0; i < vetor.length; i++) {
                System.out.printf("%5d%8d%n", i, vetor[i]);
            }
        }
    }
}
```





```
C:\Code>java Args
```

```
Erro! Digite novamente o comando:
```

```
< Args <tamanho do vetor> <valor inicial> <incremento> >
```

```
C:\Code>java Args 5 0 4
```

```
Indice      Valor
```

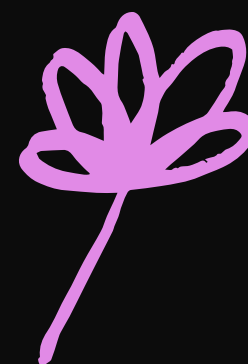
```
0           0
```

```
1           4
```

```
2           8
```

```
3          12
```

```
4          16
```





# OBRIÇADA!

Alguma Pergunta?

[professora@lucilia.com.br](mailto:professora@lucilia.com.br)

