

JAVA

4º

```
package nomeprograma;
public class NomePrograma {
    public static void main(String[] args) {
        System.out.print("Sring em qestão");
    }
}
```

Uso da tecla Tab: **psvm _ sout**.

(**public**: Torna um Método publico); (**static**: Torna um método estático. "Não precisa transformar em objeto para usa-lo"); (**void**: O método não vai retornar num valor); (**String[] args**: Este é um parâmetro indicando pelo colchetes que args é um vetor de String)

CamelCase. Sempre que a primeira letra for maiúscula e o resto for usando CamelCase, indica que é uma **classe ou interface**. (Primeira letra maiúscula... é classe.) ex; **SergioStarke**.

Quando a primeira letra for minucula e as demais maiusc. Indica que pode ser; **meuAtributo**, **minhaVariável**, **meuMétodo**.

Quando tudo minusc. Indica ser **nomedopacote**.

Quando tudo maiúsculo **UMA_CONSTANTE**

Ex.4º

```
public static void main(String[] args) {
    Date relógio = new Date ();
    System.out.println("A data e hora do Sistema é;");
    System.out.println(relógio.toString());
}
```

A data e hora do Sistema é;
Thu Jul 16 13:48:09 BRT 2020

```
package testessss;
import java.util.Calendar;
import java.util.GregorianCalendar;

public class Testesssss {
    public static void main(String[] args) {
        Calendar cal = GregorianCalendar.getInstance();
        System.out.println(cal.get(Calendar.YEAR));
    }
}
```

System.out.println(String.valueOf(Calendar.getInstance().get(Calendar.YEAR)));

String.valueOf(Calendar.getInstance().get(Calendar.YEAR))

```
Locale loc = Locale.getDefault();
System.out.print ("O seu sistema esta em ");
System.out.println(loc.getDisplayLanguage());
System.out.println (loc.getLanguage());
```

"O seu sistema esta em" **português - (pt)**

1/9/2021

1 de Outubro de 2021

```
Calendar d = GregorianCalendar.getInstance();
String dia= Integer.toString(d.get(Calendar.DAY_OF_MONTH));
String mes= Integer.toString(d.get(Calendar.MONTH));
String ano= Integer.toString(d.get(Calendar.YEAR));
lbDate.setText(dia+"/"+mes+"/"+ano);
```

```
Date date = new Date();
DateFormat form = DateFormat.getInstance(DateFormat.LONG);
lbComp.setText(form.format(date));
```

```
public static void main(String[] args) {
    Calendar c = Calendar.getInstance();
    Date data = c.getTime();

    Locale brasil = new Locale("pt", "BR"); //Retorna do país e a língua
    //Locale eua = Locale.US;
    //Locale italia = Locale.ITALIAN;

    DateFormat form = DateFormat.getDateInstance(DateFormat.FULL, brasil);
    System.out.println(form.format(data));
}

Run:
Terça-feira, 8 de Fevereiro de 2022
```

Data: 20/10/21

```
DateFormat dateformat = new SimpleDateFormat("dd/MM/YY");
Date date = new Date();
//txData.setText(""+ dateformat.format(date));
public static void main(String[] args) {
    Date horas = new Date();
    System.out.println("A hora e data do sistema é: "+ horas);
}
```

"A hora e data do sistema é:" Thu Jul 16 14:23:24 BRT 2020

ESTE EX.FOI FEITO POR CONCATENAÇÃO.

```
Toolkit tk = Toolkit.getDefaultToolkit();
Dimension d = tk.getScreenSize();
System.out.println("Screen width = " + d.width);
System.out.println("Screen height = " + d.height);
```

Screen width =1366

Screen height =768

5º Introdução ao Swing e JavaFX

```
public class TelaSwing extends javax.swing.JFrame {
    private javax.swing.JButton btnClick;
    private javax.swing.JLabel lblMensagem;
    private void btnClickActionPerformed(...) {
        lblMensagem.setText("Olá, Mundo!");
    }
}
```

Encapsulamento Herança Evento

```
import javafx.fxml.FXML;
public class FXMLDocumentController
    implements Initializable {
    @FXML
    private Button btnClick;
    private Label lblMensagem;
    @FXML
    private void clicouBotao(...) {
        lblMensagem.setText("Olá
    }
}
```

Interface

Em vermelho: Cabeçalho.
Swing usa Herança do JFrame e JavaFX usa Importação da biblioteca FXML
Coluna Verde: Declaração dos controles do botão e do label.
Coluna Azul: Método do botão.

```
public class TelaSwing extends javax.swing.JFrame {
    private javax.swing.JButton btnClick;
    private javax.swing.JLabel lblMensagem;
    private void btnClickActionPerformed(...) {
        lblMensagem.setText("Olá, Mundo!");
    }
}

import javafx.fxml.FXML;
public class FXMLDocumentController
    implements Initializable {
    @FXML
    private Button btnClick;
    private Label lblMensagem;
    @FXML
    private void clicouBotao(...) {
        lblMensagem.setText("Olá, Mundo!");
    }
}
```

Swing
JavaFX
FXML

```
package exercicio_swing.practising;

public class SwingTest extends javax.swing.JFrame {

    public SwingTest() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void ButtonActionPerformed(java.awt.event.ActionEvent evt) {
        Msg.setText("Praticando há esperança no aprendizado...");
    }

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new SwingTest().setVisible(true);
            }
        });

        // Variables declaration - do not modify
        private javax.swing.JButton Button;
        private javax.swing.JLabel Msg;
```

```
package exerciciojavafx;

import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Button;
import javafx.scene.control.Label;

public class FXMLDocumentController implements Initializable {

    @FXML
    private Label lblMensagem;
    private Button btnClick;

    @FXML
    private void clicouBotao(ActionEvent event) {
        lblMensagem.setText("A pratica e o erro ensinam...");
    }

    @Override
    public void initialize(URL url, ResourceBundle rb) {
    }
}
```

```
Date relógio = new Date();
lblHora.setText(relógio.toString());
```

Alg. Para mostrar data e hora atual no formato janela Swing.

`XX.setText("XX");` É para estabelecer "escrever" algo atribuído ou digitado. (Atribuir comando ao acionar botão)

6º Tipos Primitivos e Manipulação de Dados

```
// Este é um comentário de uma linha

/*
 * Nos comentários de muitas linhas,
 * podemos escrever o quanto quiser
 * até indicarmos o final do comentário
 */

/**
 * Exemplo para o CursoemVideo de Java
 * @author Gustavo Guanabara
 * @version 1.0
 * @since 2015-01-01
 */
```

*Comentário p. uma linha.

*Comentários mult.Linhas.

*Comentários para documentação.

Tabela de conceito Primitivo

Família	Tipo Primitivo	Classe Invólucro	Tamanho	Exemplo
Lógico	boolean	Boolean	1 bit	true
Literais	char	Character	1 byte	'A'
	-	String	1 byte/cada	"JAVA"
Inteiros	byte	Byte	1 byte	127
	short	Short	2 bytes	32 767
	int	Integer	4 bytes	2 147 483
	long	Long	8 bytes	2 ⁶³
Reais	float	Float	4 bytes	3.4e ⁺³⁸
	double	Double	8 bytes	1.8e ⁺³⁰⁸

SAÍDA DE DADOS

```
package tiposprimitivos;

public class TiposPrimitivos {

    public static void main(String[] args) {
        float nota = 8.5f;
        System.out.println("A nota é " + nota);
    }
}
```

A nota é 8.5

```
package tiposprimitivos;

public class TiposPrimitivos {

    public static void main(String[] args) {
        String nome = "Sérgio F. Starke";
        float nota = 8.5f;
        System.out.printf("A nota de %s é %.2f.\n", nome, nota);
    }
}
```

A nota de Sérgio F. Starke é 8.50.

```
package tiposprimitivos;

public class TiposPrimitivos {

    public static void main(String[] args) {
        String nome = "Sérgio F. Starke";
        float nota = 9.7f;
        System.out.format("A nota de %s é %.1f. \n", nome, nota);
    }
}
```

A nota de Sérgio F. Starke é 9.7.

ENTRADA DE DADOS

```
public static void main(String[] args) {
    Scanner teclado = new Scanner(System.in);
    System.out.print("Digite o nome: ");
    String nome = teclado.nextLine();
    System.out.print("Dgigite a nota: ");
    float nota = teclado.nextFloat();
    System.out.format("A nota de %s é %.1f \n", nome, nota);
}
```

```
import java.util.Scanner;
```

```
Scanner teclado = new Scanner(System.in);
```

```
int idade = teclado.nextInt();
```

```
float salario = teclado.nextFloat();
```

```
String nome = teclado.nextLine();
```

PraticaEntradaDado > main >

PraticaEntradaDado (run) X

```
run:
Digite o nome: Felipe
Dgigite a nota: 8
A nota de Felipe é 8,0
CONSTRUÍDO COM SUCESSO (tempo total: 11 segundos)
```

```
Scanner leia = new Scanner(System.in);
System.out.print("Qual seu nome: ");
String n = leia.nextLine();
System.out.print("Qual sua idade: ");
int v = leia.nextInt();
System.out.print("Qual pretensão salarial? ");
float sal = leia.nextFloat();
System.out.printf("O Sr %s de idade "+v+", pretende ganhar R$%.2f ao mês.\n", n, sal);
```

JavaApplication42 (run) X

```
run:
Qual seu nome: Sergio
Qual sua idade: 46
Qual pretensão salarial? 2500
O Sr Sergio de idade 46, pretende ganhar R$2500,00 ao mês.
```


Operadores Relacionais

>	Maior que	5 > 2	true
<	Menor que	4 < 1	false
>=	Maior ou igual a	8 >= 3	true
<=	Menor ou igual a	6 <= 6	true
==	Igual a	9 == 8	false
!=	Diferente de	4 != 5	true

```
Scanner n = new Scanner(System.in);
System.out.print("Dig 1º n°: ");
byte x = n.nextByte();
System.out.print("Dig 2º n°: ");
byte y = n.nextByte();
System.out.print("Dig 3º n°: ");
byte z = n.nextByte();
System.out.print("Triangulo: ");
System.out.println((x < y + z) && (y < x + z) && (z < x + y));
System.out.print("Equilatero: ");
System.out.println((x == y) && (y == z));
System.out.print("Escaleno: ");
System.out.println((x != y) && (y != z) && (x != z));
```

Exercício do Triangulo com operadores Logicos; (Igual ==), (Diferente !=) e (Maior ou Menor > <), (E &&), (Ou ||).

&&	.E.	true && false	false
	.OU.	false true	true
^	.XOU.	true ^ true	false
!	.NAO.	! false	true

p	q	p && q	p q	p ^ q	p	!p
V	V	V	V	F	V	F
V	F	F	V	V	V	F
F	V	F	V	V	F	V
F	F	F	F	F	F	V

E

OU

XOU (Exclusivo)

NÃO

Solução para incompatibilidade entre String e Número.

```
int idade = 30;
String valor = idade;
String valor = (String) idade;
String valor = Integer.toString(idade);
```

```
String valor = "30";
int idade = valor;
int idade = (int) valor;
int idade = Integer.parseInt(valor);
```

```
/*float teste = 76f;
String troco = Float.toString(teste);
System.out.printf("%s \n", troco);

/*int teste = 64;
String troco = Integer.toString(teste);
System.out.println(troco);

/*String teste = "10.0";
float troco = Float.parseFloat(teste);
System.out.printf("%.3f. \n ", troco);

/*String tes = "46";
int tro = Integer.parseInt(tes);
System.out.println(tro);*/
```

Quatro exemplos de como resolver incompatibilidade entre caracteres e números.

Através do uso da Classe Envolcro "Wrapper Class"

```
int n1 = Integer.parseInt(txtN1.getText());
int n2 = Integer.parseInt(txtN2.getText());
int s = n1 + n2;
lblSoma.setText(Integer.toString(s));
```

Atribuindo aos "n" 1 e 2 os valores que serão digitados pelo usuário.

XX.getText() e XX.setText(). São para obter e estabelecer comandos.

Ex.6º

```
int an = Integer.parseInt(txtAN.getValue().toString());
int id = 2020 - an;
lblIdade.setText(Integer.toString(id));
```

Ano de Nascimento

 Calcular

Idade **46**



7º Operadores Aritméticos e Classe Math

+	Adição	5 + 2	7
-	Subtração	5 - 2	3
*	Multiplicação	5 * 2	10
/	Divisão	5 / 2	2.5
%	Resto	5 % 2	1

Operadores Aritméticos

```
int n = Integer.parseInt(txNum.getText());
int d = Integer.parseInt(txDen.getText());
float div = n / d;
float res = n % d;
lblDiv.setText(Float.toString(div));
lblRes.setText(Float.toString(res));
```

Operadores Unários

```
int n = 5;
n++;
System.out.println(n);
```

6

```
int n = 5;
int x = 5 + n++;
System.out.println(x);
```

10

```
int y = 4;
y = y += 8;
System.out.println(y);
```

12

Operadores Unários

```
int n = 10;
int x = 5 + ++n;
System.out.println(x);
```

16

```
int n = 10;
int x = 4 + n--;
System.out.println(x);
System.out.println(n);
```

14 ... 9

```
float x = 4;
x = x / 8;
System.out.println(x);
```

0,5

+=	Somar e atribuir	a += b	a = a + b
-=	Subtrair e atribuir	a -= b	a = a - b
*=	Multiplicar e atribuir	a *= b	a = a * b
/=	Dividir e atribuir	a /= b	a = a / b
%=	Resto e atribuir	a %= b	a = a % b

Operadores de Atribuição

Classe Math

PI	Constante π	Math.PI	3.1415...
pow	Exponenciação	Math.pow(5,2)	25
sqrt	Raiz Quadrada	Math.sqrt(25)	5
cbrt	Raiz Cúbica	Math.cbrt(27)	3

Raiz Quadrada

```
int n = Integer.parseInt(txtN.getText());
float rq = (float) Math.sqrt(n);
lblRes.setText(Float.toString(rq));
```

5,0

Exponenciação ou Potenciação

```
float x = (float) Math.pow(5, 2);
System.out.println(x);
```

25,0

```
double x = Math.pow(5, 2);
System.out.println(x);
```

Arredondamentos

abs	Valor Absoluto	Math.abs(-10)	10
floor	Arredonda para Baixo	Math.floor(3.9)	3
ceil	Arredonda para Cima	Math.ceil(4.2)	5
round	Arredonda Aritmeticamente	Math.round(5.6)	6

```
float v = 8.9f;
int ar = (int) Math.floor(v);
System.out.println(ar);
```

8

```
float v = 8.3f;
int ar = (int) Math.ceil(v);
System.out.println(ar);
```

9

```
float v = 8.5f;
int ar = (int) Math.round(v);
System.out.println(ar);
```

9

```
float v = 8.4f;
int ar = (int) Math.round(v);
System.out.println(ar);
```

8

Métodos para arredondamentos.

```
double aleatorio = Math.random();
System.out.println(aleatorio);
```

O Java tem suporte a geradores de números, que é da classe Math.random(). Ele gera aleatoriamente números entre 0.0 a 1.0.

5 + Math.random() * (10-5)

0.0 a 1.0

```
double al = Math.random();
int n = (int) (0 + al * (1000 - 0));
System.out.println(n);
```

```
double aleatorio = Math.random();
int n = (int) (5 + aleatorio * (10-5));
System.out.println(n);
```

Método p. Criar Números ALEATÓRIOS.

```
/*Random n = new Random();
int na = n.nextInt(2);
System.out.println(na);*/

/*double n = 0+Math.random()*(2-0);
int na = (int) n;
System.out.println(na);*/

/*int na = (int) (0+Math.random()*(2-0));
System.out.println(na);*/
```

```
public class JanelaCalculos extends javax.swing.JFrame {

    /**
     * Creates new form JanelaCalculos
     */
    public JanelaCalculos() {
        initComponents();
        panCalc.setVisible(false);
    }
}
```

Construtor é um método com o mesmo nome da **classe** no qual dentro dele é criado o algoritmo para: Tornar a Janela dos “Ibl” invisível antes de iniciar o comando do botão.



```
int num = Integer.parseInt(txtNum.getValue().toString());

int r = num % 2;

lblResto.setText(Integer.toString(r));

double c = Math.pow(num, 3);

lblCubo.setText(Double.toString(c));

double rq = Math.sqrt(num);

lblRaizQ.setText(String.format("%.2f", rq));

double rc = Math.cbrt(num);

lblRaizC.setText(String.format("%.2f", rc));

int abs = Math.abs(num);

lblAbs.setText(Integer.toString(abs));

panCalc.setVisible(true);
```

8º Operadores Lógicos e Relacionais

Operador Ternário.

```
Se (n1 > n2) entao
  maior <- n1
senao
  maior <- n2
FimSe
```

```
int n1, n2, r;  
n1 = 4;  
n2 = 8;  
r = n1 > n2 ? 0 : 1;  
System.out.println(r);
```

```
int n1, n2, r;  
n1 = 14;  
n2 = 8;  
r = n1 > n2 ? n1 : n2;  
System.out.println(r);
```

```
int n1, n2, r;  
n1 = 14;  
n2 = 8;  
r = n1 > n2 ? n1+n2 : n1-n2;  
System.out.println(r);
```

```
String nl = "Starke";
String x = "Starke";
String terceiro = new String("Starke");
String r;
r = nl==x ? "igual":"diferente";
System.out.println(r);
```

```
String nl = "Starke";
String x = "Starke";
String terceiro = new String("Starke");
String r;
r = nl.equals(terceiro) ? "igual":"diferente";
System.out.println(r);
```

Operadores Lógicos

&&	.E.	true && false	false
	.OU.	false true	true
^	.XOU.	true ^ true	false
!	.NAO.	! false	true

Operadores Lógicos

p	q	$p \leftrightarrow q$	p	q	$p \vee q$	p	q	$p \wedge q$	p	$\neg p$
V	V	V	V	V	V	V	V	F	V	F
V	F	F	V	F	V	V	F	V	V	F
F	V	F	F	V	V	F	V	V	F	V
F	F	F	F	F	F	F	F	F		


```
int x, y, w;
x=3;
y=6;
w=9;
boolean r;
r = x>y ^ y<w ? true : false;
System.out.println(r);
```

true

```
int a = Integer.parseInt(txAno.getText());
int i = 2020 - a;
lbId.setText(Integer.toString(i));
String st;
st = i >=18 ? "Maior de Idade" : "Menor de Idade";
lbSt.setText(st);
```

```
int a = Integer.parseInt(txAno.getText());
int i = 2020 - a;
lbId.setText(Integer.toString(i));
String st;
st = ((i>=16 && i<18) || (i > 70)) ? "É opcional" : "Obrigatório";
lbSt.setText(st);
```

Ex.8º

```
public TelaGenio() {
    initComponents();
    lbFrase.setText("<html>Vou pensar em um valor entre <strong>1 e 5</strong>. Tente adivinhar...</html>");
}

@SuppressWarnings("unchecked")
Generated Code

private void btPalActionPerformed(java.awt.event.ActionEvent evt) {
    double n = 1 + Math.random() * (6-1);
    int v = (int) n;
    int escolha = Integer.parseInt(txVal.getValue().toString());
    String r = (escolha == v) ? "ACERTOU!!!" : "<html>ERROU!!! Pensei no n° "+v+"</html>";

    lbFrase.setFont(new Font("Tahoma", Font.BOLD, 18));
    lbFrase.setText(r);
}
```



9º Estruturas de Controles. 1ª Parte Estruturas Condicionais Simples e Composta. (IF , ELSE)

```
public static void main(String[] args) {
    Scanner leia= new Scanner(System.in);
    System.out.print("Dig. 1º nº: ");
    float n1 = leia.nextFloat();
    System.out.print("Dig. 2º nº: ");
    float n2 = leia.nextFloat();
    float m = (n1 + n2)/2;
    System.out.println("Sua média é "+m);
    if (m>=7) {
        System.out.println("Parabêns.. Você passou.");
    }
```

```
Scanner ler = new Scanner(System.in);
System.out.print("Informe o ano de nascimento: ");
int n = ler.nextInt();
int id = 2020 - n;
System.out.print("Sua idade é "+id+" . Portanto és... ");
if (id>=18){
    System.out.println("Maior");
} else{
    System.out.println("Menor");
}
```

```
int val = Integer.parseInt(txVal.getText());
if (val%2 == 0){
    lbRes.setText("PAR");
}else{
    lbRes.setText("IMPAR");
}
```

Valor 3518

Verificar

PAR

```
public static void main(String[] args) throws IOException {
    Scanner ler = new Scanner(System.in);
    // 2. instanciando e criando um objeto Scanner
    String nome;
    char sexo;

    System.out.printf("Informe um nome:\n");
    nome = ler.nextLine();
    // 3. entrada de dados (lendo uma String)

    System.out.printf("\nInforme o sexo (M/F): ");
    sexo = (char)System.in.read();
    // entrada de dados (lendo um caractere)

    System.out.printf("\nResultado:\n");
    if ((sexo == 'M') || (sexo == 'm'))
        System.out.printf("Seja bem vindo Sr. \"%s\".\n", nome);
    else System.out.printf("Seja bem vinda Sra. \"%s\".\n", nome);
}
```

Ex.09º

<div> <div> <div>1</div> <div>x²</div> <div>+</div> <div>2</div> <div>x</div> <div>+</div> <div>1</div> <div>= 0</div> </div> <div> <div>Δ = 2² - 4 . 1 . 1</div> <div>Calcular Δ</div> </div> <div> <div>Valor Δ 0,0</div> <div>Tipos de Raízes..... Existem Raízes Reais</div> </div> <div> <div>1</div> <div>x²</div> <div>+</div> <div>1</div> <div>x</div> <div>+</div> <div>1</div> <div>= 0</div> </div> <div> <div>Δ = 1² - 4 . 1 . 1</div> <div>Calcular Δ</div> </div> <div> <div>Valor Δ -3,0</div> <div>Tipos de Raízes..... Não Existem Raízes Reais</div> </div> </div>	<pre> private void txAStateChanged(javax.swing.event.ChangeEvent evt) { lbA.setText(txA.getValue().toString()); } private void txBStateChanged(javax.swing.event.ChangeEvent evt) { lbB.setText(txB.getValue().toString()); } private void txCStateChanged(javax.swing.event.ChangeEvent evt) { lbC.setText(txC.getValue().toString()); } private void btCalActionPerformed(java.awt.event.ActionEvent evt) { int a = Integer.parseInt(txA.getValue().toString()); int b = Integer.parseInt(txB.getValue().toString()); int c = Integer.parseInt(txC.getValue().toString()); double D = Math.pow(b,2)-4*a*c; lbDelta.setText(String.format("%.1f",D)); if (D<0){ lbRaiz.setText("Não Existem Raízes Reais"); }else{ lbRaiz.setText("Existem Raízes Reais"); } panRes.setVisible(true); } </pre> <div> <div>Calcular Δ</div> <div><html><sup>2</sup> - 4.</html></div> </div>
---	--

10º 2ª Parte Estruturas Condicionais Encadeadas.

(IF , ELSE / SWITCH)

Condição Composta Encadeada:

```
int x = Integer.parseInt(txId.getText());
if (x<16){
    txRes.setText("Isento");
}else if ((x>=16 && x<18) || (x>70)){
    txRes.setText("Opcional");
}else{
    txRes.setText("Obrigatório");
}
```

Verificador de obrigatoriedade
para votação:

Informe sua Idade

Verificar Condição

Sua condição é: **Aguarde...**

```
Scanner n = new Scanner(System.in);
System.out.print("Nº de pernas: ");
int np = n.nextInt();
System.out.print("Isso é um(a) ");
String res;
switch (np){
    case 1:
        res = "Saci.";
        break;
    case 2:
        res = "Bipede.";
        break;
    case 3:
        res = "Tripé.";
        break;
    case 4:
        res = "Quadrúpede.";
        break;
    case 6:
    case 8:
        res = "Aracnídio.";
        break;
    default:
        res = "E.T.";
}
System.out.println(res);
```

SWITCH

Escolha caso.

```
int n = -1;
String r = "";
switch (n){
    case 1:
    case 2:
    case 3:
        r = "1,2 ou 3...";
        break;
    case 4:
        r = "4...";
        break;
    default :
        r = "Outros nº...";
        break;
}

System.out.printf(r);
}
```

Ex. 10º

```
private void spAStateChanged(javax.swing.event.ChangeEvent evt) {
    lbVa.setText(Integer.toString(spA.getValue()));
}

private void spBStateChanged(javax.swing.event.ChangeEvent evt) {
    lbVb.setText(Integer.toString(spB.getValue()));
}

private void spCStateChanged(javax.swing.event.ChangeEvent evt) {
    lbVc.setText(Integer.toString(spC.getValue()));
}
```

A 12

B 10

C 10

Verificar

É um Triângulo

Isóceles




```

panRes.setVisible(true);
int a = spA.getValue();
int b = spB.getValue();
int c = spC.getValue();
if ((a < b + c) && (b < a + c) && (c < a + b)) {
    lbFor.setText("É um Triângulo");
    if ((a == b) && (b == c) && c == a) {
        lbTip.setText("Equilátero");
    } else if ((a != b) && (b != c) && (c != a)) {
        lbTip.setText("Escaleno");
    } else {
        lbTip.setText("Isóceles");
    }
} else {
    lbFor.setText("Não é um Triângulo");
    lbTip.setText("-----");
}
}

```

Exercício do "TRIANGULO".
 Equilátero, três lados iguais.
 Isósceles, dois lados iguais.
 Escaleno, todos lados diferentes.

11º Estrutura de Repetição 1ª Parte: “Enquanto”.

(WHILE)

(Repetição com teste no início).

```

//Contador com While...
int c = 0;
while (c<5){
    c++;
    System.out.print(+c+"º. ");
}

```

1º, 2º, 3º, 4º, 5º.

Mudança do Fluxo do Laço.

```

int c = 0;
while (c<5){
    c++;
    if (c ==2 || c==4){
        continue;
    }
    System.out.print(+c+"º. ");
}

```

1º, 3º, 5º.

Exercício de contagem usando Swing.

```

int c = 0;
String x = "";
while (c<=10){
    x += c + ". ";
    c++;
}
lbN.setText(x);

```

Contagem... 0. 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

Início

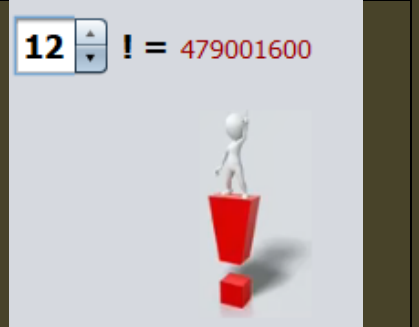
```

int c = 0;
while (c<100){
    c++;
    if (c==3 || c==5){
        continue;
    }
    if (c==9){
        break;
    }
    System.out.print(+c+"º. ");
}

```

1º, 2º, 4º, 6º, 7º, 8º.

```
private void spFStateChanged(javax.swing.event.ChangeEvent evt) {
    int n = Integer.parseInt(spF.getValue().toString());
    int f = 1;
    int c = n;
    while (c >= 1) {
        f *= c;
        c--;
    }
    lbF.setText(Integer.toString(f));
}
}
```



```
int n = 5;
int f = 1;
int c = n;

while (c >= 1) {
    f *= c;
    c--;
}

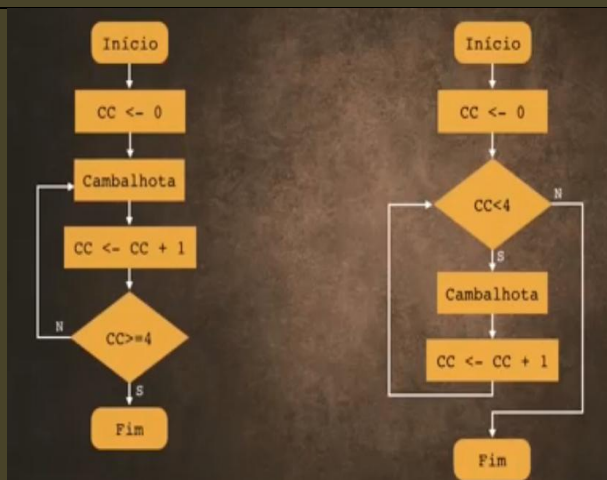
System.out.printf("O fatorial de "+n+" é: "+f+"\n");
```

```
public static void main(String[] args) {
    int n = 5;
    int f = 1;
    int c = 1;

    while (c <= n) {
        f *= c;
        c++;
    }

    System.out.printf("O fatorial de "+n+" é: "+f+"\n");
}
```

12º Estrutura de Repetição 2ª Parte: (DO, WHILE)



Estrutura de Repetição "Repita, Até" 2ª parte.

1º ex. (WHILE)

```
/* while (x <= 5){
    System.out.print(x+ ". ");
    x+=1;
}*/
```

2º ex. (DO ... WHILE)

```
/*do {
    System.out.print(x+ ". ");
    x += 1;
}while (x <= 5);*/
```

Estes exemplos acima não se aplica nos do lado esquerdo...
As duas alternativas necessitam que os resultados sejam **verdadeiros** para continuar o looping.

```
int l = 0;
do{
    System.out.print(l+ ". ");
    l++;
} while (l<=4);
```

0. 1. 2. 3. 4.

```
Scanner x = new Scanner(System.in);
int n, s = 0;
String r;
do {
    System.out.print("Dig nº para soma: ");
    n = x.nextInt();
    s += n;
    System.out.print("Continuar? [N/S]... ");
    r = x.next();
} while (r.equals("s") || r.equals("S"));
System.out.println("A soma dos valores é: "+s);
```

```

1 package exerciciorepita;
2
3 import javax.swing.JOptionPane;
4
5 public class ExercicioRepita {
6
7     public static void main(String[] args) {
8         JOptionPane.showMessageDialog(null, "Olá Denovo...");
9     }
10 }

```



Olá Denovo...

OK

```

JOptionPane.showMessageDialog(null, "E não é que consegui...", "Mensagem do S T A R K E.", JOptionPane.QUESTION_MESSAGE);

```

Mensagem do S T A R K E. X



E não é que consegui...

OK

```

int n = Integer.parseInt(JOptionPane.showInputDialog(null, "Dig. um nº; "));
JOptionPane.showMessageDialog(null, "VC. Dig. o nº "+n);

```



Dig. um nº;

OK

Cancelar



VC. Dig. o nº 6

OK

```

int n, s=0;
do{
    n = Integer.parseInt(JOptionPane.showInputDialog(null,
        "<html>Dig. nº para soma, <br><em>"+
        "(O nº 0 interrompe para realizar sua soma)</em></html> "));
    s+=n;
}while (n != 0);
JOptionPane.showMessageDialog(null, "<html>Resultado Final<hr>"+
    "<br>Somatória vale: "+s+".</html>");

```

Dig. nº para soma,
(O nº 0 interrompe para realizar sua soma)

20

OK

Cancelar



Resultado Final

Somatória vale: 20.

OK

```

import javax.swing.JOptionPane;

```

```

class ExJOption {

```

```

public static void main(String[] args) {
    JOptionPane.showMessageDialog(null, "Olá Denovo...");
}

```

Object message

Object message, Object initialValue

Component parentComponent, Object message

Component parentComponent, Object message, Object initialValue

Component parentComponent, Object message, String title, int messageType

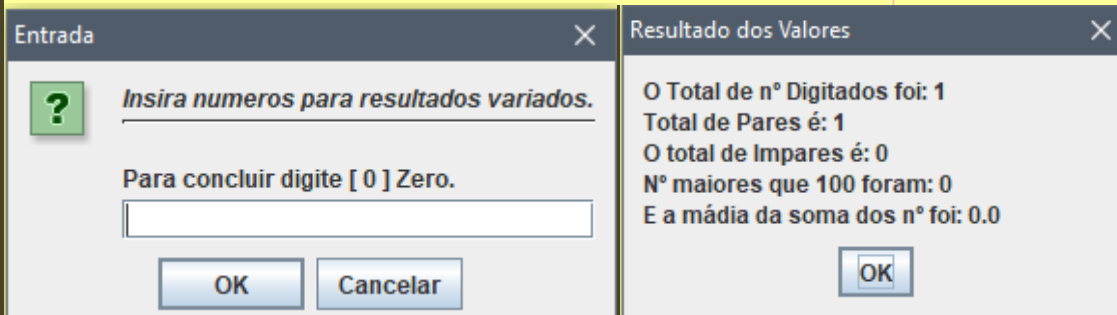
Component parentComponent, Object message, String title, int messageType, Icon icon, Object[] selectionValues, Object initialValue

args);

```

int n,tn=0,np=0,ni=0,nl00=0,s=0;
float m = 0;
do{
    n = Integer.parseInt(JOptionPane.showInputDialog("<html><em>Insira numeros para resultados variados.<em><hr><br>" +
    "Para concluir digite [ 0 ] Zero.</html>"));
    s += n;
    tn++;
    if(n%2==0){
        np++;
    }else{
        ni++;
    }
    if (n>100){
        nl00++;
    }
}while (n != 0);
tn-=1;
m = s/tn;
np-=1;
JOptionPane.showMessageDialog(null,"<html>O Total de n° Digitados foi: "+tn+
"<br>Total de Pares é: "+np+"<br>" +
"O total de Impares é: "+ni+"<br>N° maiores que 100 foram: "+nl00+"<br>" +
"E a média da soma dos n° foi: "+m+"</html>", "Resultado dos Valores",JOptionPane.PLAIN_MESSAGE);

```



13º Estrutura de Repetição 3ª Parte: (FOR)

```

for (int c = 0; c < 4; c++ ){
    System.out.print("Looping... ");
}

```

Looping... Looping... Looping... Looping...

```

for (int c = 15; c >= 5; c-=2){
    System.out.print(c+ "...");
}

```

15...13...11...9...7...5...

Ex.13º

```

import javax.swing.DefaultListModel;

int i = slIn.getValue();
int f = slFn.getValue();
int p = slPs.getValue();
DefaultListModel res = new DefaultListModel();
for (int l = i; l <= f; l += p){
    res.addElement(l);
}

lstRes.setModel(res);

```

Foi criado um Objeto chamado lista que é um DefaultListModel.


```
private void slInStateChanged(javax.swing.event.ChangeEvent evt) {
    int i = slIn.getValue();
    lbIn.setText(Integer.toString(i));
}

private void slFnStateChanged(javax.swing.event.ChangeEvent evt) {
    int f = slFn.getValue();
    lbFn.setText(Integer.toString(f));
}

private void slPsStateChanged(javax.swing.event.ChangeEvent evt) {
    int p = slPs.getValue();
    lbPs.setText(Integer.toString(p));
}
```

Início 1

Fim 10

Passo 1

Verificar

1
2
3
4
5
6
7
8
9
10

14º Variáveis Compostos (Vetores):

Algoritmos

var

n: vetor[0..3] de inteiro

início

n[0] <- 3

n[1] <- 5

n[2] <- 8

n[3] <- 2

Vetores em Algoritmos

```
int n[] = new int[4];
n[0] = 3;
n[1] = 5;
n[2] = 8;
n[3] = 2;
```

int n[] = {3,5,8,2};

Java

```
public static void main(String[] args) {
    int v[] = {5, 2, 8, 9};
    System.out.println("O nº total de vetores são: "+v.length);
    for (int l = 0; l <=3; l++){
        System.out.println("No vetor ["+l+"] ha o valor, "+v[l]);
    }
}
```

O nº total de vetores são: 4

No vetor [0] ha o valor, 5
No vetor [1] ha o valor, 2
No vetor [2] ha o valor, 8
No vetor [3] ha o valor, 9

```
int v[] = {5, 2, 8, 9};
System.out.println("O nº total de vetores são: "+v.length);
for (int l = 0; l<= v.length-1; l++){
    System.out.println("No vetor ["+l+"] ha o valor, "+v[l]);
}
```

Os dois algoritmos são iguais.

```
public static void main(String[] args) {
    String m[] = {"Jan.", "Fev.", "Mar.", "Abr.", "Maio", "Jun.",
        "Jul.", "Agos.", "Set.", "Otu.", "Nov.", "Dez."};
    int d[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    for (int l = 0; l < m.length; l++){
        System.out.print("O Mês "+m[l]+" têm "+d[l]+" dias.\n");
    }
}
```

O Mês Jan. têm 31 dias.
O Mês Fev. têm 28 dias.
O Mês Mar. têm 31 dias.
O Mês Abr. têm 30 dias.
O Mês Maio têm 31 dias.
O Mês Jun. têm 30 dias.
O Mês Jul. têm 31 dias.
O Mês Agos. têm 31 dias.
O Mês Set. têm 30 dias.
O Mês Otu. têm 31 dias.
O Mês Nov. têm 30 dias.
O Mês Dez. têm 31 dias.

(Matriz)

```
4
5 public class Filhos {
6
7     public static void main(String[] args) {
8         Scanner read = new Scanner(System.in);
9         int pessoas = 0, filhos = 0;
10
11         System.out.print("Quantas pessoas serão entrevistadas? ");
12         pessoas = read.nextInt();
13         String fam[][][] = new String[pessoas][][];
14
15         for (int i = 0; i < fam.length; i++) {
16             System.out.print(i + 1 + "ª Entrevistado:\nQuantos filhos você tem? ");
17             filhos = read.nextInt();
18             fam[i] = new String [filhos][];
19             for (int j = 0; j < fam[i].length; j++) {
20                 fam[i][j] = new String [2];
21                 for (int k = 0; k < fam[i][j].length; k++) {
22                     if (k == 0) {
23                         System.out.print("Nome do filho: ");
24                         fam[i][j][k] = read.next();
25                     } else if (k == 1) {
26                         System.out.print("Idade: ");
27                         fam[i][j][k] = read.next();
28                     }
29                 }
30             }
31         }
32         for (int i = 0; i < fam.length; i++) {
33             for (int j = 0; j < fam[i].length; j++) {
34                 for (int j2 = 0; j2 < fam[i][j].length; j2++) {
35                     System.out.println(fam[i][j][j2]);
36                 }
37             }
38         }
39         read.close();
40     }
41 }
42
```

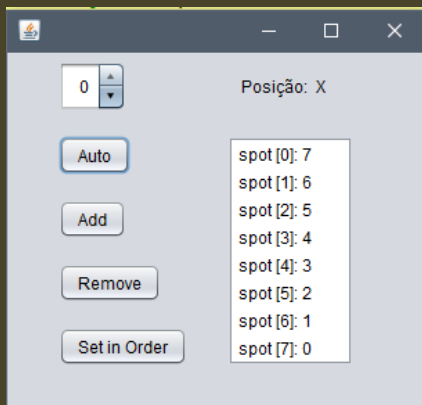
/* for(? X: Y) {...} */

<div>Para Cada...</div> <pre>num[] <table><tr><td>3</td><td>5</td><td>1</td><td>8</td><td>4</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> int num[] = {3, 5, 1, 8, 4}; for (int valor: num) { System.out.println(valor); }</pre>	3	5	1	8	4	0	1	2	3	4	<div>For Which One: Para cada elemento de [X] ponha em "n".</div> <pre>double x[] = {9.5, 5.2, -5, 3.7}; for (double n: x){ System.out.printf(n+" "); }</pre> <p>9.5. 5.2. -5.0. 3.7.</p>
3	5	1	8	4							
0	1	2	3	4							
<div>Colocando em Ordem</div> <pre>num[] <table><tr><td>3</td><td>5</td><td>1</td><td>8</td><td>4</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> int num[] = {3, 5, 1, 8, 4}; Arrays.sort(num); for (int valor: num) { System.out.println(valor); }</pre>	3	5	1	8	4	0	1	2	3	4	<pre>/* import Java.util.Arrays; */ double v[] = {5.7, -3.9, 7, 0.1}; Arrays.sort(v); for (double x: v){ System.out.print(x+".."); }</pre> <p>-3.9..0.1..5.7..7.0..</p>
3	5	1	8	4							
0	1	2	3	4							
<div>Busca</div> <pre>num[] <table><tr><td>3</td><td>5</td><td>1</td><td>8</td><td>4</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> int num[] = {3, 5, 1, 8, 4}; int pos = Arrays.binarySearch(num, 1); System.out.println(pos);</pre>	3	5	1	8	4	0	1	2	3	4	<pre>double vetor[] = {-9.7, 7, 8.3, 0, -1.99, 5, 10.3, 10}; for (double l: vetor){ System.out.print("["+l+" "); } System.out.println(""); int posicao = Arrays.binarySearch(vetor, 0); System.out.println("O nº 0 está no vetor: "+posicao);</pre> <p>[-9.7]. [7.0]. [8.3]. [0.0]. [-1.99]. [5.0]. [10.3]. [10.0]. O nº 0 está no vetor: 3</p>
3	5	1	8	4							
0	1	2	3	4							
<div>Preenchendo o Vetor</div> <pre>num[] <table><tr><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> int num[] = int[5]; Arrays.fill(num, 8); for (int valor: num) { System.out.println(valor); }</pre>	8	8	8	8	8	0	1	2	3	4	<pre>/* import Java.util.Arrays; */ int v[] = new int [13]; Arrays.fill(v, 0); for (int posicao: v){ System.out.print ("["+posicao+" "); }</pre> <p>[0]. [0]. [0]. [0]...</p>
8	8	8	8	8							
0	1	2	3	4							

```

1  package Classe;
2  import javax.swing.DefaultListModel;
3  import java.util.Arrays;
4
5  public class VeterSwing extends javax.swing.JFrame {
6      DefaultListModel lista = new DefaultListModel();
7      int vetor[] = new int[8];
8      int spot = 0;
9
10     public VeterSwing() {
11         initComponents();
12         for (int i = 0; i < vetor.length; i++){
13             lista.addElement("spot ["+i+"]: "+vetor[i]);
14         }
15     }
16
17     @SuppressWarnings("unchecked")
18     Generated Code
19
120
121     private void btAutoActionPerformed(java.awt.event.ActionEvent evt) {
122         lista.removeAllElements();
123         int i = 0;
124         for (int c = vetor.length-1; c >=0; c--){
125             vetor[i] = c;
126             lista.addElement("spot ["+i+"]: "+vetor[i]);
127             i++;
128         }
129     }
130
131     private void btAddActionPerformed(java.awt.event.ActionEvent evt) {
132         vetor[spot]= Integer.parseInt(spN.getValue().toString());
133         lista.removeAllElements();
134         for (int i = 0; i < vetor.length; i++){
135             lista.addElement("spot ["+i+"]: "+vetor[i]);
136         }
137     }
138
139     private void btRemActionPerformed(java.awt.event.ActionEvent evt) {
140         vetor[spot] = 0;
141         lista.removeAllElements();
142         for (int i = 0; i < vetor.length; i++){
143             lista.addElement("spot ["+i+"]: "+vetor[i]);
144         }
145     }
146
147     private void btOrdActionPerformed(java.awt.event.ActionEvent evt) {
148         Arrays.sort(vetor);
149         lista.removeAllElements();
150         for (int i = 0; i < vetor.length; i++){
151             lista.addElement("spot ["+i+"]: "+vetor[i]);
152         }
153     }
154
155     private void lsVetMouseClicked(java.awt.event.MouseEvent evt) {
156         spot = lsVet.getSelectedIndex();
157         lbSpot.setText("[ "+spot+" ]");
158     }

```

15º Métodos; “Rotinas: **Procedimento** e **Função**”.

PROCEDIMENTO

Método sem retorno de valor (void).

```

1 package metodo;
2
3 public class Metodo {
4
5     static void soma (int a, int b){
6         int s = a + b;
7         System.out.println("A soma é: " + s);
8     }
9
10    public static void main(String[] args) {
11        System.out.println("O programa começou...");
12        soma (5,2);
13    }
14
15 }

```

```

package exerciciossss;
import java.util.Scanner;

public class Exerciciossss {
    static Scanner read = new Scanner (System.in);

    static void call (int x){
        int y = x%2;
        if (y == 0){
            System.out.println("O n° "+x+" é PAR");
        }else{
            System.out.println("O n° "+x+" é IMPAR");
        }
        System.out.println("");
    }

    static void call2 (String a){
        System.out.print("Dig. outro n°: ");
        int n = read.nextInt();
        call(n);
    }

    public static void main(String[] args) {
        Scanner read = new Scanner(System.in);
        System.out.print("Dig. um n° e saiba se é par ou impar: ");
        int n = read.nextInt();
        call(n);
        call2 ("");
        call2 ("");
    }
}


```

O programa começou...

A soma é: 7

FUNÇÕES

Método com retorno de valor.



Algoritmos


Funções em Algoritmos

```
funcao Soma(A, B: Inteiro): Inteiro
var
    S: Inteiro
Inicio
    S <- A + B
    retorne S
fimfuncao

sm <- Soma(5,2)
```

```
int soma (int a,int b){
    int s = a + b;
    return s;
}

int sm = soma(5,2);
```



```
1 package metodo;
2
3 public class Metodo {
4     static int soma(int a, int b){
5         int r = a + b;
6         return r;
7     }
8
9     public static void main(String[] args) {
10        int sn = soma (5,2);
11        System.out.println("O resultado é:: "+ sn);
12    }
```

O resultado é:: 7

Comparação entre os Métodos

Procedimento Sem Retorno

```
1 package metodo;
2
3 public class Metodo {
4     static void soma(int x, int y){
5         int r = x + y;
6         System.out.println("O resultado é: "+r);
7     }
8
9     public static void main(String[] args) {
10        soma (5,2);
11    }
```

Função Com Retorno

```
1 package metodo;
2
3 public class Metodo {
4     static int soma(int x, int y){
5         int r = x + y;
6         return r;
7     }
8
9     public static void main(String[] args) {
10        int res = soma(5,2);
11        System.out.println("O resultado é: "+res);
12    }
```

```

1 package metodo;
2 import java.util.Scanner;
3
4 public class Metodo {
5     static void res(int a){
6         int r = a%2;
7         if (r == 1){
8             System.out.println("O nº "+a+" é Impar...");
9         }else{
10            System.out.println("O nº "+a+" é Par...");
11        }
12    }
13
14    public static void main(String[] args) {
15        Scanner x = new Scanner(System.in);
16        System.out.print("Dig. nº e saiba se é par ou impar: ");
17        int n = x.nextInt();
18        res(n);
19        System.out.println("");
20        System.out.print("Dig. outro nº:");
21        int nn = x.nextInt();
22        res (nn);
23    }
24 }

```

```

1 package metodo;
2 import java.util.Scanner;
3 public class Metodo {
4     static String res(int a){
5         int r = a%2;
6         if (r == 0){
7             return "Par";
8         }else{
9             return "Impar";
10        }
11    }
12
13    public static void main(String[] args) {
14        Scanner x = new Scanner(System.in);
15        System.out.print("Dig. nº e saiba se é Par ou Impar...");
16        int n = x.nextInt();
17        String q = res(n);
18        System.out.println("O nº "+n+" é "+q);
19        System.out.println("");
20        System.out.print("Dig outro nº: ");
21        int n1 = x.nextInt();
22        String q1 = res(n1);
23        System.out.println("O nº "+n1+" é "+q1);
24    }
25 }

```

MULTIPLAS CLASSES

```

package multiplasclasses;

public class MultiplasClasses {

    public static void main(String[] args) {
        System.out.println("Vai começar a contagem...");
        System.out.println(Operacao.contador(1,5));
    }
}

```

```

package multiplasclasses;

public class Operacao {

    public static String contador (int i, int f){
        String s = "";
        for (int c = 1; c <= f; c++){
            s += c + " ";
        }
        return s;
    }
}

```

Ex.15º (Método)

Função

```

package newbrainfitnessfator;
public class NewBrainFitNessFator {
    public static void main(String[] args) {
        Fator f = new Fator();
        f.setFatorial(3);
        System.out.println(f.getFormula()+f.getFatorial());
    }
}

```

3 x 2 x 1 = 6

Fatorial

```

package sergistarke;
public class Fator {
    private int tot = 1;
    private String form = "";

    public void setFatorial(int x){
        for(int i = x; i > 1; i--){
            this.tot*=i;
            this.form += i+" x ";
        }
        this.form += "1 = ";
    }

    public int getFatorial(){
        return this.tot;
    }

    public String getFormula(){
        return this.form;
    }
}

```

Função no Swing

```
private void btCalcActionPerformed(java.awt.event.ActionEvent evt) {
    int n = Integer.parseInt(spN.getValue().toString());
    Fator f = new Fator();
    f.setFatorial(n);
    lbFormula.setText(f.getFormula());
    lbFatorial.setText(Integer.toString(f.getFatorial()));
}
```



5 x 4 x 3 x 2 x 1 = 120

Fatorial

```
package sergistarke;
public class Fator {
    private int tot = 1;
    private String form = "";

    public void setFatorial(int x) {
        for(int i = x; i > 1; i--) {
            this.tot *= i;
            this.form += i + " x ";
        }
        this.form += "1 = ";
    }

    public int getFatorial() {
        return this.tot;
    }

    public String getFormula() {
        return this.form;
    }
}
```

Informações Complementares

ArrayList

Apresentação feita em formato de **Object**.

```
package test;
import java.util.ArrayList;
public class Test {
    public static void main(String[] args) {
        ArrayList nome = new ArrayList();
        nome.add("Sergio");
        nome.add("Felipe");
        nome.add("Starke");
        for (Object tot: nome) {
            System.out.println(tot);
        }
    }
}
```

Apresentação feita usando um conceito chamado **Genéricos**.

```
package test;
import java.util.ArrayList;
public class Test {
    public static void main(String[] args) {
        ArrayList<String> nome = new ArrayList();
        nome.add("Sergio");
        nome.add("Felipe");
        nome.add("Starke");
        for (String tot: nome) {
            System.out.println(tot);
        }
    }
}
```

Uso da **Operação Funcional**.

```
package test;
import java.util.ArrayList;
public class Test {
    public static void main(String[] args) {
        ArrayList<String> nome = new ArrayList();
        nome.add("Sergio");
        nome.add("Felipe");
        nome.add("Starke");
        nome.forEach((tot) -> {System.out.println(tot);});
    }
}
```

Controle Fluxo

Selection

If

Else

Switch

Case

Loop

While

Do

For

Exception

Throw

Try

Catch

Finally

Branch

Return

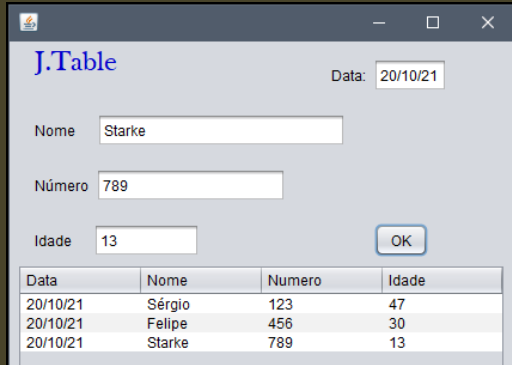
Break

Continue

Label

J.Table com Date

Adição Simples ao JTable



J.Table

Data: 20/10/21

Nome:

Número:

Idade:

Data	Nome	Numero	Idade
20/10/21	Sérgio	123	47
20/10/21	Felipe	456	30
20/10/21	Starke	789	13

```
package framejtable;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.table.DefaultTableModel;

public class JTable extends javax.swing.JFrame {

    DateFormat dateFormat = new SimpleDateFormat("dd/MM/YY");
    Date date = new Date();
    //Calendar cal = Calendar.getInstance();
    DefaultTableModel defaultTableModel;

    public JTable() {
        initComponents();//"table" é o JTable
        defaultTableModel = (DefaultTableModel) table.getModel();
        txData.setText(""+ dateFormat.format(date));
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void btOkActionPerformed(java.awt.event.ActionEvent evt) {
        defaultTableModel.insertRow(defaultTableModel.getRowCount(),new Object[]
        {txData.getText(),txNome.getText(),txNumero.getText(),txIdade.getText()});
    }
}
```

DEFINIÇÕES BÁSICAS

Classe Descreve um tipo particular de objeto. Ele pode conter métodos e variáveis relacionadas.

Método Uma função definida em uma classe. Métodos para implementar o comportamento de objectos.

Objeto Os principais blocos de construção de Java. Objectos consistem em variáveis (dados) e métodos (funcionalidade).

COMENTÁRIOS

HTML Comments

`<!-- Comentário -->` Enviado ao cliente no código fonte da página visível.

JSP comments (não enviado ao Cliente)

`<% - Comentário -%>` Comentários em arquivo JSP.

`// Comentário` Comentário em parte scriptlet de arquivo JSP.

ELEMENTOS

Declaração

`<%! declaração%>` Cria uma variável ou método global.

Expressão

`<% = Expressão%>` Demonstrações avaliadas no servidor antes que a página é passada para o cliente.

Directiva página

`<% @ Directiva%>` Os atributos que se aplicam a toda a página.

Scriptlet

`<Fragmento de código% de uma ou mais linhas%>` Contém um bloco de código de script que é executado quando a página é gerada.

Directiva Taglib

`<% @ Taglib uri = "URIToTagLibrary" prefix = "tagPrefix"%>` Define uma biblioteca de tags e prefixo para tags usadas em uma página JSP.

SEPARADORES

`()` Usado para cercar parâmetros

`[]` Define um bloco de código para uma classe ou método ou para conter os valores das matrizes inicializados automaticamente

`[]` Declara matrizes ou referências valores da matriz

`;` Indica o fim de uma instrução

`,` Separa variáveis

`.` Separa os nomes dos pacotes de subpackages / classes ou uma variável / método a partir de uma variável de referência

Tipos de dados primitivos		
Type	Descrição	Bits
	<i>(inteiros)</i>	
byte	Byte de comprimento inteiro	8
short	Inteiro curto	16
int	Número inteiro	32
long	Inteiro longo	64
	<i>(números reais)</i>	
float	De precisão simples de ponto flutuante	32
double	De precisão dupla de ponto flutuante	64
	<i>(Outros)</i>	
char	Um único caractere	16
boolean	Um valor booleano (verdadeiro ou falso)	1

OPERADORES	
+	somar e também concatenar uma String
+=	adicionar e atribuir números, concatenar e atribuir
-	subtração
-=	subtrair e atribuir
*	multiplicação
*=	multiplicar e atribuir
/	divisão
/=	dividir e atribuir
%	resto da divisão
%=	obter o resto da divisão e atribuir
++	i ++ equivale a i = i + 1
--	i -- equivale a i = i - 1
>	maior do que
>=	maior ou igual
<	menor que
<=	menor ou igual
!	NOT boolean
!=	diferente
&&	boolean AND
	booleano OR
==	boolean equals
=	atribuição