

CURSO JAVA

Curso em vídeo (Gustavo Guanabara)

https://www.youtube.com/playlist?list=PLHz_AreHm4dkl2ZdjTwZA4mPMxWTfNspR

Resumo do curso feito por Roberto Pinheiro

Primeiro Programa - Olá Mundo!

```
package olamundo;

public class OlaMundo {

    public static void main(String[] args) {

        System.out.println("Olá Mundo!");

    }

}
```

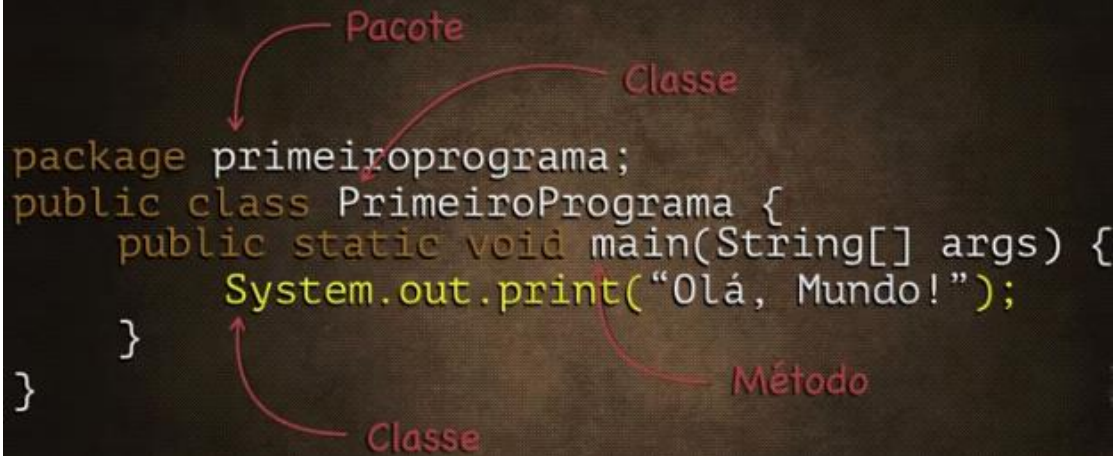
Dica: Usando o NetBeans:

- 1) digite psvm e pressione a tecla TAB, você terá: `public static void main(String[] args) {};`
- 2) digite sout e pressione a tecla TAB, você terá: `System.out.println("");`

O Java é case sensitive.

Nomes em Java

	Classe / Interface
Minha C lasse	AlunosCursoEmVideo
meu A tributo	nomeAluno
minha V ariável	mediaPrimeiroBimestre
meu M étodo	lancarNotaAluno
meu_pacote	alunoscursoemvideo
MINHA_CONSTANTE	VALOR_DE_PI



The image shows a snippet of Java code with four red labels and arrows pointing to specific parts of the code:

- Pacote**: Points to the `primeiroprograma` package name.
- Classe** (top): Points to the `PrimeiroPrograma` class name.
- Classe** (bottom): Points to the opening curly brace of the class body.
- Método**: Points to the `main` method.

```
package primeiroprograma;  
public class PrimeiroPrograma {  
    public static void main(String[] args) {  
        System.out.print("Olá, Mundo!");  
    }  
}
```

INTRODUÇÃO AO SWING E AO JAVA FX

Pacotes e Bibliotecas Java

Por ser uma linguagem multiplataforma, o Java tem que utilizar pacotes adicionais para que os seus arquivos não sejam sobrecarregados sem necessidade.

Para isso, é necessário utilizar a instrução “import” para carregar bibliotecas de pacotes adicionais.

Por padrão, apenas o pacote java.lang é carregado automaticamente. Porém, caso você precise carregar bibliotecas adicionais, utilize a importação desses pacotes.

Alguns pacotes Java

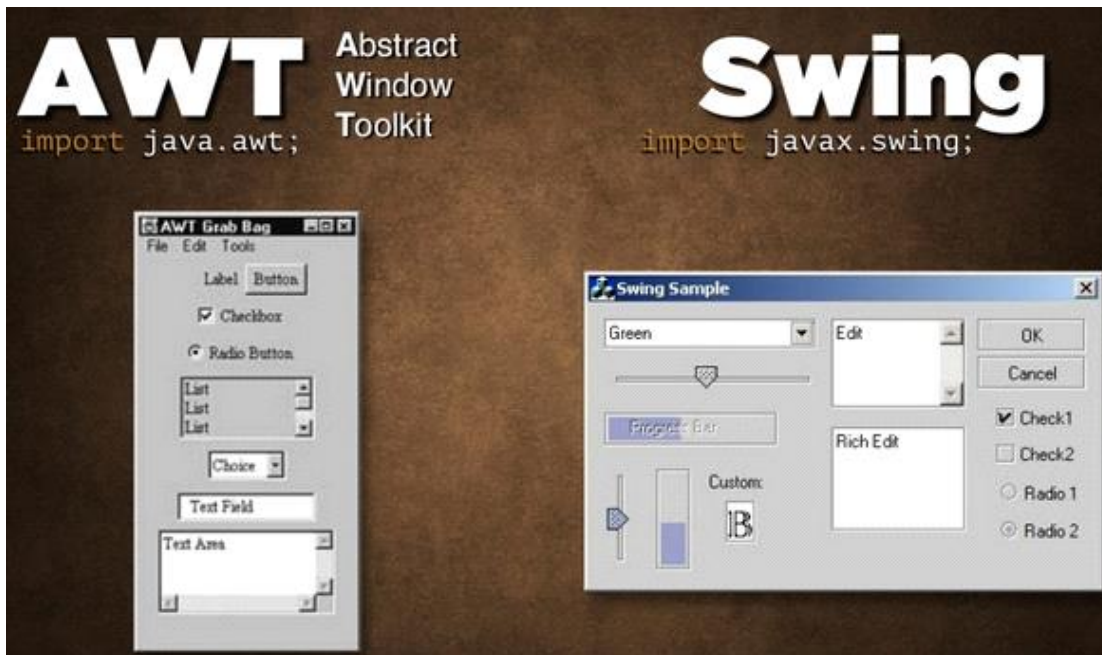
- java.applet
- java.util
- java.math
- java.net
- javax.sound
- javax.media
- javax.swing
- javafx.fxml

Biblioteca Swing

O Swing é uma evolução do antigo AWT (Abstract Windows Toolkit) que permite criar interfaces gráficas atraentes para qualquer sistema operacional baseado em janelas.

Plataforma JavaFX

Com o objetivo de substituir o Swing, o JavaFX cria interfaces para qualquer tipos de dispositivos, como ambientes de Janela, celulares e navegadores. Para a próxima versão, já está previsto o suporte para videogames, blu-rays e smart TVs.



1. Nome do Projeto: OlaMundoSwing
2. Desmarcar caixa: Criar Classe Principal

1. Criar Novo Arquivo
2. Forms GUI Swing --> Form JFrame
3. Nome da Classe: OlaMundoJava
4. Pacote: cursoemvideo.olamundo

```
package cursoemvideo.olamundo;
```

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

```
    private javax.swing.JButton btnClick;
```

```
    private javax.swing.JLabel lblMensagem;
```

```
    private void btnClickActionPerformed(java.awt.event.ActionEvent evt) {
```


```
        lblMensagem.setText("Olá, Mundo!");
```

```
    }
```

```
}
```

```
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




```
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



COMENTÁRIOS EM JAVA

```
// Este é um comentário de uma linha  
  
/*  
 * Nos comentários de muitas linhas,  
 * podemos escrever o quanto quiser  
 * até indicarmos o final do comentário  
 */
```


TIPOS PRIMITIVOS DO JAVA

O Java tem os tipos primitivos:

- Lógico: boolean
- Literal: char
- Inteiro: byte, short, int, long
- Real: float, double

Além disso existem as classes Wrapper para os tipos: Boolean, Character, String, Byte, Short, Integer, Long, Float e Double.

```
int idade = 3;
float sal = 1825.54f;
char letra = 'G';
boolean casado = false;

int idade = (int) 3;
float sal = (float) 1825.54;
char letra = (char) 'G';
boolean casado = (boolean) false;

Integer idade = new Integer(3);
Float sal = new Float(1825.54);
Character letra = new Character('G');
Boolean casado = new Boolean(false);
```

The diagram illustrates the relationship between primitive types and their wrapper classes in Java. It shows three ways to declare variables: directly with primitive types, with explicit typecasts, and using wrapper classes. Red arrows indicate the mapping from primitive types to their respective wrapper classes.

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) {

        String nome = "Roberto";

        float nota = 8.5f;

        System.out.println("A nota é " + nota );

        System.out.printf("A nota de %s é %.2f \n", nome, nota );

        System.out.format("A nota de %s é %.2f \n", nome, nota );

    }

}
```

ENTRADA DE DADOS

```
import java.util.Scanner;

Scanner teclado = new Scanner(System.in);

int idade = teclado.nextInt();
float salario = teclado.nextFloat();
String nome = teclado.nextLine();
```

```
package tiposprimitivos;

import java.util.Scanner;

public class TiposPrimitivos {

    public static void main(String[] args) {

        Scanner teclado = new Scanner (System.in);

        System.out.print("Digite o nome do aluno: ");

        String nome = teclado.nextLine();

        System.out.print("Digite a nota do aluno: ");

        float nota = teclado.nextFloat();

        System.out.format("A nota de %s é %.2f \n", nome, nota );

    }

}
```


INCOMPATIBILIDADE ENTRE NÚMEROS E STRING

Inteiro para String

```
package testetipos;

public class TesteTipos {

    public static void main(String[] args) {

        int idade = 30;

        String valor = Integer.toString(idade);

        System.out.println(valor);

    }

}
```

String para inteiro

```
package testetipos;

public class TesteTipos {

    public static void main(String[] args) {

        String valor = "30";

        int idade = Integer.parseInt(valor);

        System.out.println(idade);

    }

}
```

String para float

```
package testetipos;

public class TesteTipos {

    public static void main(String[] args) {

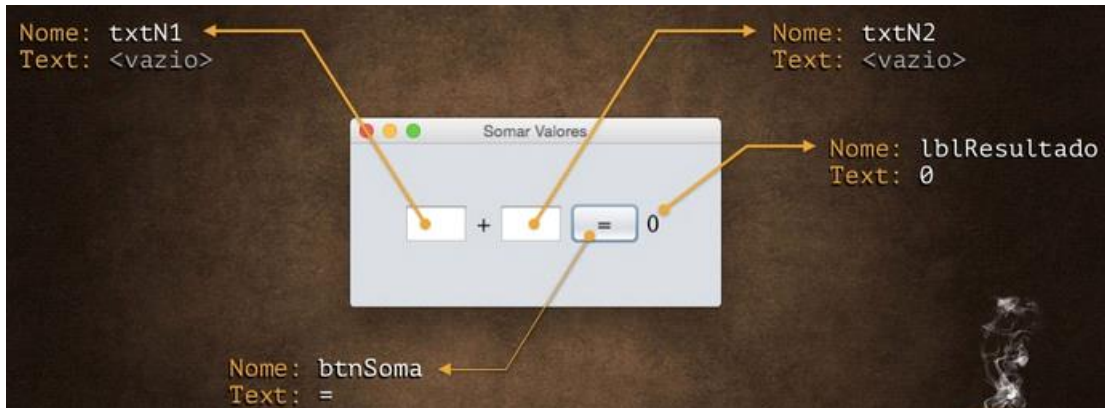
        String valor = "30.5";

        float idade = Float.parseFloat(valor);

        System.out.printf("%.3f", idade);  }

}
```

INTERFACE GRÁFICA COM O SWING



```
package com.cursoemvideo.soma;

public class TelaSoma extends javax.swing.JFrame {

    private void btnSomaActionPerformed(java.awt.event.ActionEvent evt) {

        int n1 = Integer.parseInt(txtN1.getText());

        int n2 = Integer.parseInt(txtN2.getText());

        int s = n1 + n2;

        lblSoma.setText(Integer.toString(s));

    }

}
```

OPERADORES ARITMÉTICOS

Operadores básicos

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

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        int n1 = 3;

        int n2 = 5;

        float m = (n1 + n2)/2;

        System.out.println("A média é igual a " + m);

    }

}
```

OPERADORES UNÁRIOS

Pós incremento

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        int numero = 5;

        int valor = 5 + numero++;

        System.out.println(valor);

    }

}

run:
10
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

Pré incremento

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        int numero = 5;

        int valor = 5 + ++numero;

        System.out.println(valor);

    }

}

run:
11
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

Pós decremento

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        int numero = 5;

        int valor = 5 + numero--;

        System.out.println(valor);

    }

}

run:
10
CONSTRUÍDO COM SUCESSO (tempo total: 3 segundos)
```

Pré decremento

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        int numero = 5;

        int valor = 5 + --numero;

        System.out.println(valor);

    }

}

run:
9
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```


OPERADORES DE ATRIBUIÇÃO

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

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        int x = 4;

        x += 2; // x = x + 2

        System.out.println(x);

    }

}

run:
6
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

Classe Math no Java

O Java não vem com todas as operações matemáticas automaticamente disponíveis. Por isso, é preciso carregar a biblioteca `java.Math` para realizar outras operações como exponenciação, raiz quadrada, etc.

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

Arredondamentos

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

```
package operadoresaritmeticos;
```

```
public class OperadoresAritmeticos {
```

```
    public static void main(String[] args) {
```

```
        float v = 8.9f;
```

```
        int ar = (int) Math.floor(v);
```

```
        System.out.println(ar);
```

```
    }
```

```
}
```

```
run:
```

```
8
```

```
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        float v = 8.9f;

        int ar = (int) Math.ceil(v);

        System.out.println(ar);

    }

}

run:
9
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        float v = 8.4f;

        int ar = (int) Math.round(v);

        System.out.println(ar);

    }

}

run:
8
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

Gerador de números (números aleatórios)

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        double ale = Math.random();

        System.out.println(ale);

    }

}

run:
0.4207426532594323
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

número inteiro aleatório de 15 a 50

```
package operadoresaritmeticos;

public class OperadoresAritmeticos {

    public static void main(String[] args) {

        double ale = Math.random();

        int n = (int) (15 + ale * (50 - 15));

        System.out.println(n);

    }

}

run:
46
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

OPERADOR TERNÁRIO

```
package operadorternario;

public class OperadorTernario {

    public static void main(String[] args) {

        int n1, n2, r;

        n1 = 4;

        n2 = 8;

        r = (n1 > n2) ? n1 : n2;

        System.out.println(r);

    }

}

run:
8
CONSTRUÍDO COM SUCESSO (tempo total: 2 segundos)
```

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

```
package comparacaostring;

public class ComparacaoString {

    public static void main(String[] args) {

        String nome1 = "Gustavo";

        String nome2 = "Gustavo";

        String nome3 = new String("Gustavo");

        String res;

        res = (nome1==nome2)?"Igual":"Diferente";

        System.out.println(res);

    }

}

run:
Igual
CONSTRUÍDO COM SUCESSO (tempo total: 2 segundos)
```

```
package comparacaostring;

public class ComparacaoString {

    public static void main(String[] args) {

        String nome1 = "Gustavo";

        String nome2 = "Gustavo";

        String nome3 = new String("Gustavo");

        String res;

        res = (nome1==nome3)?"Igual":"Diferente";

        System.out.println(res);

    }

}

run:
Diferente
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```



```
package comparacaostring;

public class ComparacaoString {

    public static void main(String[] args) {

        String nome1 = "Gustavo";

        String nome2 = "Gustavo";

        String nome3 = new String("Gustavo");

        String res;

        res = (nome1.equals (nome3))?"Igual":"Diferente";

        System.out.println(res);

    }

}

run:
Igual
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

OPERADORES LÓGICOS

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

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

p	! p
V	F
F	V

```

package operadoreslogicos;

public class OperadoresLogicos {

    public static void main(String[] args) {

        int x, y, z;

        x = 4;

        y = 7;

        z = 12;

        boolean r;

        r = ((x<y) && (y<z))?true:false;

        System.out.println(r);

    }

}

```

```
run:
true
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

```
package operadoreslogicos;

public class OperadoresLogicos {

    public static void main(String[] args) {

        int x, y, z;

        x = 4;

        y = 7;

        z = 12;

        boolean r;

        // r = ((x<y) && (y<z))?true:false;

        r = ((x<y) || (y==z))?true:false;

        System.out.println(r);

    }

}
```

```
run:
true
CONSTRUÍDO COM SUCESSO (tempo total: 3 segundos)
```

```
package operadoreslogicos;

public class OperadoresLogicos {

    public static void main(String[] args) {

        int x, y, z;

        x = 4;

        y = 7;

        z = 12;

        boolean r;

        // r = ((x<y) && (y<z))?true:false;
```

```

// r = ((x<y) || (y==z))?true:false;

r = ((x<y) ^ (y<z))?true:false;

System.out.println(r);
}
}

run:
false
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)

```

```

private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {

    int ano = Integer.parseInt(txtAno.getText());

    int idade = 2015 - ano;

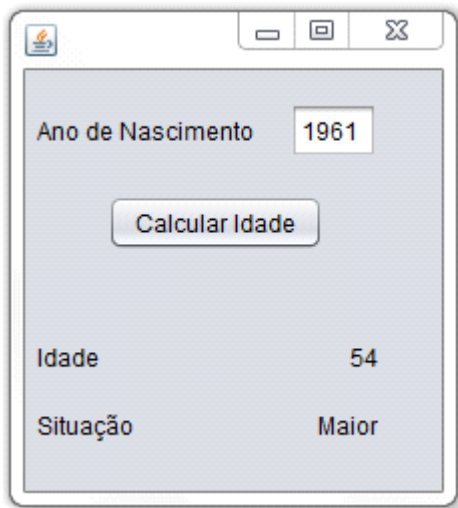
    lblIdade.setText(Integer.toString(idade));

    String sit = (idade >= 18) ? "Maior" : "Menor";

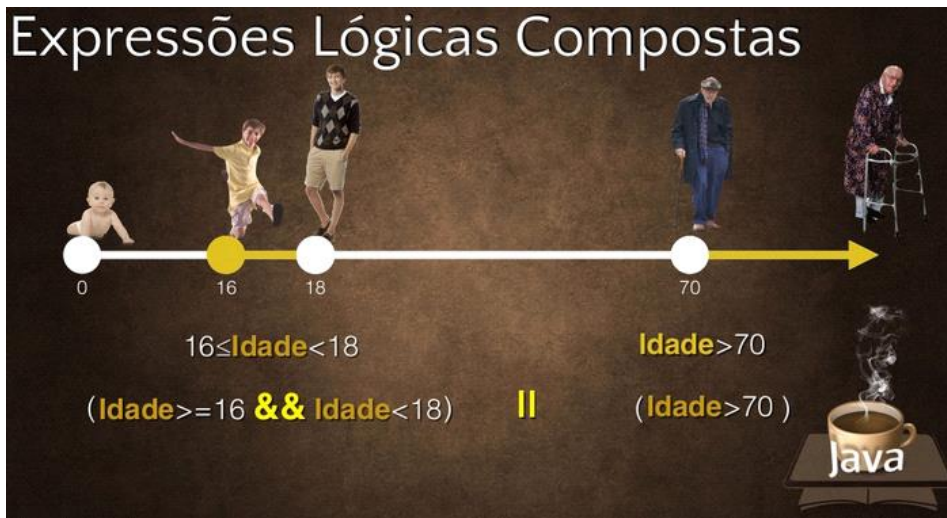
    lblSituacao.setText(sit);

}

```



EXPRESSÕES LÓGICAS COMPOSTAS



```
private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int ano = Integer.parseInt(txtAno.getText());  
    int idade = 2015 - ano;  
    lblIdade.setText(Integer.toString(idade));  
    String sit = ((idade >= 16 && idade < 18) || (idade > 70)) ? "É opcional" : "Não é opcional";  
    lblSituacao.setText(sit);  
}
```

Ano de Nascimento 1998

Calcular Idade

Idade 17

Situação É opcional

ESTRUTURAS CONDICIONAIS

Estrutura de controle condicional simples

```
package programamedia;

import java.util.Scanner;

public class ProgramaMedia {

    public static void main(String[] args) {

        Scanner teclado = new Scanner (System.in);

        System.out.print("Primeira nota: ");

        float n1 = teclado.nextFloat();

        System.out.print("Segunda nota: ");

        float n2 = teclado.nextFloat();

        float m = (n1 + n2)/2;

        System.out.println("Sua média é " + m);

        if (m > 9){

            System.out.println("Parabéns, pequeno gafanhoto!");

        }

    }

}
```


Estrutura de controle condicional composta

```
package programaidade;

import java.util.Scanner;

public class Programaldade {

    public static void main(String[] args) {

        Scanner t = new Scanner (System.in);

        System.out.print("Em que ano você nasceu? ");

        int nasc = t.nextInt();

        int i = 2015 - nasc;

        System.out.println("Sua idade é " + i);

        if (i >= 18){

            System.out.println("Maior de idade");

        } else {

            System.out.println("Menor de idade");

        }

    }

}
```

ESTRUTURA CONDICIONAL COMPOSTA ENCADEADA

if

```
private void btnVotoActionPerformed(java.awt.event.ActionEvent evt) {  
    panResultado.setVisible(true);  
    int a = Integer.parseInt(txtAno.getText());  
    int i = 2015 - a;  
    lblIdade.setText(String.format("%d", i) + " anos");  
  
    if (i < 16){  
        lblR.setText("Não pode votar!");  
    } else {  
        if ((i >= 16 && i < 18) || (i > 70)){  
            lblR.setText("Voto Opcional");  
        } else {  
            lblR.setText("Voto Obrigatório");  
        }  
    }  
}
```

else if

```
private void btnVotoActionPerformed(java.awt.event.ActionEvent evt) {  
    panResultado.setVisible(true);  
    int a = Integer.parseInt(txtAno.getText());  
    int i = 2015 - a;  
    lblIdade.setText(String.format("%d", i) + " anos");  
  
    if (i < 16){
```

```
    lblR.setText("Não pode votar!");  
} else if ((i >= 16 && i < 18) || (i > 70)){  
    lblR.setText("Voto Opcional");  
} else {  
    lblR.setText("Voto Obrigatório");  
}  
}
```

CONDIÇÃO DE MÚLTIPLA ESCOLHA

swing

```
int pernas = teclado.nextInt();
String tipo;
switch (pernas) {
    case 1:
        tipo = "Saci";
        break;
    case 2:
        tipo = "Bípede";
        break;
    case 4:
        tipo = "Quadrúpede";
        break;
    case 6,8:
        tipo = "Aranha";
        break;
    default:
        tipo = "ET";
}
System.out.print(tipo);
```

REPETIÇÃO COM TESTE LÓGICO NO INÍCIO

Comando while

```
package contador01;

public class Contador01 {

    public static void main(String[] args) {

        int cc = 1;

        while (cc <= 4) {

            System.out.println("Cambalhota " + cc);

            cc++;

        }

    }

}
```

```
run:
Cambalhota 1
Cambalhota 2
Cambalhota 3
Cambalhota 4
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

Comando continue

```
package contador01;

public class Contador01 {

    public static void main(String[] args) {

        int cc = 0;

        while (cc < 10) {

            cc++;

            if (cc==5 || cc==7 || cc==9){

                continue;

            }

            System.out.println("Cambalhota " + cc);

        }

    }

}
```

```
run:
Cambalhota 1
Cambalhota 2
Cambalhota 3
Cambalhota 4
Cambalhota 6
Cambalhota 8
Cambalhota 10
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

Comando break

```
package contador01;

public class Contador01 {

    public static void main(String[] args) {

        int cc = 0;

        while (cc < 10) {

            cc++;

            if (cc==2 || cc==3 || cc==4){

                continue;

            }

            if (cc==7){

                break;

            }

            System.out.println("Cambalhota " + cc);

        }

    }

}

run:
Cambalhota 1
Cambalhota 5
Cambalhota 6
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```


REPETIÇÃO COM TESTE LÓGICO NO FINAL

```
package repeticao01;

public class Repeticao01 {

    public static void main(String[] args) {

        int cc = 0;

        do {

            System.out.println("Cambalhota");

            cc++;

        } while (cc < 4);

    }

}

run:
Cambalhota
Cambalhota
Cambalhota
Cambalhota
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

```
package numeros;
```

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

```
public class Numeros {
```

```
    public static void main(String[] args) {
```

```
        int n, s = 0;
```

```
        String resp;
```

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

```
        do {
```

```
            System.out.print("Digite um número: ");
```

```
            n = teclado.nextInt();
```

```
            s += n;
```

```
            System.out.println("Quer continuar? (S/N): ");
```

```
            resp = teclado.next();
```

```
        } while (resp.equals("S"));
```

```
        System.out.println("A soma de todos os valores é " + s);
```

```
    }
```

```
}
```

javax.swing.JOptionPane

```
package exerciciorepita;

import javax.swing.JOptionPane;

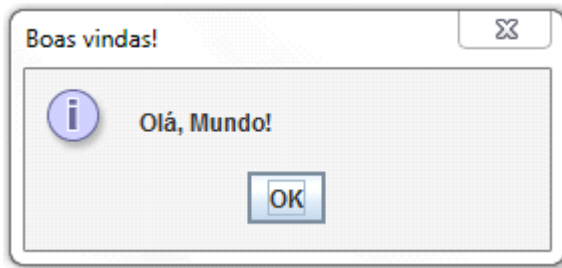
public class ExercicioRepita {

    public static void main(String[] args) {

        JOptionPane.showMessageDialog(null, "Olá, Mundo!", "Boas vindas!", JOptionPane.INFORMATION_MESSAGE);

    }

}
```



```
package exerciciorepita;

import javax.swing.JOptionPane;

public class ExercicioRepita {

    public static void main(String[] args) {


        int n = Integer.parseInt(JOptionPane.showInputDialog(null, "Informe um número: "));

        JOptionPane.showMessageDialog(null, "Você digitou o valor " + n);

    }

}
```


Entrada

 Informe um número:

54

OK Cancelar

Mensagem

 Você digitou o valor 54

OK

REPETIÇÃO COM VARIÁVEL DE CONTROLE

for

```
package repeticaofor;

public class RepeticaoFor {

    public static void main(String[] args) {

        for(int cc = 0; cc <= 3; cc++){

            System.out.println("Cambalhota");

        }

    }

}

run:
Cambalhota
Cambalhota
Cambalhota
Cambalhota
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

VETORES EM JAVA

Em Java, o vetor é um objeto.

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

ou

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

```

package vetor01;

public class Vetor01 {

    public static void main(String[] args) {

        // int n[] = new int[4];

        int n[] = {3,2,8,7,5,4};

        for(int c = 0; c <= 5; c++){

            System.out.print(n[c] + " ");

        }

    }

}

run:
3 2 8 7 5 4 CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)

```

```

package vetor01;

public class Vetor01 {

    public static void main(String[] args) {

        int n[] = {3,2,8,7,5,4};

        for(int c = 0; c <= 5; c++){

            System.out.println("Na posição " + c + " temos o valor " + n[c]);

        }

    }

}

run:
Na posição 0 temos o valor 3
Na posição 1 temos o valor 2
Na posição 2 temos o valor 8
Na posição 3 temos o valor 7
Na posição 4 temos o valor 5
Na posição 5 temos o valor 4
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)

```

Tamanho de um vetor (length)

```
package vetor01;

public class Vetor01 {

    public static void main(String[] args) {

        int n[] = {3,2,8,7,5,4};

        System.out.println("Total de casas de n: " + n.length);

    }

}
```

```
run:
Total de casas de n: 6
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

```
package vetor01;

public class Vetor01 {

    public static void main(String[] args) {

        int n[] = {3,2,8,7,5,4};

        System.out.println("Total de casas de n: " + n.length);

        for(int c = 0; c <= n.length - 1; c++){

            System.out.println("Na posição " + c + " temos o valor " + n[c]);

        }

}
```

```
run:
Total de casas de n: 6
Na posição 0 temos o valor 3
Na posição 1 temos o valor 2
Na posição 2 temos o valor 8
Na posição 3 temos o valor 7
Na posição 4 temos o valor 5
Na posição 5 temos o valor 4
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

```

package vetor02;

public class Vetor02 {

    public static void main(String[] args) {

        String mes[] = {"Janeiro","Fevereiro","Março","Abril","Maio","Junho",
                        "Julho","Agosto","Setembro","Outubro","Novembro","Dezembro"};

        int tot[] = {31,28,31,30,31,30,31,31,30,31,30,31};

        for (int c = 0; c < mes.length; c++){

            System.out.println("O mês de " + mes[c] + " tem " + tot[c] + " dias");

        }

    }

}

```

```

run:
O mês de Janeiro tem 31 dias
O mês de Fevereiro tem 28 dias
O mês de Março tem 31 dias
O mês de Abril tem 30 dias
O mês de Maio tem 31 dias
O mês de Junho tem 30 dias
O mês de Julho tem 31 dias
O mês de Agosto tem 31 dias
O mês de Setembro tem 30 dias
O mês de Outubro tem 31 dias
O mês de Novembro tem 30 dias
O mês de Dezembro tem 31 dias
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)

```


Lendo vetores com foreach

```
package vetor03;

public class Vetor03 {

    public static void main(String[] args) {

        double v[] = {3.5, 2.75, 0, -4.5};

        for (double valor: v){

            System.out.print(valor + " ");

        }

    }

}

run:
3.5 2.75 0.0 -4.5 CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

Colocando em ordem o vetor

```
package vetor03;

import java.util.Arrays;

public class Vetor03 {

    public static void main(String[] args) {

        double v[] = {3.5, 2.75, 0, -4.5};

        Arrays.sort(v);

        for (double valor: v){

            System.out.print(valor + " ");

        }

    }

}
```

```
run:
-4.5 0.0 2.75 3.5 CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

Busca binária

```
package vetor04;
```

```
import java.util.Arrays;
```

```
public class Vetor04 {

    public static void main(String[] args) {

        int vet[] = {3, 7, 6, 1, 9, 4, 2};

        int busca = 1;

        for(int v:vet){

            System.out.print(v + " ");

        }

        int p = Arrays.binarySearch(vet, busca);

        if (p < 0){

            System.out.println("\nValor " + busca + " não encontrado!");

        } else {

            System.out.println("\nEncontrei o valor " + busca + " na posição " + p);

        }

    }

}
```

```
run:
3 7 6 1 9 4 2
Encontrei o valor 1 na posição 3
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

```

public class Vetor04 {

    public static void main(String[] args) {

        int vet[] = {3, 7, 6, 1, 9, 4, 2};

        int busca = 8;

        for(int v:vet){

            System.out.print(v + " ");

        }

        int p = Arrays.binarySearch(vet, busca);

        if (p < 0){

            System.out.println("\nValor " + busca + " não encontrado!");

        } else {

            System.out.println("\nEncontrei o valor " + busca + " na posição " + p);

        }

    }

}

```

```

run:
3 7 6 1 9 4 2
Valor 8 não encontrado!
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)

```

Preenchendo todas as posições do vetor com um mesmo valor

```
package vetor05;

import java.util.Arrays;

public class Vetor05 {

    public static void main(String[] args) {

        int v[] = new int[20];

        Arrays.fill(v, 0);

        for(int valor: v){

            System.out.print(valor + " ");

        }

    }

}

run:
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

MÉTODOS EM JAVA

Procedimentos

Não retornam valor.

```
package testefuncao;

public class TesteFuncao {

    static void soma(int a, int b){

        int s = a + b;

        System.out.println("A soma é " + s);

    }

    public static void main(String[] args) {

        soma(5, 19);

    }

}
```

run:
A soma é 24
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)

Funções

Retornam valor.

```
package testefuncao;

public class TesteFuncao {

    static int soma(int a, int b){

        int s = a + b;

        return s;

    }

    public static void main(String[] args) {

        System.out.println("Começou o programa");

        int sm = soma(23, 18);

        System.out.println("A soma é " + sm);

    }

}

run:
Começou o programa
A soma é 41
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

Chamada a uma função externa

```
package testefuncao02;

public class Operacoes {

    public static String contador (int i, int f){

        String s = "";

        for(int c = i; c <= f; c++){

            s += c + " ";

        }

        return s;

    }

}
```

```
package testefuncao02;

public class TesteFuncao02 {

    public static void main(String[] args) {

        System.out.println("Vai começar a contagem");

        System.out.println(Operacoes.contador(1,5));

    }

}
```

```
run:
Vai começar a contagem
1 2 3 4 5
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```