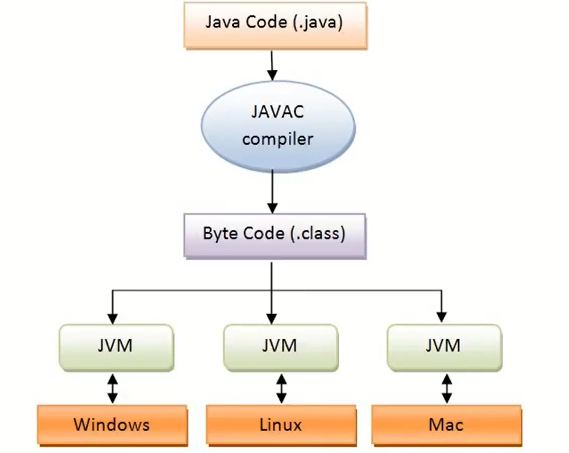
**Curso de Java**

**João Ribeiro (Sys4Soft)**

# Aula 01 - Introdução



## Como funciona o Java

1) Escrevemos o código da aplicação gravando em arquivos .java;

2) O código é compilado pelo JAVAC e passa para Byte Code (.class);

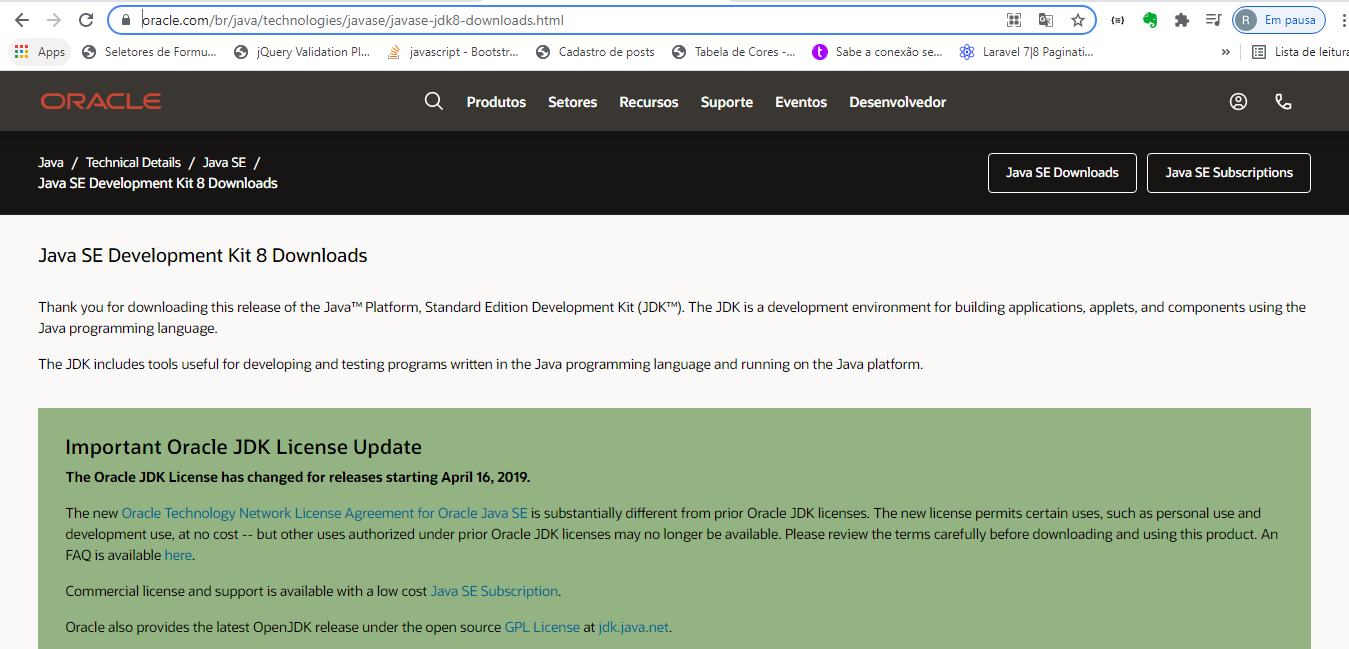
3) O Byte Code é executado sob uma Java Virtual Machine que funciona em diferentes sistemas operacionais.

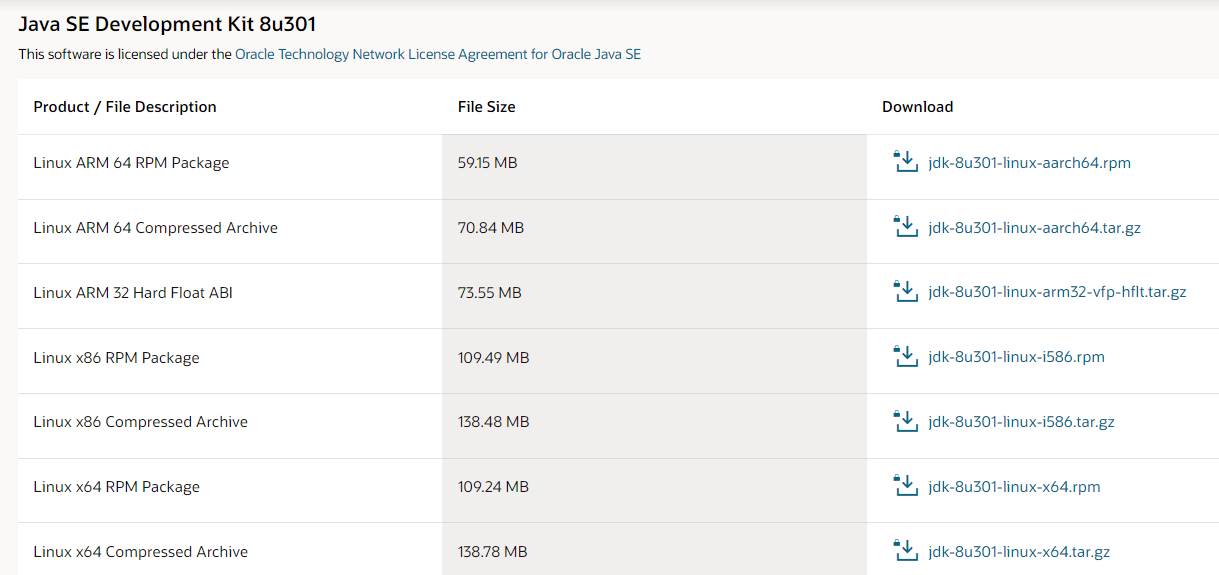
# Aula 02 - Instalação do JDK e Eclipse

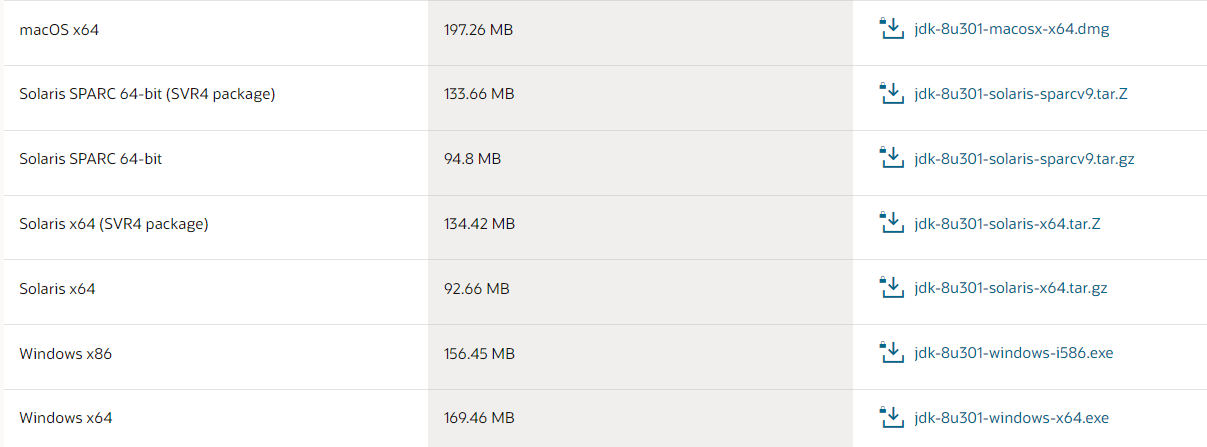
## Instalação do JDK (Java Development Kit)

Java Development Kit é o pacote de software que contém o necessário para desenvolver aplicações em Java.

https://www.oracle.com/br/java/technologies/javase/javase-jdk8-downloads.html







## Eclipse

Eclipse é um IDE (Interface Develpment Environment) para escrever código Java.

www.eclipse.org

https://eclipse.br.uptodown.com/windows/versions



- Baixar a versão 4.4.2 (64 bit)

Eclipse IDE for Java Developers

- Copie a pasta Eclipse para C:\Program Files

- Crie um atalho para Eclipse.exe na área de trabalho

# Aula 04 - Primeiro programa em Java

- Crie a pasta C:\java\_sys4soft

- Dentro dela crie com o Notepad++ o arquivo EstouAqui.java

**EstouAqui.java**

public class EstouAqui {

public static void main(String[] args){

System.out.println("Estou Aqui!");

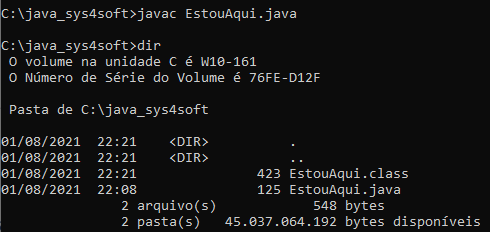
}

}

- Compile com:

javac EstouAqui.java

- Será criado o arquivo EstouAqui.class



- Para executá-lo, entre com:

java EstouAqui



# Aula 05 - Criação de arquivo JAR

- Um arquivo com extensão JAR é um arquivo executável de JAVA.

**MeuProjeto.java**

public class MeuProjeto {

public static void main(String[] args){

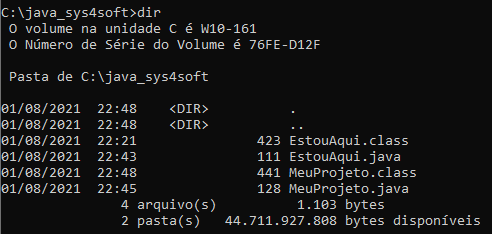
System.out.println("O MEU NOVO PROJETO EM JAVA!");

}

}

javac MeuProjeto.java

Será criado o arquivo MeuProjeto.class



- Crie um arquivo chamado manifest.txt

**manifest.txt**

Main-Class: MeuProjeto

Atenção: é necessário clicar na tecla <Enter> após digitar o texto acima (caso contrário não irá funcionar).

## Criando o arquivo .JAR

jar cvfm MeuProjeto.jar manifest.txt MeuProjeto.class



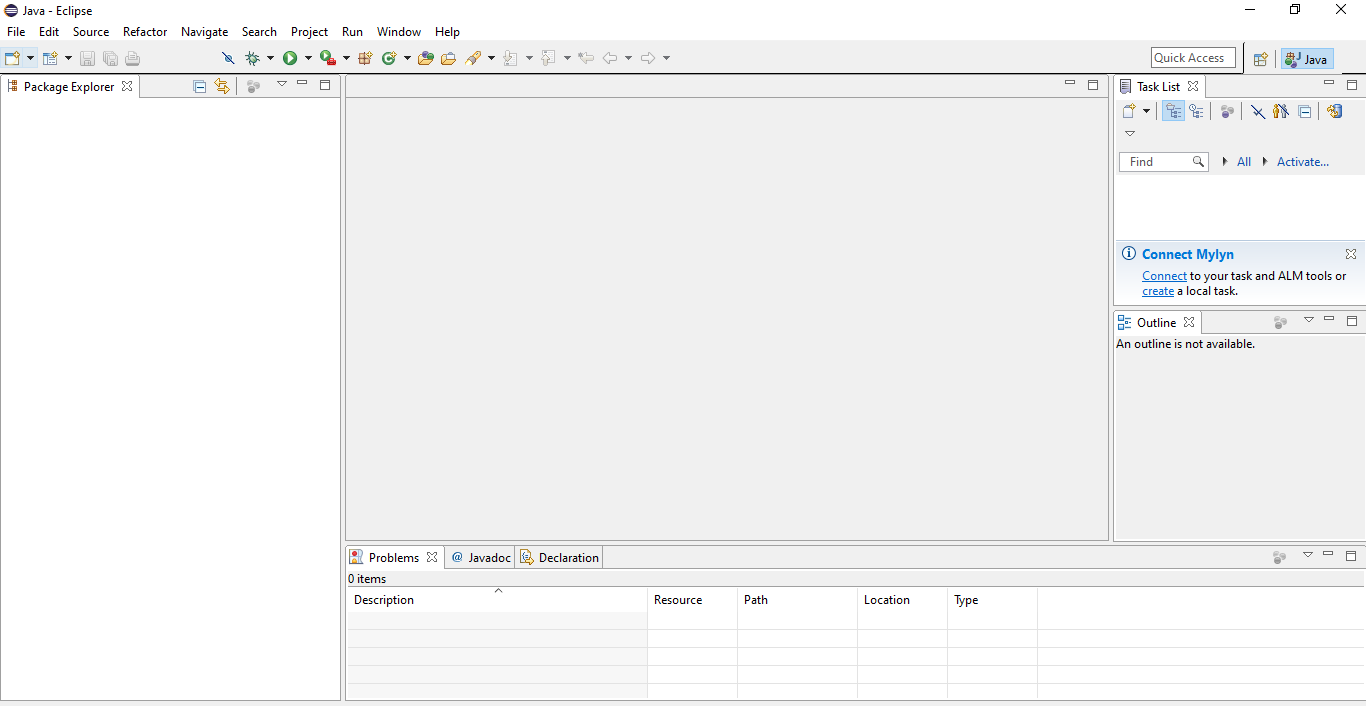
- Para executar o arquivo .JAR:

java -jar MeuProjeto.jar



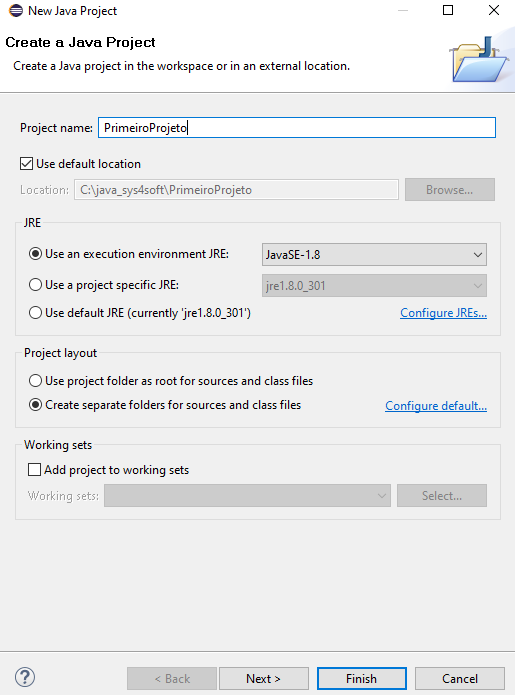
# Aula 06 - Criando o primeiro projeto Java no Eclipse

- Abra o Eclipse

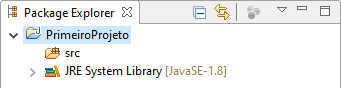


## Criando um novo projeto

File -> New -> Java Project



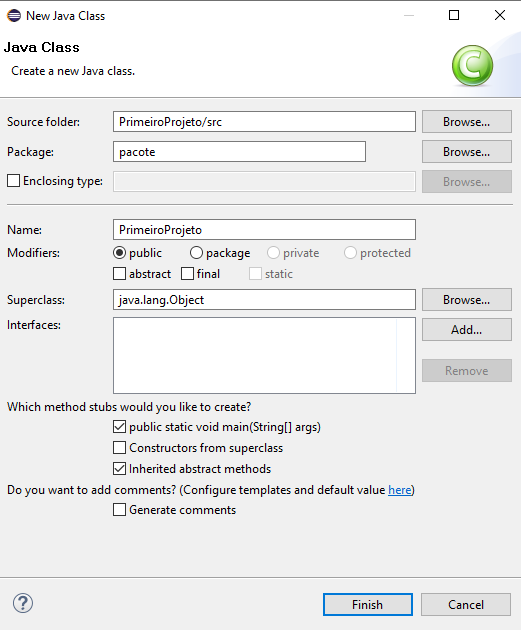
- Clique no botão "Finish"



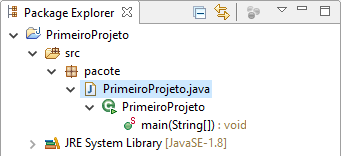
## Acrescentando uma classe ao projeto

- Clique com o botão direito do mouse sobre a pasta src e selecione:

New -> Class



- Clique no botão "Finish"



**PrimeiroProjeto.java**

package pacote;

public class PrimeiroProjeto {

public static void main(String[] args) {

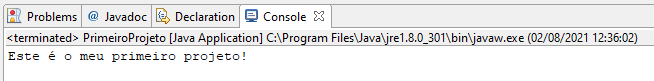
System.out.println("Este é o meu primeiro projeto!");

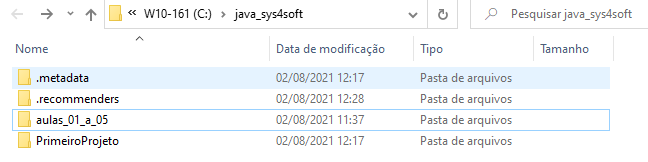
}

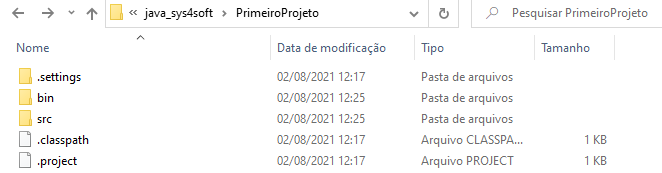
}

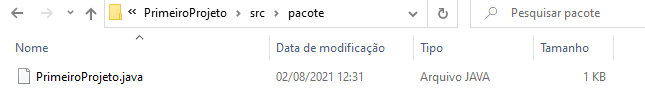
## Executando o projeto

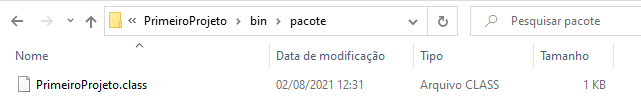
- Clique no botão "Run": 







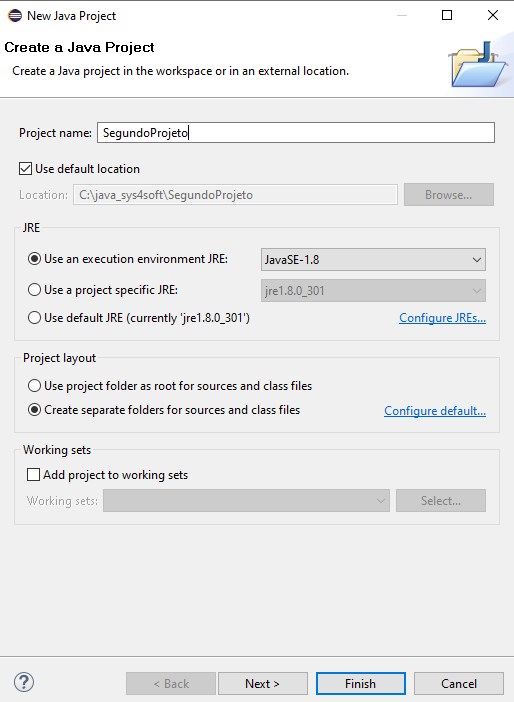


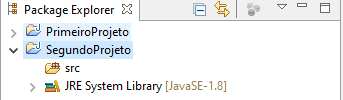


# Aula 07 - Criando o segundo projeto Java no Eclipse

- Crie um novo projeto:

File -> New -> Java Project





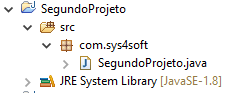
- Clique com o botão direito do mouse sobre a pasta src e crie uma nova classe:

New -> Class

Package: com.sys4soft

Name: SegundoProjeto

Marque a caixa: public static void main (String[] args)



**SegundoProjeto.java**

package com.sys4soft;

public class SegundoProjeto {

public static void main(String[] args) {

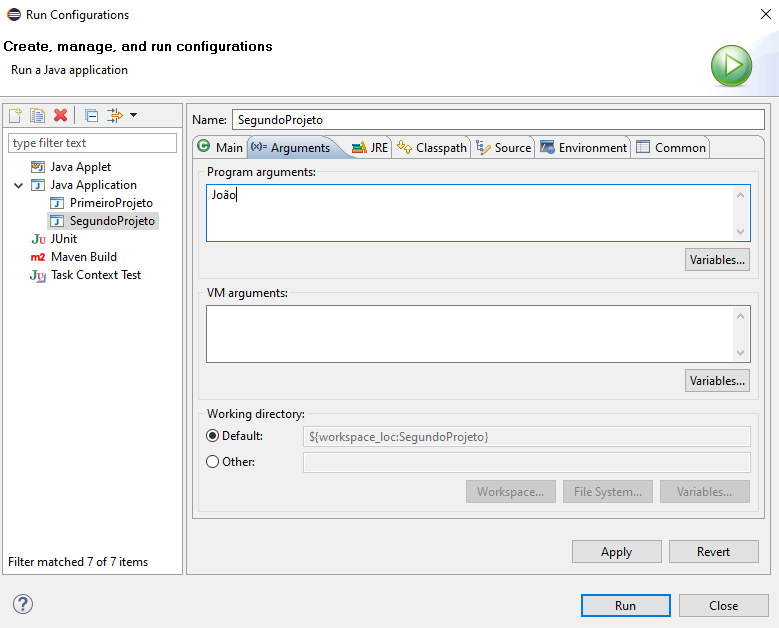
System.out.println("Bom dia, " + args[0] + "!");

}

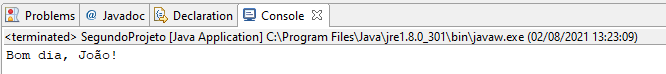
}

- Clique com o botão direito do mouse sobre "SegundoProjeto.java" e selecione: "Run As -> Run Configurations"

- Clique na aba "Arguments" e no campo "Program arguments" insira o nome da pessoa.



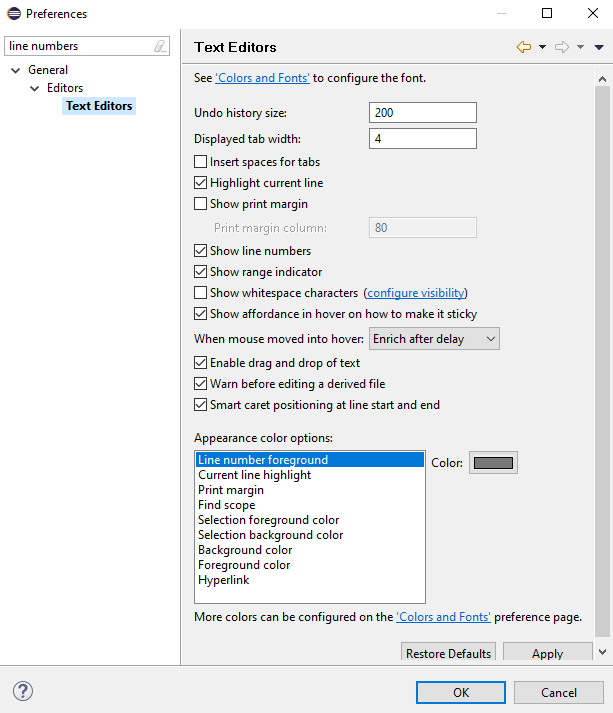
- Em seguida clique no botão "Run"



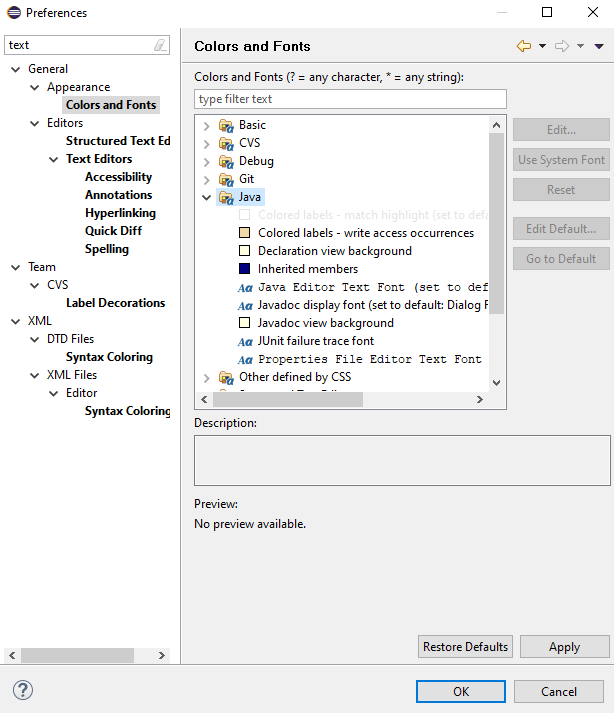
# Aula 08 - Configurando o Eclipse

## Exibindo o número das linhas

Window -> Preferences

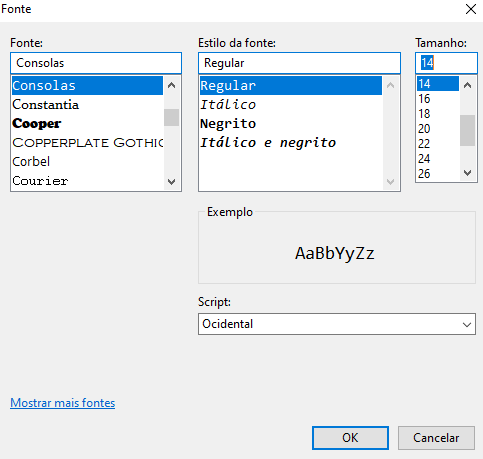


## Alterando a fonte



- Clique em "Java Editor Text Font"

- Altere o tipo de fonte para "Consolas" (é monospaced) e tamanho de 10 para 14:

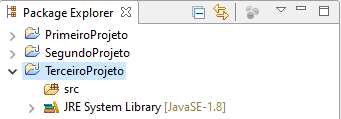


# Aula 09 - Terceiro projeto Java no Eclipse

- Crie um novo projeto

File -> New -> Java Project

- Nomeie como "TerceiroProjeto"



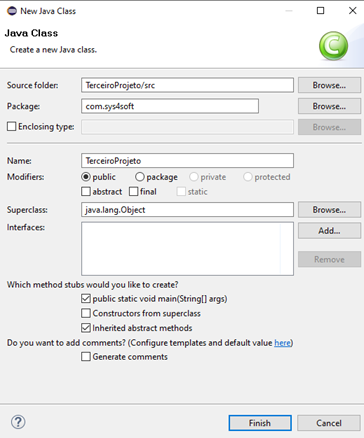
- Clique com o botão direito do mouse sobre a pasta src e crie uma nova classe:

New -> Class

Package: com.sys4soft

Name: TerceiroProjeto

Marque a caixa: public static void main (String[] args)



**TerceiroProjeto.java**

package com.sys4soft;

import java.util.Scanner;

public class TerceiroProjeto {

public static void main(String[] args) {

// fazer a adição de duas parcelas

System.out.println("Adicionar duas parcelas.");

// definir variáveis

int parcela1;

int parcela2;

int resultado;

// criar o objeto de inserção de parcelas

Scanner input = new Scanner(System.in);

// pedido da primeira parcela

System.out.println("Informe o valor da primeira parcela");

parcela1 = input.nextInt();

// pedido da segunda parcela

System.out.println("Informe o valor da segunda parcela");

parcela2 = input.nextInt();

// realização da adição

resultado = parcela1 + parcela2;

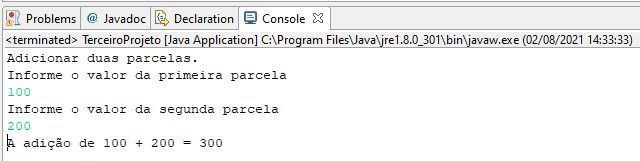
// apresentação do resultado

System.out.printf("A adição de %d + %d = %d", parcela1, parcela2, resultado);

}

}

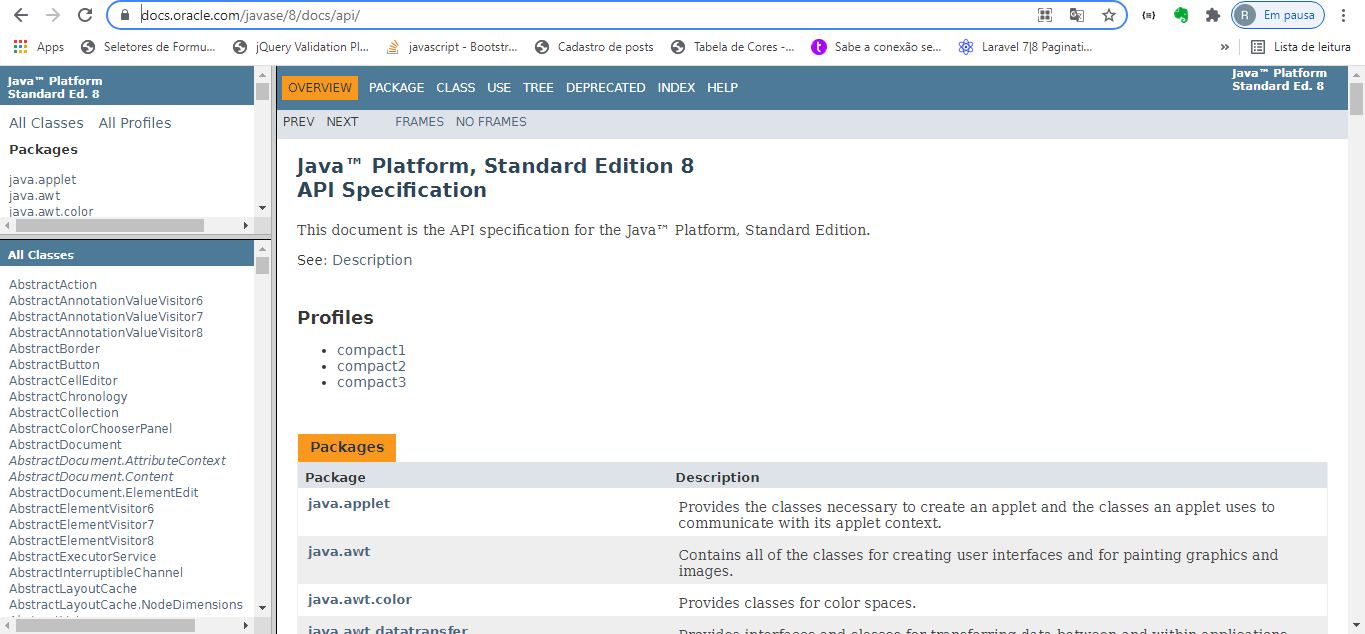
- Rodando (Run):



# Aula 10 - Analisando o projeto e código Java - Parte1

## Documentação Java SE8

https://docs.oracle.com/javase/8/docs/api/



- A classe java.lang é importada implicitamente para qualquer classe dentro do Java e providencia funcionalidades que são fundamentais para a execução do código Java.

# Aula 12 - Comentários no código Java

// Comentário de uma linha

/\* Comentário de várias linhas

\* Linha 1

\* Linha 2

\* Linha 3

\*/

/\*\*

Classe para guardar os dados e comportamentos do aluno

\*/

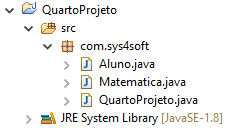
# Aula 13 - Breve introdução às classes em Java

Uma classe é um modelo a partir do qual podemos criar objetos.

- Crie um novo projeto 🡪 QuartoProjeto

- Na pasta src adicione duas classes:

1. QuartoProjeto com "public static void main(String[] args)"
2. Aluno sem "public static void main(String[] args)"



**QuartoProjeto.java**

package com.sys4soft;

public class QuartoProjeto {

public static void main(String[] args) {

Aluno aluno\_1 = new Aluno();

aluno\_1.nome = "João";

aluno\_1.idade = 20;

Aluno aluno\_2 = new Aluno();

aluno\_2.nome = "Carlos";

aluno\_2.idade = 25;

// Apresentando os alunos

aluno\_1.ApresentaAluno();

aluno\_2.ApresentaAluno();

}

}

**Aluno.java**

package com.sys4soft;

public class Aluno {

public String nome;

public int idade;

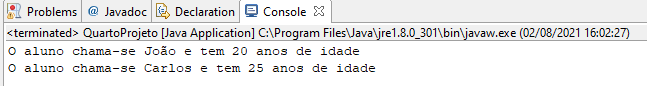
public void ApresentaAluno(){

System.out.println("O aluno chama-se " + nome + " e tem " + idade + " anos de idade");

}

}

- Rodando (Run):



# Aula 14 - Modificadores de acesso nas classes

**QuartoProjeto.java**

package com.sys4soft;

public class QuartoProjeto {

public static void main(String[] args) {

Aluno aluno\_1 = new Aluno();

aluno\_1.nome = "João";

aluno\_1.idade = 20;

aluno\_1.DefineSexo("masculino");

Aluno aluno\_2 = new Aluno();

aluno\_2.nome = "Carlos";

aluno\_2.idade = 25;

aluno\_2.DefineSexo("masculino");

// Apresentando os alunos

aluno\_1.ApresentaAluno();

aluno\_2.ApresentaAluno();

Matematica.Adicao(20, 30);

}

}

**Aluno.java**

package com.sys4soft;

public class Aluno {

public String nome;

public int idade;

private String sexo;

public void ApresentaAluno(){

System.out.println("O aluno chama-se " + nome + ", tem " + idade + " anos de idade e é do sexo " + sexo);

}

public void DefineSexo(String s){

// Define o valor de sexo do aluno

sexo = s;

}

}

**Matemática.java**

package com.sys4soft;

public class Matematica {

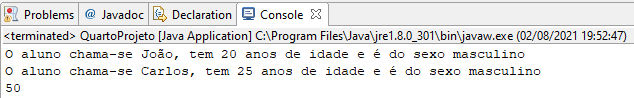
public static void Adicao(int parcela1, int parcela2){

System.out.println(parcela1 + parcela2);

}

}

Rodando (Run):



# Aula 15 - Conceito de Variável em Java - Parte 1

## Variáveis numéricas

byte 🡪 -128 a 127 (8 bits)

byte num1 = 120;

short 🡪 -32768 a 32767 (16 bits)

exemplo: short num2 = 32000;

int 🡪 -2,147,483,648 a 2,147,483,647 (32 bits)

exemplo: int num3 = -223682345;

long 🡪 - 9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 (64 bits)

exemplo: long num4 = 9223372036854775807L;

float 🡪 valores decimais de 16 bits

exemplo: float num5 = 134134131.33456876f;

double 🡪 valores decimais de 32 bits

exemplo: double num6 = 3692590114789379168.429321689322D;

# Aula 16 - Conceito de Variável em Java - Parte 2

## Variável booleana

boolean result = false; // false ou true

## Variáveis alfanuméricas

char caracter = 'c';

String palavra = "texto";

OBS.: String é uma classe e tem vários métodos associados.

# Aula 18 - Operadores aritméticos

**TestesOperadoresAritmeticos.java**

package com.sys4soft;

public class TestesOperadoresAritmeticos {

public static void main(String[] args) {

int valor1 = 10;

int valor2 = 3;

int soma, subtracao, multiplicacao, divisao, resto;

soma = valor1 + valor2;

subtracao = valor1 - valor2;

multiplicacao = valor1 \* valor2;

divisao = valor1 / valor2;

resto = valor1 % valor2;

System.out.println(valor1 + " + " + valor2 + " = " + soma);

System.out.println(valor1 + " - " + valor2 + " = " + subtracao);

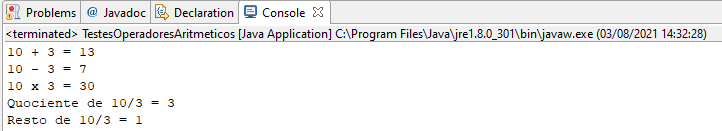
System.out.println(valor1 + " x " + valor2 + " = " + multiplicacao);

System.out.println("Quociente de " + valor1 + "/" + valor2 + " = " + divisao);

System.out.println("Resto de " + valor1 + "/" + valor2 + " = " + resto);

}

}



# Aula 19 - Precedencia e algebra aritmética em Java

**PrecedenciaAlgebraAritmetica.java**

package com.sys4soft;

public class PrecedenciaAlgebraAritmetica {

public static void main(String[] args) {

int calculo = 20 + 30 \* 10;

System.out.println(calculo);

calculo = (20 + 30) \* 10;

System.out.println(calculo);

calculo = 20 + 30 \* 10 + 20;

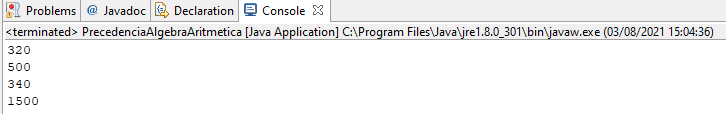
System.out.println(calculo);

calculo = (20 + 30) \* (10 + 20);

System.out.println(calculo);

}

}



# Aula 20 - Mecanismos de decisão - condição If

**CondicaoIf.java**

package com.sys4soft;

public class CondicaoIf {

public static void main(String[] args) {

int idade = 20;

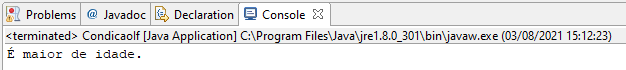
if(idade >= 18){

System.out.println("É maior de idade.");

}

}

}



# Aulas 22, 23 e 24 - Condição If Else e Else If

**CondicaoIfElse**

package com.sys4soft;

public class CondicaoIfElse {

public static void main(String[] args) {

String resposta = "";

int idade;

idade = 15;

resposta = situacao\_voto(idade);

System.out.println(resposta);

idade = 17;

resposta = situacao\_voto(idade);

System.out.println(resposta);

idade = 30;

resposta = situacao\_voto(idade);

System.out.println(resposta);

idade = 70;

resposta = situacao\_voto(idade);

System.out.println(resposta);

idade = 105;

resposta = situacao\_voto(idade);

System.out.println(resposta);

}

public static String situacao\_voto(int idade){

String resultado = "";

if(idade < 16){

resultado = idade + " anos: Não pode votar!";

} else if (idade >= 18 && idade < 70) {

resultado = idade + " anos: O voto é obrigatório!";

} else if ((idade >= 16 && idade < 18) || (idade >= 70 && idade <= 100)) {

resultado = idade + " anos: O voto é opcional!";

} else {

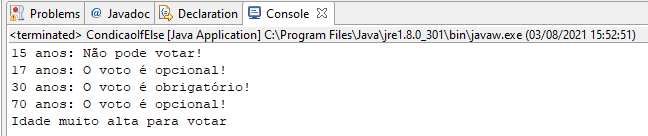
resultado = "Idade muito alta para votar";

}

return resultado;

}

}



# Aula 25 - Operador condicional / operador ternário

**TestesOperadorTernario.java**

package com.sys4soft;

public class TestesOperadorTernario {

public static void main(String[] args) {

String nome = "";

nome = "Rui";

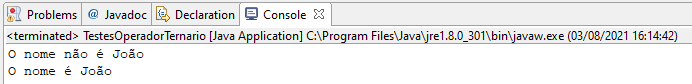
System.out.println(nome == "João" ? "O nome é João" : "O nome não é João");

nome = "João";

System.out.println(nome == "João" ? "O nome é João" : "O nome não é João");

}

}



# Aula 26 - Operadores de atribuição nas variáveis

**TestesOperadoresAtribuicao.java**

package com.sys4soft;

public class TestesOperadoresAtribuicao {

public static void main(String[] args) {

int valor = 10;

valor += 50; // 10 + 50 = 60

System.out.println(valor);

valor -= 30; // 60 - 30 = 30

System.out.println(valor);

valor \*= 3; // 30 x 3 = 90

System.out.println(valor);

valor /= 5; // 90 / 5 = 18

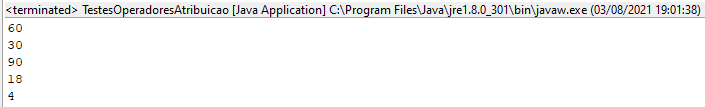
System.out.println(valor);

valor %= 7; // 18/7 = 2 + resto = 4

System.out.println(valor);

}

}



# Aula 27 - Noção de ciclo e ciclos While

**TestesWhile.java**

package com.sys4soft;

public class TestesWhile {

public static void main(String[] args) {

int valor = 1;

while(valor <= 10){

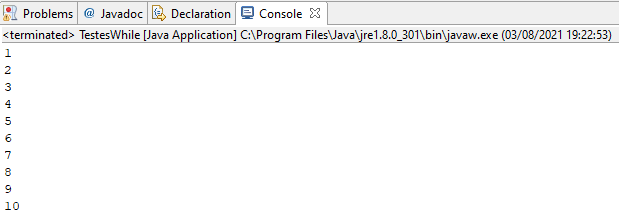
System.out.println(valor);

valor++;

}

}

}



# Aula 28 - Mais detalhes sobre o ciclo while

**InterrompendoCicloWhile.java**

package com.sys4soft;

public class InterrompendoCicloWhile {

public static void main(String[] args) {

int valor = 1;

int ultimo = 5;

while(valor <= 10){

System.out.println(valor);

if(valor == 5){

break;

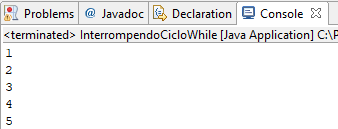
}

valor++;

}

}

}



# Aula 29 - Ciclo Do While e diferença com o ciclo While

**TestesCicloDoWhile.java**

package com.sys4soft;

public class TestesCicloDoWhile {

public static void main(String[] args) {

int valor = 1;

do {

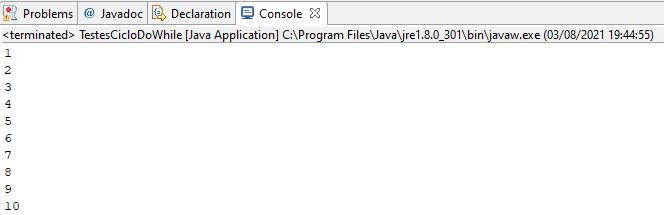
System.out.println(valor);

valor++;

} while(valor <= 10);

}

}



# Aula 30 - Ciclo For

**TestesCicloFor.java**

package com.sys4soft;

public class TestesCicloFor {

public static void main(String[] args) {

for(int i = 0; i <= 100; i+=10){

if(i == 40) continue;

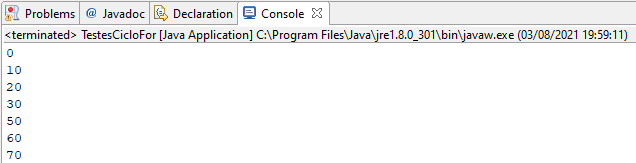
if(i == 80) break;

System.out.println(i);

}

}

}



# Aula 31 - Estrutura condicional Switch

**EstruturaCondicionalSwitch.java**

package com.sys4soft;

public class EstruturaCondicionalSwitch {

public static void main(String[] args) {

String nome = "Maria";

String sexo = "";

switch(nome){

case "João":

sexo = "masculino";

break;

case "Maria":

sexo = "feminino";

break;

case "Manuel":

sexo = "masculino";

break;

default:

sexo = "indefinido";

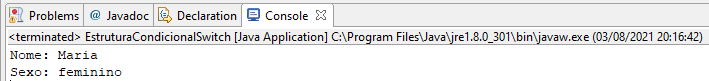
}

System.out.println("Nome: " + nome);

System.out.println("Sexo: " + sexo);

}

}



# Aula 32 - Utilização do operador lógico de negação

**TestesOperadorLogicoNegacao.java**

package com.sys4soft;

public class TestesOperadorLogicoNegacao {

public static void main(String[] args) {

boolean valor = false;

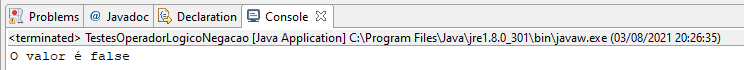
if(!valor){

System.out.println("O valor é false");

}

}

}



# Aula 34 - Principais métodos associados a Strings - Parte 1

**MetodosParte1.java**

package com.sys4soft;

public class MetodosParte1 {

public static void main(String[] args) {

String nome = "João";

String sobrenome = "Ribeiro";

String nome\_completo = nome + " " + sobrenome;

System.out.println(nome\_completo + " tem " + nome\_completo.length() + " caracteres");

System.out.println("=======================================================");

String texto = "Frase de testes.";

for(int index =0; index < texto.length(); index++){

char c = texto.charAt(index);

System.out.println(c);

}

System.out.println("=======================================================");

String teste\_substring1 = texto.substring(9);

String teste\_substring2 = texto.substring(6, 15);

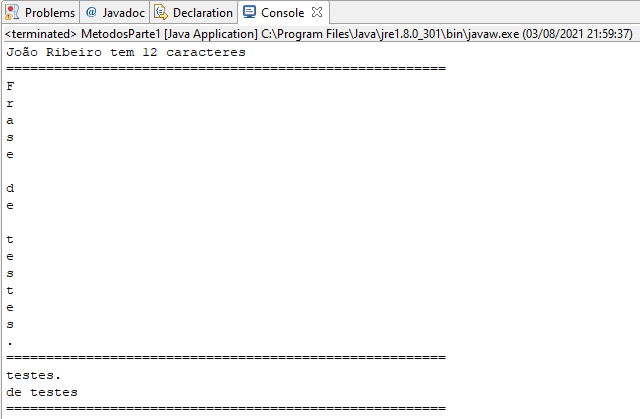
System.out.println(teste\_substring1);

System.out.println(teste\_substring2);

System.out.println("=======================================================");

}

}



# Aula 35 - Principais métodos associados a Strings - Parte 2

**MetodosParte2.java**

package com.sys4soft;

public class MetodosParte2 {

public static void main(String[] args) {

String nome1 = "João";

String nome2 = "João";

String nome3 = "";

if(nome1.equals(nome2)){

System.out.println("os nomes são iguais!");

}

System.out.println("========================================");

String sobrenome1 = "Ribeiro";

String nome\_completo = nome1.concat(" " + sobrenome1);

System.out.println(nome\_completo);

System.out.println("========================================");

int posicao1 = nome\_completo.indexOf('i');

int posicao2 = nome\_completo.lastIndexOf('i');

System.out.println(posicao1);

System.out.println(posicao2);

System.out.println("========================================");

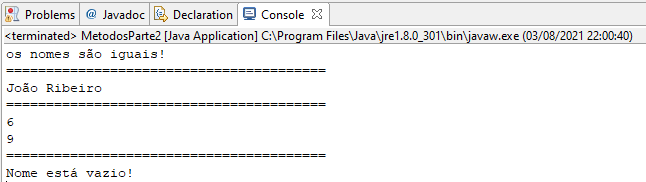
if(nome3.isEmpty()){

System.out.println("Nome está vazio!");

}

}

}



# Aula 36 - Principais métodos associados a Strings - Parte 3

**MetodosParte3.java**

package com.sys4soft;

public class MetodosParte3 {

public static void main(String[] args) {

String texto = "Frase de testes de strings no Java";

String texto2 = texto.toLowerCase();

String texto3 = texto.toUpperCase();

String subtexto1 = "testes";

String subtexto2 = "java";

String subtexto3 = "JAVA";

if(texto.contains(subtexto1)){

System.out.println(texto + " ==> CONTÉM a string " + subtexto1 );

} else {

System.out.println(texto + " ==> NÃO CONTÉM a string " + subtexto1 );

}

if(texto.contains(subtexto2)){

System.out.println(texto + " ==> CONTÉM a string " + subtexto2);

} else {

System.out.println(texto + " ==> NÃO CONTÉM a string " + subtexto2);

}

if(texto2.contains(subtexto2)){

System.out.println(texto2 + " ==> CONTÉM a string " + subtexto2);

} else {

System.out.println(texto2 + " ==> NÃO CONTÉM a string " + subtexto2);

}

if(texto3.contains(subtexto3)){

System.out.println(texto3 + " ==> CONTÉM a string " + subtexto3);

} else {

System.out.println(texto3 + " ==> NÃO CONTÉM a string " + subtexto3);

}

System.out.println("====================================================================");

String novo\_texto = texto.replace('J', 'j');

System.out.println(novo\_texto);

System.out.println("====================================================================");

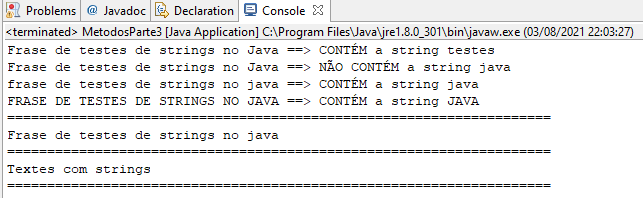
String texto4 = " Textes com strings ";

System.out.println(texto4.trim());

System.out.println("====================================================================");

}

}



# Aula 37 - Princípios básicos orientadores sobre OOP

OOP - Object Oriented Programming (Programação Orientada a Objetos)

Classe (Class):

* Contém propriedades (atributos);
* Contém métodos;

- Objeto é uma instância de classe

- Métodos podem retornar valores

**GeraAnimal.java**

package com.sys4soft;

public class GeraAnimal {

public static void main(String[] args) {

Animal gato\_1 = new Animal();

gato\_1.nome = "Fred";

gato\_1.especie = "gato";

gato\_1.cor = "branco";

gato\_1.sexo = "macho";

gato\_1.som = "mia";

Animal cachorro\_1 = new Animal();

cachorro\_1.nome = "Silloeh";

cachorro\_1.especie = "cachorro";

cachorro\_1.cor = "branca";

cachorro\_1.sexo = "femea";

cachorro\_1.som = "late";

System.out.println("Nome: " + gato\_1.nome + "\n"

+ "Espécie: " + gato\_1.especie + "\n"

+ "Cor: " + gato\_1.cor + "\n"

+ "Sexo: " + gato\_1.sexo + "\n");

System.out.println("Nome: " + cachorro\_1.nome + "\n"

+ "Espécie: " + cachorro\_1.especie + "\n"

+ "Cor: " + cachorro\_1.cor + "\n"

+ "Sexo: " + cachorro\_1.sexo + "\n");

}

}

**Animal.java**

package com.sys4soft;

public class Animal {

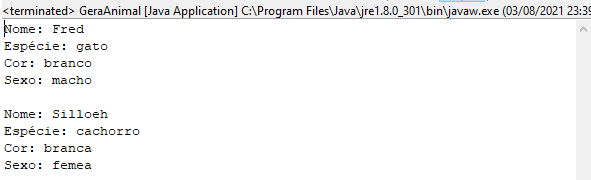
public String nome;

public String especie;

public String cor;

public String sexo;

}



# Aula 38 - Modificadores public e static

**GeraAnimal.java**

package com.sys4soft;

public class GeraAnimal {

public static void main(String[] args) {

Animal gato\_1 = new Animal();

gato\_1.nome = "Fred";

gato\_1.especie = "gato";

gato\_1.cor = "branco";

gato\_1.sexo = "macho";

gato\_1.som = "mia";

Animal cachorro\_1 = new Animal();

cachorro\_1.nome = "Silloeh";

cachorro\_1.especie = "cachorro";

cachorro\_1.cor = "branca";

cachorro\_1.sexo = "femea";

cachorro\_1.som = "late";

System.out.println("Nome: " + gato\_1.nome + "\n"

+ "Espécie: " + gato\_1.especie + "\n"

+ "Cor: " + gato\_1.cor + "\n"

+ "Sexo: " + gato\_1.sexo + "\n");

System.out.println("Nome: " + cachorro\_1.nome + "\n"

+ "Espécie: " + cachorro\_1.especie + "\n"

+ "Cor: " + cachorro\_1.cor + "\n"

+ "Sexo: " + cachorro\_1.sexo + "\n");

gato\_1.voz\_animal();

cachorro\_1.voz\_animal();

MethodStatic.saltar();

}

}

**Animal.java**

package com.sys4soft;

public class Animal {

public String nome;

public String especie;

public String sexo;

public String som;

public void voz\_animal(){

System.out.println("O " + especie + " " + som);

}

}

**MethodStatic.java**

package com.sys4soft;

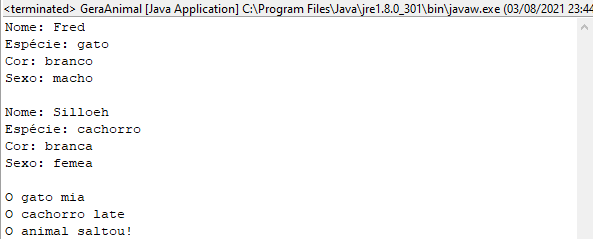
public class MethodStatic {

static void saltar(){

System.out.println("O animal saltou!");

}

}



# Aula 39 - Métodos com parâmetros

**GeraAnimal.java**

package com.sys4soft;

public class GeraAnimal {

public static void main(String[] args) {

Animal gato\_1 = new Animal();

gato\_1.nome = "Fred";

gato\_1.especie = "gato";

gato\_1.cor = "branco";

gato\_1.sexo = "macho";

gato\_1.definir\_raca("SRD");

gato\_1.som = "mia";

Animal cachorro\_1 = new Animal();

cachorro\_1.nome = "Silloeh";

cachorro\_1.especie = "cachorro";

cachorro\_1.cor = "branca";

cachorro\_1.sexo = "femea";

cachorro\_1.definir\_raca("Pug");

cachorro\_1.som = "late";

System.out.println("Nome: " + gato\_1.nome + "\n"

+ "Espécie: " + gato\_1.especie + "\n"

+ "Raça: " + gato\_1.raca + "\n"

+ "Cor: " + gato\_1.cor + "\n"

+ "Sexo: " + gato\_1.sexo + "\n");

System.out.println("Nome: " + cachorro\_1.nome + "\n"

+ "Espécie: " + cachorro\_1.especie + "\n"

+ "Raça: " + cachorro\_1.raca + "\n"

+ "Cor: " + cachorro\_1.cor + "\n"

+ "Sexo: " + cachorro\_1.sexo + "\n");

gato\_1.voz\_animal();

cachorro\_1.voz\_animal();

MethodStatic.saltar();

}

}

**Animal.java**

package com.sys4soft;

public class Animal {

public String nome;

public String especie;

public String cor;

public String sexo;

public String som;

public String raca;

public void voz\_animal(){

System.out.println("O " + especie + " " + som);

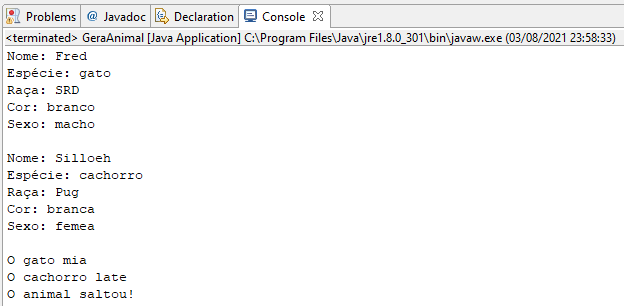
}

public void definir\_raca(String raca\_animal){

raca = raca\_animal;

}

}



# Aula 40 - Métodos que retornam valores

**GeraAnimal.java**

package com.sys4soft;

public class GeraAnimal {

public static void main(String[] args) {

Animal gato\_1 = new Animal();

gato\_1.nome = "Fred";

gato\_1.especie = "gato";

gato\_1.cor = "branco";

gato\_1.sexo = "macho";

gato\_1.definir\_raca("SRD");

gato\_1.som = "mia";

Animal cachorro\_1 = new Animal();

cachorro\_1.nome = "Silloeh";

cachorro\_1.especie = "cachorro";

cachorro\_1.cor = "branca";

cachorro\_1.sexo = "femea";

cachorro\_1.definir\_raca("Pug");

cachorro\_1.som = "late";

String resultado1 = gato\_1.exibir\_dados\_animal();

String resultado2 = cachorro\_1.exibir\_dados\_animal();

System.out.println(resultado1);

System.out.println(resultado2);

gato\_1.voz\_animal();

cachorro\_1.voz\_animal();

MethodStatic.saltar();

}

}

**Animal.java**

package com.sys4soft;

public class Animal {

public String nome;

public String especie;

public String cor;

public String sexo;

public String som;

public String raca;

public void voz\_animal(){

System.out.println("O " + especie + " " + som);

}

public void definir\_raca(String raca\_animal){

raca = raca\_animal;

}

public String exibir\_dados\_animal(){

String dados = "";

dados = "Nome: " + nome + "\n"

+ "Espécie: " + especie + "\n"

+ "Raça: " + raca + "\n"

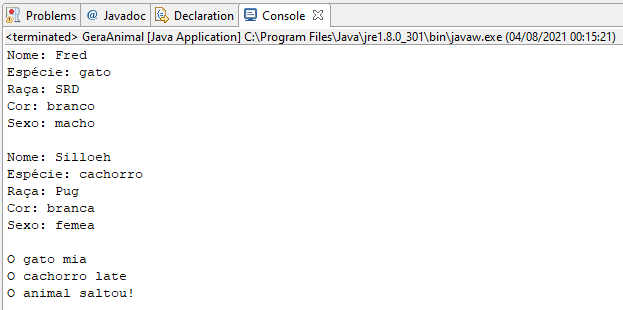
+ "Cor: " + cor + "\n"

+ "Sexo: " + sexo + "\n";

return dados;

}

}



# Aula 41 - Métodos com vários parâmetros e overloading

**GeraCalculo.java**

package com.sys4soft;

public class GeraCalculo {

public static void main(String[] args) {

Calculo1 operacao\_1 = new Calculo1();

Calculo1 operacao\_2 = new Calculo1();

Calculo1 operacao\_3 = new Calculo1();

Calculo1 operacao\_4 = new Calculo1();

int resultado1 = operacao\_1.Somar(16, 73);

int resultado2 = operacao\_2.Subtrair(87, 78);

int resultado3 = operacao\_3.Dividir(120, 5);

int resultado4 = operacao\_4.Multiplicar(4, 8);

System.out.println(resultado1);

System.out.println(resultado2);

System.out.println(resultado3);

System.out.println(resultado4);

System.out.println("================================");

Calculo2 calc = new Calculo2();

int result1 = calc.Operacao(45, 30, "somar");

int result2 = calc.Operacao(45, 30, "subtrair");

int result3 = calc.Operacao(4, 30, "multiplicar");

int result4 = calc.Operacao(48, 4, "dividir");

System.out.println(result1);

System.out.println(result2);

System.out.println(result3);

System.out.println(result4);

System.out.println("================================");

Overloading calculo = new Overloading();

calculo.soma(14, 85);

calculo.soma(25, 17, 63);

}

}

**Calculo1.java**

package com.sys4soft;

public class Calculo1 {

public int Somar(int num1, int num2){

return num1 + num2;

}

public int Subtrair(int num1, int num2){

return num1 - num2;

}

public int Multiplicar(int num1, int num2){

return num1 \* num2;

}

public int Dividir(int num1, int num2){

return num1/num2;

}

}

**Calculo2.java**

package com.sys4soft;

public class Calculo2 {

public int Operacao(int num1, int num2, String operacao){

int resultado = 0;

switch(operacao){

case "somar":

resultado = num1 + num2;

break;

case "subtrair":

resultado = num1 - num2;

break;

case "multiplicar":

resultado = num1 \* num2;

break;

case "dividir":

resultado = num1 / num2;

break;

}

return resultado;

}

}

**Overloading.java**

package com.sys4soft;

public class Overloading {

public void soma(int num1, int num2){

System.out.println(num1 + num2);

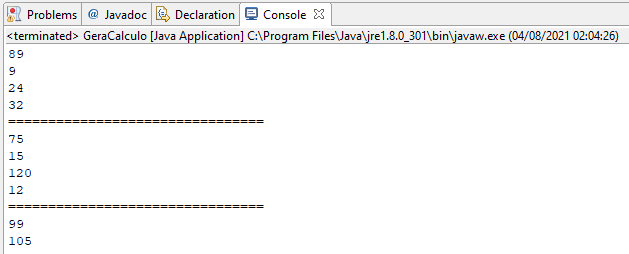
}

public void soma(int num1, int num2, int num3){

System.out.println(num1 + num2 + num3);

}

}



# Aula 42 - Mais sobre overloading e parâmetros

**GeraCalculo.java**

package com.sys4soft;

public class GeraCalculo {

public static void main(String[] args) {

Calculo1 operacao\_1 = new Calculo1();

Calculo1 operacao\_2 = new Calculo1();

Calculo1 operacao\_3 = new Calculo1();

Calculo1 operacao\_4 = new Calculo1();

int resultado1 = operacao\_1.Somar(16, 73);

int resultado2 = operacao\_2.Subtrair(87, 78);

int resultado3 = operacao\_3.Dividir(120, 5);

int resultado4 = operacao\_4.Multiplicar(4, 8);

System.out.println(resultado1);

System.out.println(resultado2);

System.out.println(resultado3);

System.out.println(resultado4);

System.out.println("================================");

Calculo2 calc = new Calculo2();

int result1 = calc.Operacao(45, 30, "somar");

int result2 = calc.Operacao(45, 30, "subtrair");

int result3 = calc.Operacao(4, 30, "multiplicar");

int result4 = calc.Operacao(48, 4, "dividir");

System.out.println(result1);

System.out.println(result2);

System.out.println(result3);

System.out.println(result4);

System.out.println("================================");

Overloading calculo = new Overloading();

calculo.soma(14, 85);

calculo.soma(25, 17, 63);

System.out.println("================================");

PassagemParametros param = new PassagemParametros();

param.nota(5, "Joaquim");

param.nota(8, "João");

param.nota(10, "Maria");

}

}

**PassagemParametros.java**

package com.sys4soft;

public class PassagemParametros {

int num1 = 0;

String nome;

public void nota(int num1, String nome){

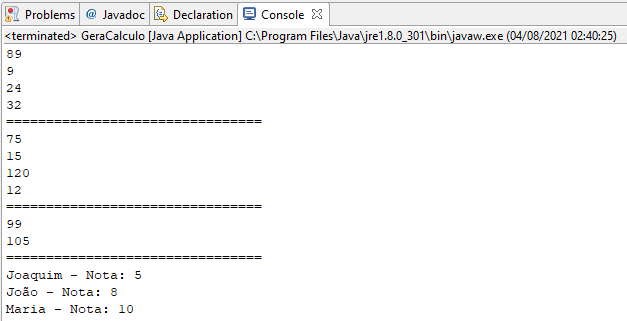
this.num1 = num1;

this.nome = nome;

System.out.println(nome + " - Nota: " + num1);

}

}



# Aula 43 - Construtor de uma classe

**GeraSocio.java**

package com.sys4soft;

public class GeraSocio {

public static void main(String[] args) {

Socio s1 = new Socio("João");

s1.ApresentarSocio();

Socio s2 = new Socio("Paula");

s2.ApresentarSocio();

Socio s3 = new Socio("Carlos");

s3.ApresentarSocio();

}

}

**Socio.java**

package com.sys4soft;

public class Socio {

String nome;

int idade;

String sexo;

boolean vip;

// construtor da classe

public Socio(String nome){

// define o nome do sócio

this.nome = nome;

// System.out.println("Novo sócio criado!");

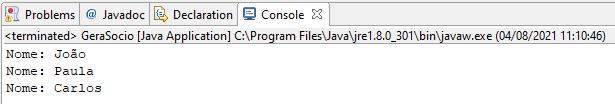
}

public void ApresentarSocio(){

System.out.println("Nome: " + nome);

}

}



# Aula 44 - Recapitulando conhecimentos sobre classes

- As variáveis estáticas não necessitam ser instanciadas para serem usadas.

## Atributos e Métodos de instanciação e estáticos

**GeraPessoas.java**

package com.sys4soft;

public class GeraPessoas {

public static void main(String[] args) {

Pessoas p1 = new Pessoas();

p1.nome = "João";

Pessoas.idade = 25;

p1.ApresentarNome(p1.nome);

Pessoas.ApresentarIdade(Pessoas.idade);

}

}

**Pessoas.java**

package com.sys4soft;

public class Pessoas {

public String nome;

public static int idade;

public void ApresentarNome(String nome){

this.nome = nome;

System.out.println("Nome: " + nome);

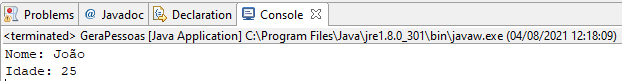
}

public static void ApresentarIdade(int idade){

System.out.println("Idade: " + idade);

}

}



# Aula 45 - Utilizando o termo Final para definir constantes

**GeraPessoas.java**

package com.sys4soft;

public class GeraPessoas {

public static void main(String[] args) {

Pessoas p1 = new Pessoas();

p1.nome = "João";

Pessoas.idade = 25;

p1.ApresentarNome(p1.nome);

Pessoas.ApresentarIdade(Pessoas.idade);

Pessoas.ApresentarIdade(Pessoas.idade2);

}

}

**Pessoas.java**

package com.sys4soft;

public class Pessoas {

public String nome;

public static int idade = 38;

public static final int idade2 = 54;

public void ApresentarNome(String nome){

this.nome = nome;

System.out.println("Nome: " + nome);

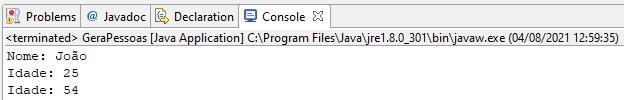
}

public static void ApresentarIdade(int idade){

System.out.println("Idade: " + idade);

}

}



# Aula 46 - Criando multiplos construtores para uma classe

**GeraSocio.java**

package com.sys4soft;

public class GeraSocio {

public static void main(String[] args) {

Socio s1 = new Socio("João");

s1.ApresentarSocio();

Socio s2 = new Socio("Paula");

s2.ApresentarSocio();

Socio s3 = new Socio("Carlos");

s3.ApresentarSocio();

System.out.println("========================================");

Socio1 s4 = new Socio1("Carlos");

System.out.println("");

Socio1 s5 = new Socio1();

}

}

**Socio1.java**

package com.sys4soft;

public class Socio1 {

String nome;

int idade;

String sexo;

boolean vip;

// Construtor da classe sem parâmetros

public Socio1() {

this("Roberto");

System.out.println("Construtor sem parâmetros");

}

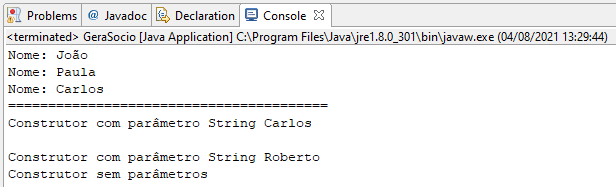
// construtor da classe com parâmetros

public Socio1(String nome){

System.out.println("Construtor com parâmetro String " + nome);

}

}



# Aula 47 - Hereditariedade e subclasses

**GeraHumanos.java**

package com.sys4soft;

public class GeraHumanos {

public static void main(String[] args) {

Homem h1 = new Homem();

h1.nome = "Ricardo";

h1.idade = 19;

h1.tem\_barba = false;

Mulher m1 = new Mulher();

m1.nome = "Solange";

m1.idade = 32;

m1.usa\_battom = true;

h1.ApresentarHumano();

h1.info\_barba(h1.tem\_barba);

m1.ApresentarHumano();

m1.info\_battom(m1.usa\_battom);

}

}

**Humano.java**

package com.sys4soft;

public class Humano {

public String nome;

public int idade;

public void ApresentarHumano(){

System.out.println("Nome: " + nome + "\n" + "Idade: " + idade);

}

}

**Homem.java**

package com.sys4soft;

public class Homem extends Humano {

boolean tem\_barba;

public void info\_barba(boolean tem\_barba){

String resposta;

if(tem\_barba){

resposta = "sim";

} else {

resposta = "não";

}

System.out.println("Tem barba: " + resposta + "\n");

}

}

**Mulher.java**

package com.sys4soft;

public class Mulher extends Humano {

boolean usa\_battom;

public void info\_battom(boolean usa\_battom){

String resposta;

if(usa\_battom){

resposta = "sim";

} else {

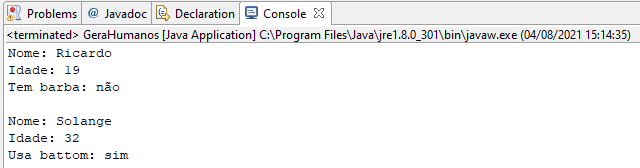
resposta = "não";

}

System.out.println("Usa battom: " + resposta + "\n");

}

}



# Aula 48 - Classes que não podem ser super classes

- Para que uma classe não possa ser usada como uma super classe, ou seja, não possa ser herdada, basta acrescentar o termo final.

Exemplo:

**Humano.java**

package com.sys4soft;

public final class Humano {

public String nome;

public int idade;

public void ApresentarHumano(){

System.out.println("Nome: " + nome + "\n" + "Idade: " + idade);

}

}

# Aula 49 - Hierarquia de construtores

**Humano.java**

package com.sys4soft;

public class Humano {

public String nome;

public int idade;

public Humano(){

System.out.println("Humano");

}

}

**Homem.java**

package com.sys4soft;

public class Homem extends Humano {

boolean tem\_barba;

public Homem(){

System.out.println("Homem");

}

}

**Mulher.java**

package com.sys4soft;

public class Mulher extends Humano {

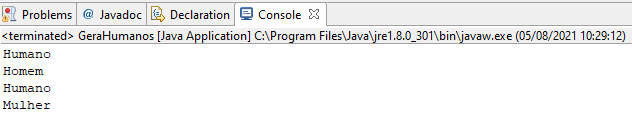
boolean usa\_battom;

public Mulher(){

System.out.println("Mulher");

}

}



# Aula 50 - Introdução aos Arrays

**Introducao.java**

package com.sys4soft;

public class Introducao {

public static void main(String[] args) {

String aluno1 = "João";

String aluno2 = "Rui";

String aluno3 = "Francisco";

String aluno4 = "Paula";

String aluno5 = "Luisa";

String aluno6 = "Ana";

// array

String[] alunos = new String[6];

alunos[0] = "João";

alunos[1] = "Rui";

alunos[2] = "Francisco";

alunos[3] = "Paula";

alunos[4] = "Luisa";

alunos[5] = "Ana";

System.out.println("O aluno escolhido é o aluno " + alunos[2]);

}

}



# Aula 51 - Iniciação de Arrays

**IniciacaoArrays.java**

package com.sys4soft;

public class IniciacaoArrays {

public static void main(String[] args) {

// array

String[] alunos = {

"João",

"Rui",

"Francisco",

"Raquel",

"Ana",

"Clara",

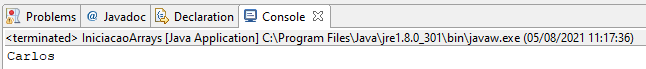
"Carlos"

};

System.out.println(alunos[6]);

}

}



# Aula 52 - Apresentando dados de Arrays

**ApresentandoDados.java**

package com.sys4soft;

public class ApresentandoDados {

public static void main(String[] args) {

// array

String[] alunos = {

"João",

"Rui",

"Francisco",

"Raquel",

"Ana",

"Clara",

"Carlos"

};

for(int i = 0; i < alunos.length; i++){

System.out.println(alunos[i]);

}

System.out.println("=============================");

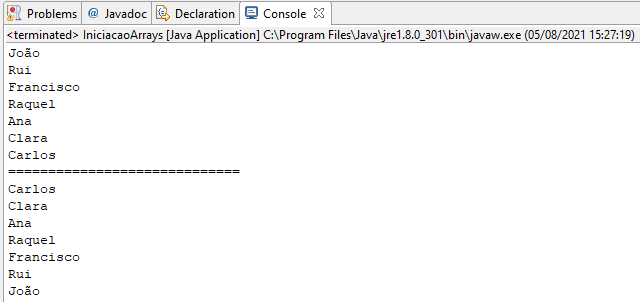
for(int i = alunos.length-1; i >= 0; i--){

System.out.println(alunos[i]);

}

}

}



# Aula 53 - Outras formas de apresentação

## Copiando parte de um array para outro array

**ArrayCopy.java**

package com.sys4soft;

public class ArrayCopy {

public static void main(String[] args) {

// array

String[] alunos = {

"João",

"Rui",

"Francisco",

"Raquel",

"Ana",

"Clara",

"Carlos"

};

String[] novo = new String[4];

System.arraycopy(alunos, 2, novo, 0, 4);

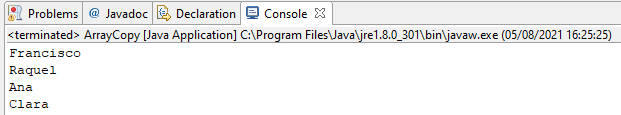
for(String item:novo){

System.out.println(item);

}

}

}



# Aula 54 - Ordenando os dados de um array de Strings

**OrdenandoDados.java**

package com.sys4soft;

import java.util.Arrays;

import java.util.Collections;

public class OrdenandoDados {

public static void main(String[] args) {

// array

String[] alunos = {

"João",

"Rui",

"Francisco",

"Raquel",

"Ana",

"Clara",

"Carlos"

};

Arrays.sort(alunos);

for(String nome:alunos){

System.out.println(nome);

};

System.out.println("==============================");

// array numerico

int[] numeros = {

45, 23, 67, 87, 42, 68

};

Arrays.sort(numeros);

for(int valor:numeros){

System.out.println(valor);

};

System.out.println("==============================");

Arrays.sort(alunos, Collections.reverseOrder());

for(String nome:alunos){

System.out.println(nome);

};

}

}



# Aula 55 - Método básico de reversão de um Array numérico

**ReversaoArrayNumerico.java**

package com.sys4soft;

public class ReversaoArrayNumerico {

public static void main(String[] args) {

// array numerico

int[] numeros = {

1, 3, 5, 7, 9, 11, 13

};

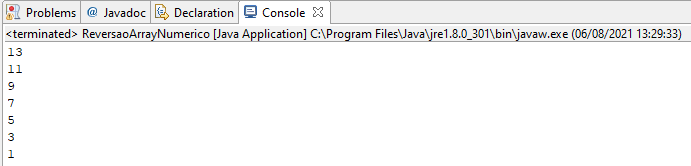
for(int i = numeros.length - 1; i >= 0; i--){

System.out.println(numeros[i]);

}

}

}



# Aula 56 - Usar um método para tornar Array numérico invertido

**OrdenarArrayInt.java**

package com.sys4soft;

public class OrdenarArrayInt {

public static void main(String[] args) {

// array numerico

int[] numeros = {

2, 5, 4, 9, 7, 3, 8, 1, 6

};

numeros = MetodoOrdenarArrayInt.OrdemAsc(numeros);

for(int valor:numeros){

System.out.println(valor);

}

System.out.println("=====");

numeros = MetodoOrdenarArrayInt.OrdemDesc(numeros);

for(int valor:numeros){

System.out.println(valor);

}

}

}

**MetodoOrdenarArrayInt.java**

package com.sys4soft;

import java.util.Arrays;

public class MetodoOrdenarArrayInt {

public static int[] OrdemAsc(int[] array\_int){

int[] array\_int\_asc = new int[array\_int.length];

Arrays.sort(array\_int);

return array\_int;

}

public static int[] OrdemDesc(int[] array\_int){

Arrays.sort(array\_int);

int[] array\_int\_desc = new int[array\_int.length];

int j = 0;

for(int i = array\_int.length - 1; i >= 0; i--){

array\_int\_desc[j] = array\_int[i];

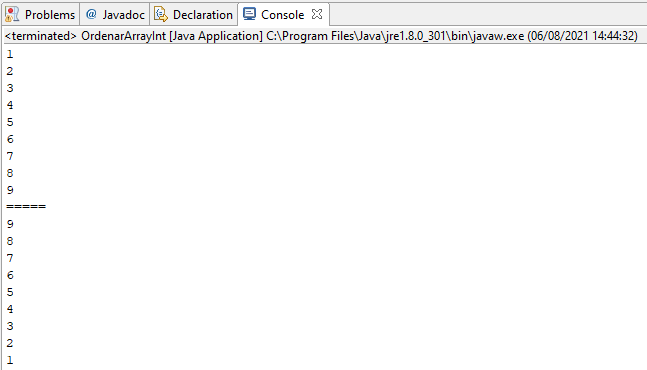
j++;

}

return array\_int\_desc;

}

}



# Aula 57 - Exemplo de um Array multidimensional

**ArrayMultidimensional.java**

package com.sys4soft;

public class ArrayMultidimensional {

public static void main(String[] args) {

// array multidimensional

int[][] numeros = {

{1,2,3,4,5,6,7},

{7,6,5,4,3,2,1}

};

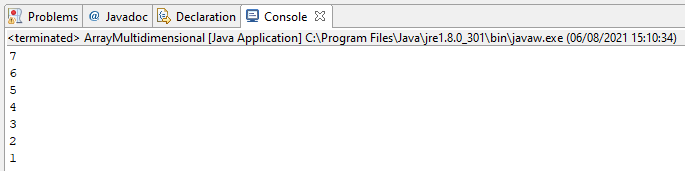
for(int i = 0; i < numeros[1].length; i++){

System.out.println(numeros[1][i]);

}

}

}



# Aula 58 - Tipos primitivos e não primitivos

**TiposPrimitivos.java**

package com.sys4soft;

public class TiposPrimitivos {

public static void main(String[] args) {

byte valor1 = 0;

short valor2 = 0;

int valor3 = 0;

long valor4 = 0;

float valor5 = 0;

double valor6 = 0;

boolean valor7 = false;

char valor8 = 'a';

Aluno aluno1 = new Aluno();

aluno1.nome = "João";

aluno1.idade = 25;

}

}

**Aluno.java**

package com.sys4soft;

public class Aluno {

String nome;

int idade;

}

# Aula 59 - Exemplo de utilização de tipos não primitivos

**GeraTuma.java**

package com.sys4soft;

public class GeraTuma {

public static void main(String[] args) {

// criar a turma

Aluno[] AlunosDaTurma = new Aluno[5];

AlunosDaTurma[0] = new Aluno() {{

nome = "João";

idade = 17;

}};

AlunosDaTurma[1] = new Aluno() {{

nome = "Rui";

idade = 15;

}};

AlunosDaTurma[2] = new Aluno() {{

nome = "Carlos";

idade = 18;

}};

AlunosDaTurma[3] = new Aluno() {{

nome = "Ernesto";

idade = 14;

}};

AlunosDaTurma[4] = new Aluno() {{

nome = "Ricardo";

idade = 19;

}};

for(Aluno a: AlunosDaTurma){

System.out.println("O aluno " + a.nome + " tem " + a.idade + " anos de idade.");

};

}

}

**Aluno.java**

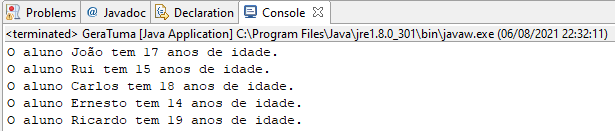
package com.sys4soft;

public class Aluno {

String nome;

int idade;

}



# Aula 60 - Exceções - Tratamento de erros no Java

Um erro é algo inesperado, ou seja, acontece durante a execução de nosso software.

Uma exceção é um erro que o programador está a espera de que possa eventualmente acontecer.

Existe a possibilidade de introduzir uma estrutura de código que nos vai permitir executar um bloco de código se o erro ocorrer, ou outro bloco de código se o erro não ocorrer.

**Erros.java**

package com.sys4soft;

public class Erros {

public static void main(String[] args) {

int[] valores = new int[5];

try {

System.out.println(valores[5]);

} catch(ArrayIndexOutOfBoundsException e) {

System.out.println("Não existe este aluno!");

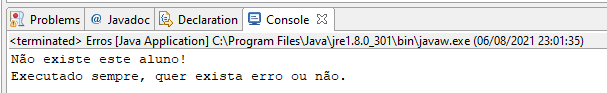
} finally {

System.out.println("Executado sempre, quer exista erro ou não.");

}

}

}



# Aula 61 - Tratamento especial de Strings no Java

## Concatenando strings com StringBuilder

**ConcatenandoStrings.java**

package com.sys4soft;

public class ConcatenandoStrings {

public static void main(String[] args) {

char[] caracteres = {'J','o','ã','o'};

String nome = new String(caracteres);

System.out.println(nome);

System.out.println(nome + " tem " + nome.length() + " letras.");

StringBuilder str = new StringBuilder();

str.append("João");

str.append(" tem ");

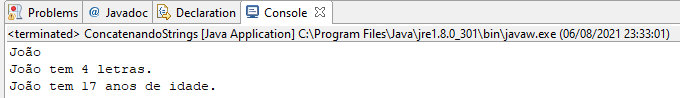
str.append(17);

str.append(" anos de idade.");

System.out.println(str);

}

}



# Aula 62 - Breve introdução à Data e Hora em Java

**IntroducaoDataHora.java**

package com.sys4soft;

import java.time.\*;

public class IntroducaoDataHora {

public static void main(String[] args) {

System.out.println(LocalDate.now().toString());

System.out.println(LocalDate.now().getYear());

System.out.println(LocalDate.now().getDayOfYear());

System.out.println(LocalDate.now().getMonthValue());

System.out.println(LocalTime.now().toString());

System.out.println(LocalTime.now().getHour());

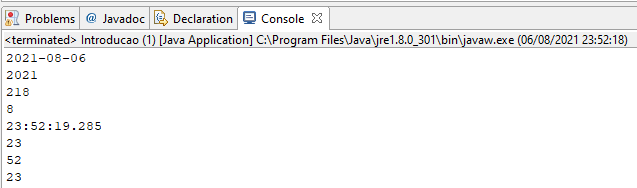
System.out.println(LocalTime.now().getMinute());

int hora = LocalTime.now().getHour();

System.out.println(hora);

}

}



# Aula 63 - Criação de arquivos e pastas no Java - Parte1

## Criando um arquivo de texto vazio

**CriandoArquivo.java**

package com.sys4soft;

import java.io.File;

public class CriandoArquivo {

public static void main(String[] args) {

String nomeArquivo = "C:\\lixo\\teste.txt";

File arquivo = new File(nomeArquivo);

try{

arquivo.createNewFile();

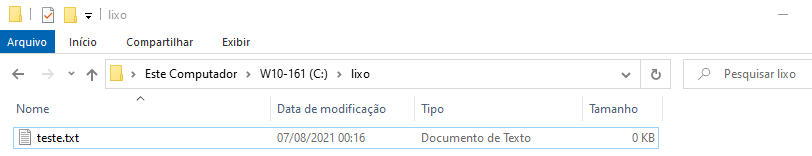
} catch(Exception e) {

System.out.println("Erro! " + arquivo.getPath());

}

}

}



# Aula 64 - Criação de arquivos e pastas no Java - Parte2

## Criando um arquivo de texto vazio

**CriandoArquivo.java**

package com.sys4soft;

import java.io.File;

public class CriandoArquivo {

public static void main(String[] args) {

// String nomeArquivo = "C:\\lixo\\teste.txt";

String nomeArquivo = "C:" + File.separator + "lixo" + File.separator + "teste.txt";

File arquivo = new File(nomeArquivo);

try{

arquivo.createNewFile();

System.out.println("Arquivo " + arquivo.getPath() + " criado com sucesso!");

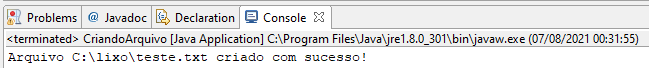
} catch(Exception e) {

System.out.println("Erro! " + arquivo.getPath());

}

}

}



## Verificando se um arquivo existe

**VerificandoSeArquivoExiste.java**

package com.sys4soft;

import java.io.File;

public class VerificandoSeArquivoExiste {

public static void main(String[] args) {

String nomeArquivo = "C:" + File.separator + "lixo" + File.separator + "teste.txt";

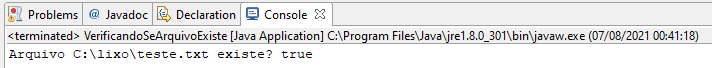
File arquivo = new File(nomeArquivo);

// determinar a existência de um arquivo específico

System.out.println("Arquivo " + arquivo + " existe? " + arquivo.exists());

}

}



# Aula 65 - Criação de arquivos e pastas no Java - Parte3

## Apagando um arquivo

**ApagandoArquivo.java**

package com.sys4soft;

import java.io.File;

public class ApagandoArquivo {

public static void main(String[] args) {

String nomeArquivo = "C:" + File.separator + "lixo" + File.separator + "teste.txt";

File arquivo = new File(nomeArquivo);

boolean existe = arquivo.exists();

if(existe){

arquivo.delete();

System.out.println("Arquivo deletado com sucesso!");

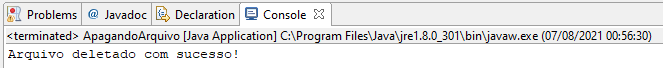
} else {

System.out.println("O arquivo não existe!");

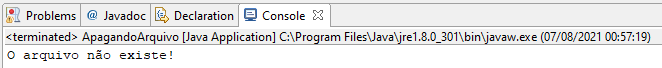
}

}

}



- Rodando novamente:



# Aula 66 - Criação de arquivos e pastas no Java - Parte4

## Criando uma pasta

**CriandoPasta.java**

package com.sys4soft;

import java.io.File;

public class CriandoPasta {

public static void main(String[] args) {

String nomePasta = "C:" + File.separator + "lixo" + File.separator + "teste1";

File pasta = new File(nomePasta);

try {

pasta.mkdir();

System.out.println("Pasta criada com sucesso!");

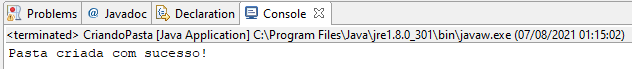
} catch(Exception e) {

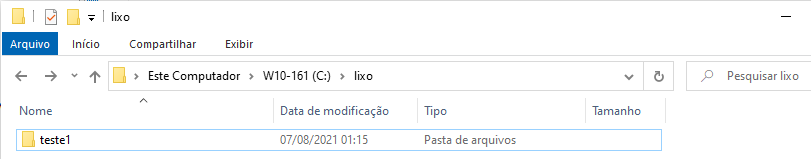
System.out.println("Erro ao criar a pasta!");

}

}

}





## Criando Pastas

**CriandoPastas.java**

package com.sys4soft;

import java.io.File;

public class CriandoPastas {

public static void main(String[] args) {

String nomePasta = "C:" + File.separator + "lixo" + File.separator + "teste2" + File.separator + "teste3";

File pasta = new File(nomePasta);

try{

pasta.mkdirs();

System.out.println("Pastas criadas com sucesso!");

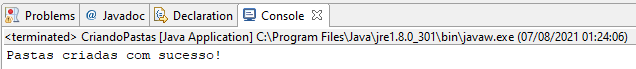
} catch(Exception e) {

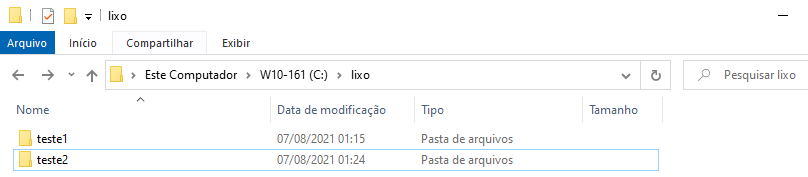
System.out.println("Erro ao criar as pastas!");

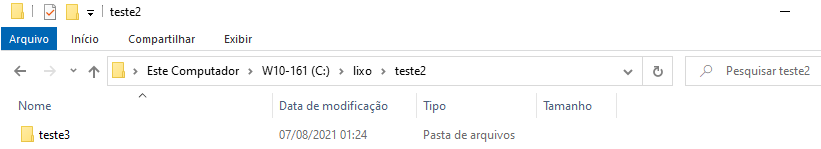
}

}

}







# Aula 67 - Criação de arquivos e pastas no Java - Parte5

**ApagandoPasta.java**

package com.sys4soft;

import java.io.File;

public class ApagandoPasta {

public static void main(String[] args) {

String nomePasta = "C:" + File.separator + "lixo"

+ File.separator + "teste2" + File.separator + "teste3";

File pasta = new File(nomePasta);

boolean existe = pasta.exists();

if(existe){

pasta.delete();

System.out.println("Pasta deletada com sucesso!");

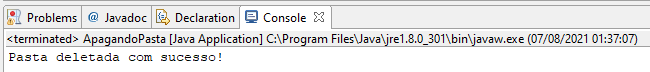
} else {

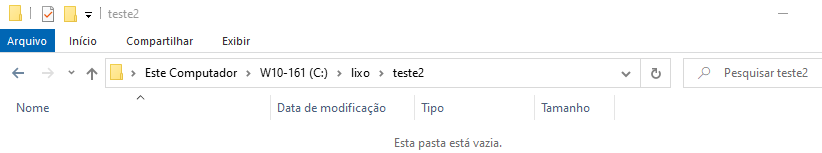
System.out.println("Pasta não encontrada!");

}

}

}





# Aula 68 - Ler os conteúdos de arquivos de texto

**C:\lixo\teste.txt**

Primeira frase.

Segunda frase.

Terceira frase.

**LerArquivo.java**

package com.sys4soft;

import java.io.BufferedReader;

import java.io.FileReader;

public class LerArquivo {

public static void main(String[] args) {

try {

BufferedReader bf = new BufferedReader(new FileReader("C:\\lixo\\teste.txt"));

StringBuilder str = new StringBuilder();

String linha = bf.readLine();

while(linha != null){

str.append(linha);

str.append(System.lineSeparator());

linha = bf.readLine();

}

String texto\_final = str.toString();

System.out.println(texto\_final);

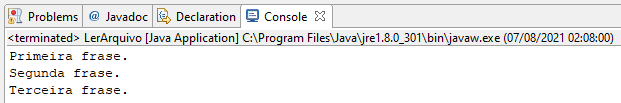
} catch(Exception e) {

System.out.println("Erro!");

}

}

}



# Aula 69 - Escrever conteúdos num arquivos de texto

**InserirTexto.java**

package com.sys4soft;

import java.io.BufferedWriter;

import java.io.FileWriter;

public class InserirTexto {

public static void main(String[] args) {

try {

BufferedWriter escrever = new BufferedWriter(new FileWriter("C:\\lixo\\teste.txt", true));

escrever.write("Primeira frase.");

escrever.newLine();

escrever.write("Segunda frase.");

escrever.newLine();

escrever.write("Terceira frase.");

escrever.newLine();

escrever.write("Quarta frase.");

escrever.newLine();

escrever.close();

System.out.println("Escrita realizada com sucesso!");

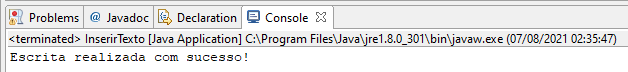
} catch(Exception e) {

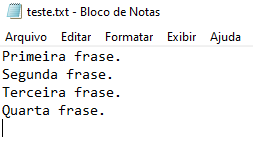
System.out.println("Erro! " + e);

}

}

}





# Aula 70 - Introdução a GUI com classes Swing

**ProjetoSwing1.java**

import javax.swing.JFrame;

public class ProjetoSwing1 {

public static void main(String[] args) {

JFrame form1 = new JFrame("Form 1");

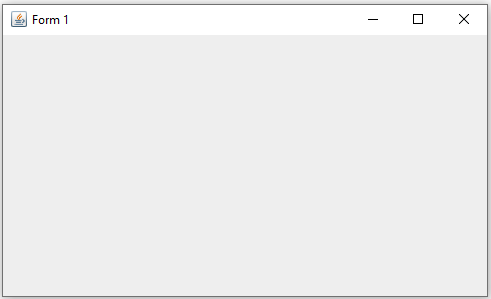
form1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

form1.setSize(500, 300);

form1.setVisible(true);

}

}



# Aula 71 - Java, GUI e Swing

**SwingJOptionPane.java**

package com.sys4soft;

import javax.swing.JOptionPane;

public class SwingJOptionPane {

public static void main(String[] args) {

// solicitar a primeira parcela

String primeira\_parcela = JOptionPane.showInputDialog("Insira a primeira parcela");

// solicitar a segunda parcela

String segunda\_parcela = JOptionPane.showInputDialog("Insira a segunda parcela");

// converter as variáveis string em int

int parcela1 = Integer.parseInt(primeira\_parcela);

int parcela2 = Integer.parseInt(segunda\_parcela);

int resultado = parcela1 + parcela2;

JOptionPane.showMessageDialog(null,

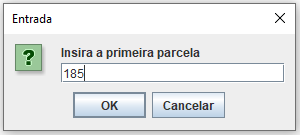
"O resultado é " + resultado, "Resultado",

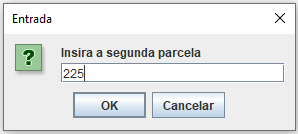
JOptionPane.INFORMATION\_MESSAGE

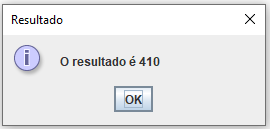
);

}

}







# Aula 72 - Adicionando um componente numa JFrame

## Adicionando um label numa JFrame

**SwingJLabel.java**

package com.sys4soft;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class SwingJLabel {

public static void main(String[] args) {

// criar a janela JFrame

JFrame janela = new JFrame();

janela.setTitle("Título da janela");

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

janela.setSize(500, 300);

// criar o JLabel

JLabel label1 = new JLabel("Este texto pertence a um label.");

label1.setHorizontalAlignment(JLabel.CENTER);

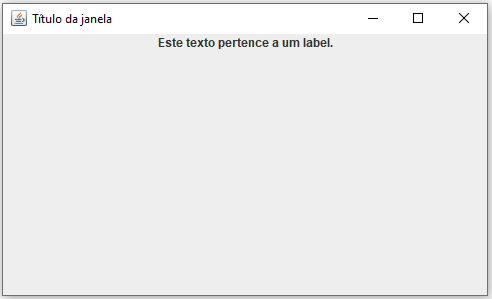
label1.setVerticalAlignment(JLabel.TOP);

janela.add(label1);

janela.setVisible(true);

}

}



# Aula 73 - Exemplo do conceito de layout manager

**SwingLayoutManager.java**

package com.sys4soft;

import java.awt.FlowLayout;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class SwingLayoutManager {

public static void main(String[] args) {

// criar a janela JFrame

JFrame janela = new JFrame("Janela de teste");

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

janela.setSize(500, 300);

// label1

JLabel label1 = new JLabel("Texto do label1.");

// label2

JLabel label2 = new JLabel("Texto do label2.");

// label3

JLabel label3 = new JLabel("Texto do label3.");

janela.setLayout(new FlowLayout());

janela.add(label1);

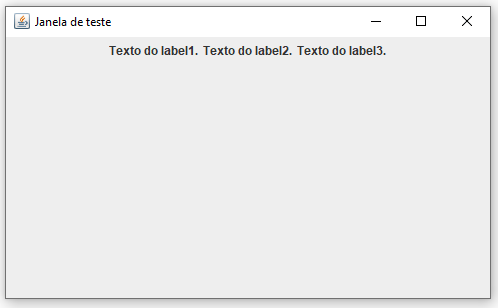
janela.add(label2);

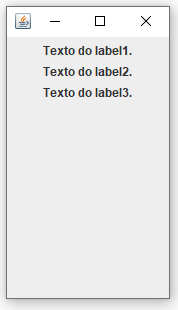
janela.add(label3);

janela.setVisible(true);

}

}





# Aula 74 - Outro exemplo básico de layout manager

**SwingGridLayout.java**

package com.sys4soft;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class SwingGridLayout {

public static void main(String[] args) {

// criar a janela JFrame

JFrame janela = new JFrame("Exemplo");

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

janela.setSize(500, 300);

// label1

JLabel label1 = new JLabel("Texto do label1.");

// label1.setHorizontalAlignment(JLabel.CENTER);

// label2

JLabel label2 = new JLabel("Texto do label2.");

// label3

JLabel label3 = new JLabel("Texto do label3.");

// GridLayout

GridLayout esquema = new GridLayout(2,2);

janela.setLayout(esquema);

janela.add(label1);

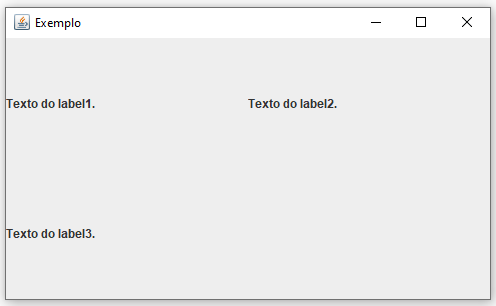
janela.add(label2);

janela.add(label3);

janela.setVisible(true);

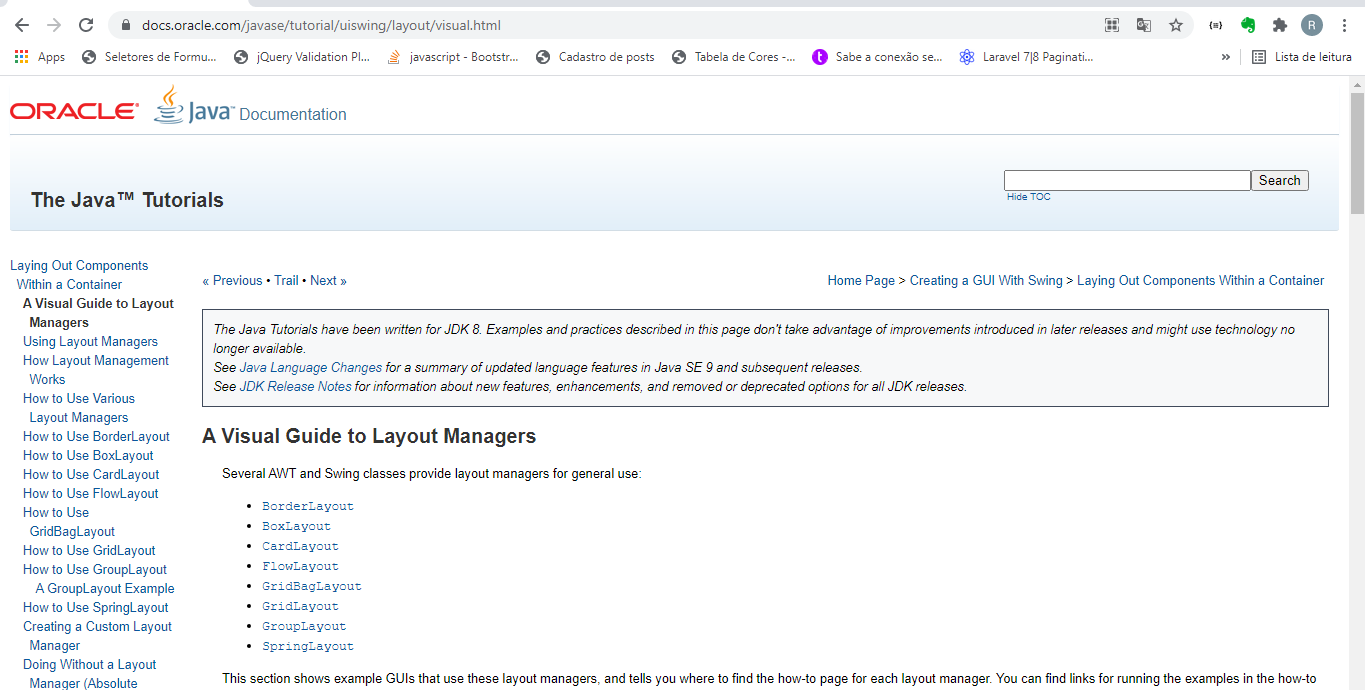
}

}



# Aula 75 - BorderLayout e outros layout manager

https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html



**SwingBorderLayout.java**

package com.sys4soft;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class SwingBorderLayout {

public static void main(String[] args) {

// criar a janela JFrame

JFrame janela = new JFrame("BorderLayout");

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

janela.setSize(500, 300);

// comandos

JButton cmd1 = new JButton("Page Start");

JButton cmd2 = new JButton("Line Start");

JButton cmd3 = new JButton("Center");

JButton cmd4 = new JButton("Line End");

JButton cmd5 = new JButton("Page End");

// layout de janela

janela.setLayout(new BorderLayout());

// adicionar os componentes à janela

janela.add(cmd1, BorderLayout.PAGE\_START);

janela.add(cmd2, BorderLayout.LINE\_START);

janela.add(cmd3, BorderLayout.CENTER);

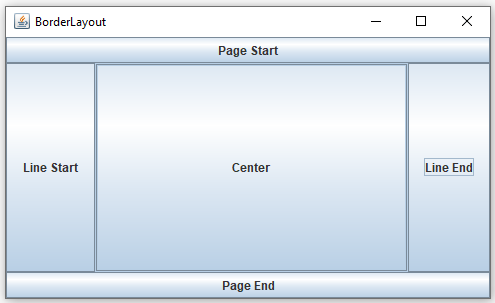
janela.add(cmd4, BorderLayout.LINE\_END);

janela.add(cmd5, BorderLayout.PAGE\_END);

janela.setVisible(true);

}

}



# Aula 76 - Organizar componentes sem layout manager

**SwingNoLayout.java**

package com.sys4soft;

import java.awt.GridLayout;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class SwingNoLayout {

public static void main(String[] args) {

// criar a janela JFrame

JFrame janela = new JFrame("Sem Layout");

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

janela.setSize(500, 300);

// comandos

JButton cmd1 = new JButton("Comando 1");

cmd1.setBounds(10, 10, 120, 30);

JButton cmd2 = new JButton("Comando 1");

cmd2.setBounds(10, 50, 120, 30);

JButton cmd3 = new JButton("Comando 3");

cmd3.setBounds(10, 90, 120, 30);

JButton cmd4 = new JButton("Comando 4");

cmd4.setBounds(150, 100, 150, 90);

janela.setLayout(null);

janela.add(cmd1);

janela.add(cmd2);

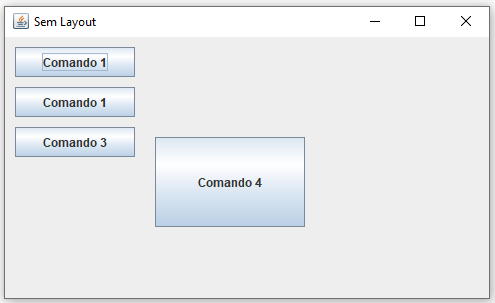
janela.add(cmd3);

janela.add(cmd4);

janela.setVisible(true);

}

}



# Aula 77 - Adicionar um evento a um botao com addActionListened

## Evento click

**ButtonClicked.java**

package com.sys4soft;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

public class ButtonClicked {

public static void main(String[] args) {

// janela

JFrame janela = new JFrame("Evento");

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

janela.setSize(500,300);

janela.setLayout(null);

// botão

JButton cmd1 = new JButton("Clique aqui");

cmd1.setBounds(100, 100, 120, 30);

janela.add(cmd1);

janela.setVisible(true);

// eventos

// cmd1 (click)

cmd1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e){

System.out.println("Botão clicado!");

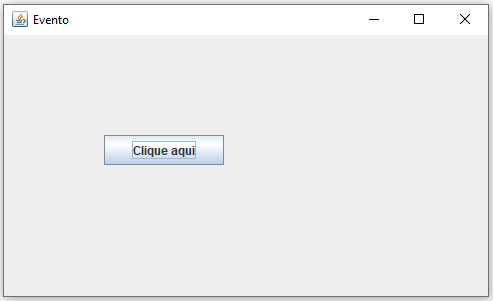
}

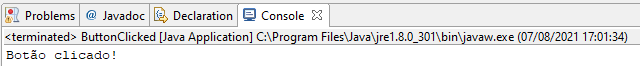
}

);

}

}





# Aula 78 - Usando hereditariedade com JFrame

**SwingJFrame.java**

package com.sys4soft;

public class SwingJFrame {

public static void main(String[] args) {

frmMenuPrincipal formulario = new frmMenuPrincipal();

}

}

**frmMenuPrincipal.java**

package com.sys4soft;

import javax.swing.JFrame;

public class frmMenuPrincipal extends JFrame {

public frmMenuPrincipal(){

// definir o título do formulário

super("Menu Principal");

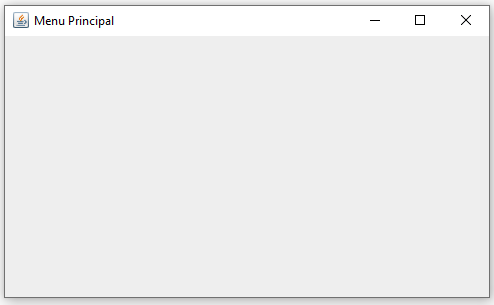
setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(500, 300);

setVisible(true);

}

}



# Aula 79 - Passando parâmetros para a JFrame

**SwingJFrame.java**

package com.sys4soft;

import java.awt.Dimension;

public class SwingJFrame {

public static void main(String[] args) {

frmMenuPrincipal formulario = new frmMenuPrincipal("Menu Principal", new Dimension(700, 500));

formulario.setVisible(true);

}

}

**frmMenuPrincipal.java**

package com.sys4soft;

import java.awt.Dimension;

import javax.swing.JFrame;

public class frmMenuPrincipal extends JFrame {

public frmMenuPrincipal(String titulo, Dimension dimensao){

// definir o título do formulário

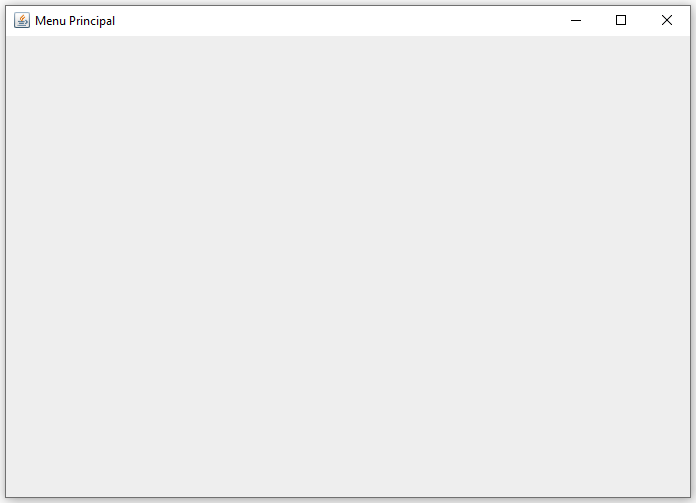
super(titulo);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(dimensao);

}

}



# Aula 80 - Adicionando botões e eventos na JFrame

**SwingJFrame.java**

package com.sys4soft;

import java.awt.Dimension;

public class SwingJFrame {

public static void main(String[] args) {

frmMenuPrincipal formulario = new frmMenuPrincipal("Menu Principal", new Dimension(500, 300));

formulario.setVisible(true);

}

}

**frmMenuPrincipal.java**

package com.sys4soft;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

public class frmMenuPrincipal extends JFrame {

// componentes

JButton cmd\_sair;

JButton cmd\_ajuda;

// eventos

ActionListener cmd\_sair\_click;

// ===========================================================

public frmMenuPrincipal(String titulo, Dimension dimensao){

// definir o título do formulário

super(titulo);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(dimensao);

setLayout(null);

// ======================================================

// eventos

// cmd\_sair\_click

cmd\_sair\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// System.out.println("Foi clicado o botão sair");

System.exit(0);

}

};

// criar os botões

cmd\_sair = new JButton("Sair");

cmd\_sair.setBounds(50, 50, 120, 30);

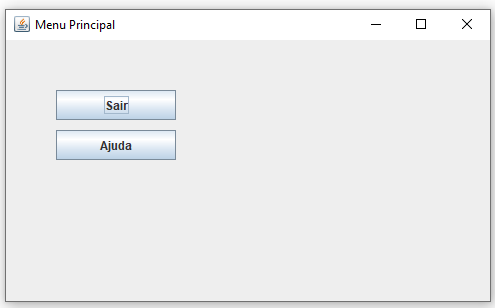
cmd\_sair.addActionListener(cmd\_sair\_click);

// adicionar os botões ao formulário

add(cmd\_sair);

}

}



# Aula 81 - Centrar uma JFrame no ambiente de trabalho

**SwingJFrame.java**

package com.sys4soft;

import java.awt.Dimension;

public class SwingJFrame {

public static void main(String[] args) {

frmMenuPrincipal formulario = new frmMenuPrincipal("Menu Principal", new Dimension(500, 300));

formulario.setVisible(true);

}

}

**frmMenuPrincipal.java**

package com.sys4soft;

import java.awt.Dimension;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

public class frmMenuPrincipal extends JFrame {

// public frmMenuPrincipal(){

// // definir o título do formulário

// super("Menu Principal");

// setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// setSize(500, 300);

// setVisible(true);

// }

// componentes

JButton cmd\_sair;

JButton cmd\_ajuda;

// eventos

ActionListener cmd\_sair\_click;

ActionListener cmd\_ajuda\_click;

// ===========================================================

public frmMenuPrincipal(String titulo, Dimension dimensao){

// definir o título do formulário

super(titulo);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(dimensao);

setLayout(null);

// ======================================================

// eventos

// cmd\_sair\_click

cmd\_sair\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// System.out.println("Foi clicado o botão sair");

System.exit(0);

}

};

// cmd\_ajuda\_click

cmd\_ajuda\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.out.println("Foi clicado o botão de ajuda!");

}

};

// criar os botões

cmd\_sair = new JButton("Sair");

cmd\_sair.setBounds(50, 50, 120, 30);

cmd\_sair.addActionListener(cmd\_sair\_click);

cmd\_ajuda = new JButton("Ajuda");

cmd\_ajuda.setBounds(50, 90, 120, 30);

cmd\_ajuda.addActionListener(cmd\_ajuda\_click);

// adicionar os botões ao formulário

add(cmd\_sair);

add(cmd\_ajuda);

// centralizar o formulário na tela do computador

Dimension dimensao\_monitor = Toolkit.getDefaultToolkit().getScreenSize();

setLocation(dimensao\_monitor.width/2 - dimensao.width/2,

dimensao\_monitor.height/2 - dimensao.height/2);

}

}

# Aula 82 - Centrar uma JFrame e mais sobre dimensões

**SwingJFrame2.java**

package com.sys4soft;

public class SwingJFrame2 {

public static void main(String[] args) {

frmMenu menu = new frmMenu();

}

}

**frmMenu.java**

package com.sys4soft;

import java.awt.Dimension;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class frmMenu extends JFrame {

public frmMenu(){

super("Menu");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// definir JPanel

JPanel painel = new JPanel();

painel.setPreferredSize(new Dimension(500,300));

add(painel);

pack();

// centralizar formulário

setLocationRelativeTo(null);

setVisible(true);

// apresentar duas dimensões de JFrame

Dimension exterior = getSize();

Dimension interior = getContentPane().getSize();

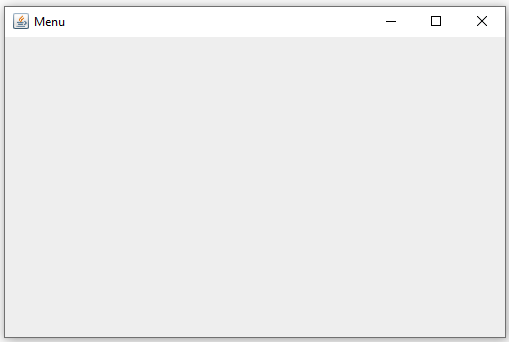
System.out.println(exterior);

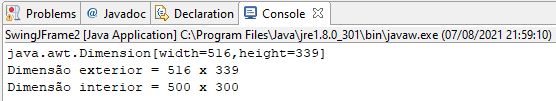
System.out.println("Dimensão exterior = " + exterior.width + " x " + exterior.height);

System.out.println("Dimensão interior = " + interior.width + " x " + interior.height);

}

}





# Aula 83 - Usando JPanel e posicionando componentes

**SwingJFrame3.java**

package com.sys4soft;

public class SwingJFrame3 {

public static void main(String[] args) {

frmMenu menu = new frmMenu();

}

}

**frmMenu.java**

package com.sys4soft;

import java.awt.Dimension;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class frmMenu extends JFrame {

JButton cmd1;

public frmMenu(){

super("Menu");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// definir JPanel

JPanel painel = new JPanel();

painel.setPreferredSize(new Dimension(600,400));

painel.setLayout(null);

add(painel);

pack();

// centralizar formulário

setLocationRelativeTo(null);

int largura = getContentPane().getSize().width;

int altura = getContentPane().getSize().height;

// botão

cmd1 = new JButton("Sair");

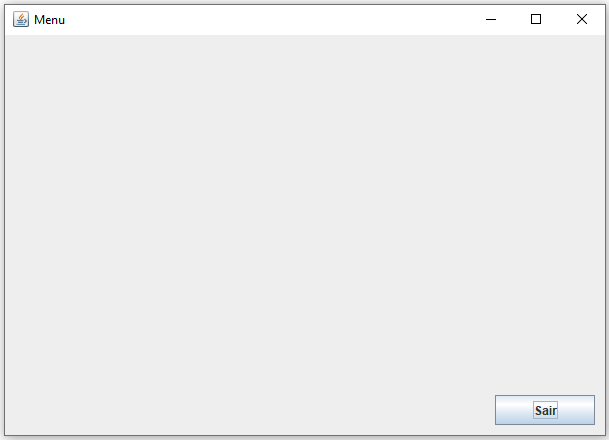
cmd1.setBounds(largura - 110,altura - 40,100,30);

painel.add(cmd1);

setVisible(true);

}

}



# Aula 84 - Duplicação de projetos em Eclipse e JLabel

**SwingJFrame4.java**

package com.sys4soft;

public class SwingJFrame4 {

public static void main(String[] args) {

frmMenu menu = new frmMenu();

}

}

**frmMenu.java**

package com.sys4soft;

import java.awt.Dimension;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

public class frmMenu extends JFrame {

JButton cmd1;

JLabel label1;

public frmMenu(){

super("Menu");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// definir JPanel

JPanel painel = new JPanel();

painel.setPreferredSize(new Dimension(600,400));

painel.setLayout(null);

add(painel);

pack();

// centralizar formulário

setLocationRelativeTo(null);

int largura = getContentPane().getSize().width;

int altura = getContentPane().getSize().height;

// botão

cmd1 = new JButton("Sair");

cmd1.setBounds(largura - 110,altura - 40,100,30);

painel.add(cmd1);

// label

label1 = new JLabel("Texto do label.");

label1.setBounds(50,50,200,20);

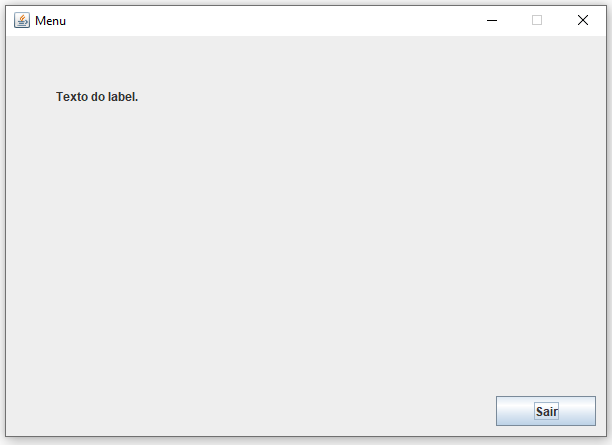
painel.add(label1);

setResizable(false);

setVisible(true);

}

}



# Aula 85 - Alterar programaticamente o texto de JLabel

**SwingJFrame5.java**

package com.sys4soft;

public class SwingJFrame5 {

public static void main(String[] args) {

frmMenu menu = new frmMenu();

}

}

**frmMenu.java**

package com.sys4soft;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

public class frmMenu extends JFrame {

JButton cmd1;

JLabel label1;

ActionListener cmd1\_click;

public frmMenu(){

super("Menu");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// definir JPanel

JPanel painel = new JPanel();

painel.setPreferredSize(new Dimension(600,400));

painel.setLayout(null);

add(painel);

pack();

// centralizar formulário

setLocationRelativeTo(null);

int largura = getContentPane().getSize().width;

int altura = getContentPane().getSize().height;

// criar o evento que permite alterar o conteúdo do label1

cmd1\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

label1.setText("Foi alterado o texto do label");

label1.setForeground(Color.RED);

label1.setOpaque(true);

label1.setBackground(Color.WHITE);

}

};

// botão

cmd1 = new JButton("Alterar label");

cmd1.setBounds(largura - 130,altura - 40,120,30);

cmd1.addActionListener(cmd1\_click);

painel.add(cmd1);

// label

label1 = new JLabel("Texto do label.");

label1.setBounds(50,50,200,20);

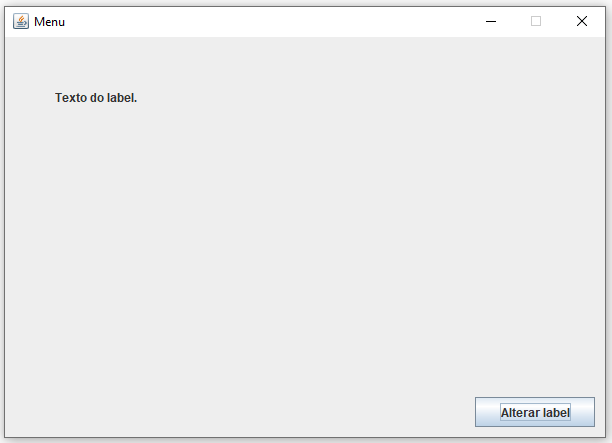
painel.add(label1);

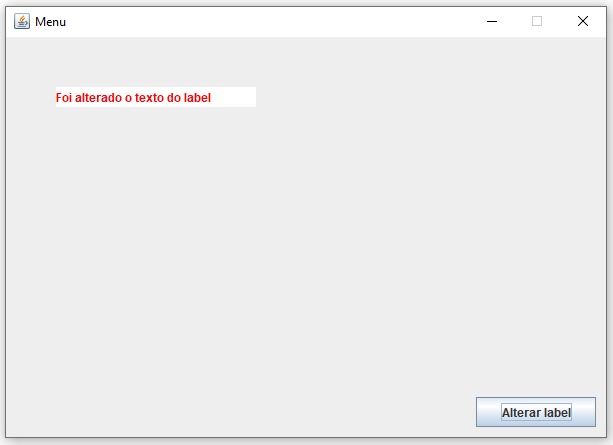
setResizable(false);

setVisible(true);

}

}





# Aula 86 - Preparação de projeto para abordagem de Swing components

**ProjetoSwing.java**

package com.sys4soft;

public class ProjetoSwing {

public static void main(String[] args) {

frmFormulario form = new frmFormulario();

}

}

**frmFormulario.java**

package com.sys4soft;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class frmFormulario extends JFrame {

JPanel painel;

JButton cmd;

ActionListener cmd\_click;

// construtor

public frmFormulario(){

super("Formulário de Testes - Swing Components");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// JPanel

painel = new JPanel();

painel.setPreferredSize(new Dimension(600,400));

painel.setLayout(null);

add(painel);

pack();

// componente

// cmd

cmd\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// código a executar no evento click

}

};

// criar cmd

cmd = new JButton("Clique aqui");

cmd.setBounds(400, 350, 120, 30);

cmd.addActionListener(cmd\_click);

painel.add(cmd);

setLocationRelativeTo(null);

setVisible(true);

}

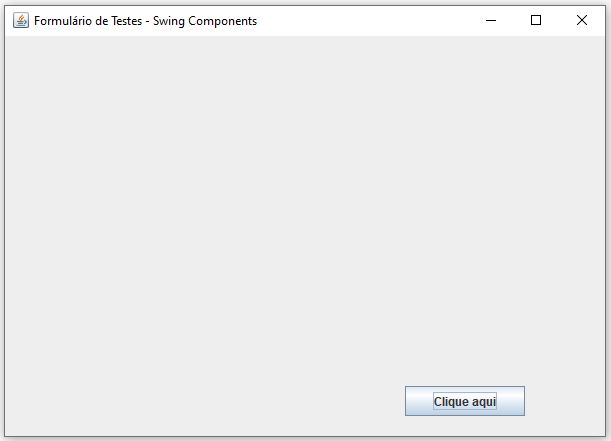
public void Console(String str){

// apresenta texto no console

System.out.println(str);

}

}



# Aula 87 - Swing components - JCheckBox

**ProjetoSwing.java**

package com.sys4soft;

public class ProjetoSwing {

public static void main(String[] args) {

frmFormulario form = new frmFormulario();

}

}

**frmFormulario.java**

package com.sys4soft;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import javax.swing.JButton;

import javax.swing.JCheckBox;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class frmFormulario extends JFrame {

JPanel painel;

JButton cmd;

ActionListener cmd\_click;

JCheckBox check;

ItemListener check\_selected;

// construtor

public frmFormulario(){

super("Formulário de Testes - JCheckBox");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// JPanel

painel = new JPanel();

painel.setPreferredSize(new Dimension(600,400));

painel.setLayout(null);

add(painel);

pack();

// componente

check\_selected = new ItemListener() {

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getStateChange() == ItemEvent.SELECTED){

Console("Sim");

} else {

Console("Não");

}

}

};

check = new JCheckBox("Gravar tudo ao sair");

check.setBounds(50,50,200,30);

check.addItemListener(check\_selected);

painel.add(check);

// cmd

cmd\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// código a executar no evento click

}

};

// criar cmd

cmd = new JButton("Clique aqui");

cmd.setBounds(400, 350, 120, 30);

cmd.addActionListener(cmd\_click);

painel.add(cmd);

setLocationRelativeTo(null);

setVisible(true);

}

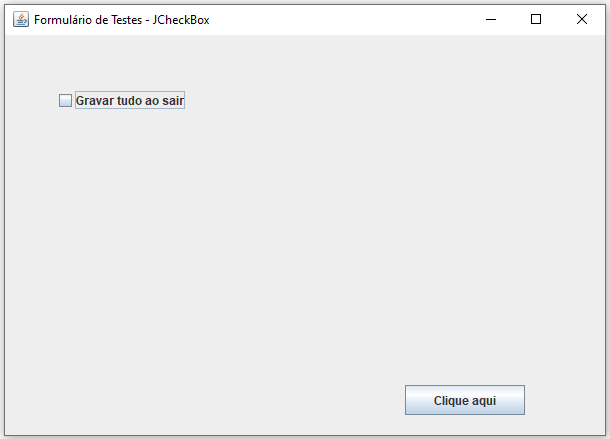
public void Console(String str){

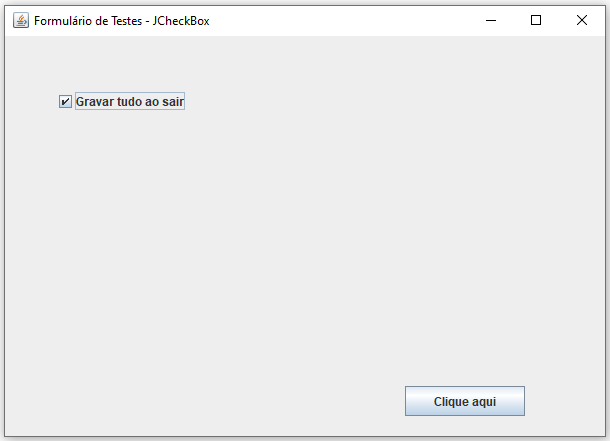
// apresenta texto no console

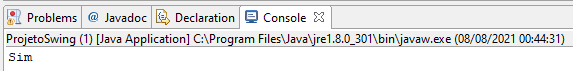
System.out.println(str);

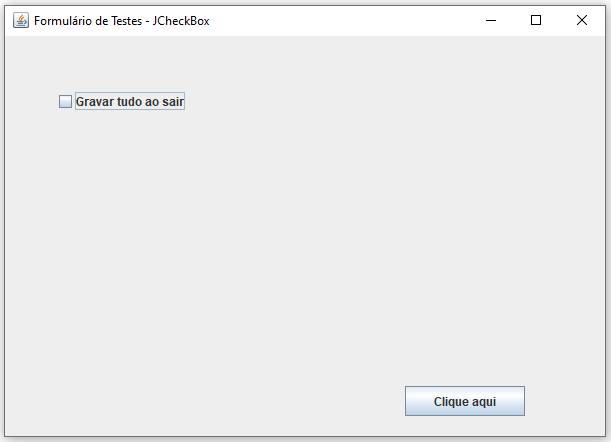
}

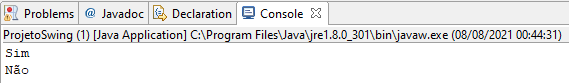
}











# Aula 88 - Swing components - JRadioButton e ButtonGroup

**ProjetoSwing.java**

package com.sys4soft;

public class ProjetoSwing {

public static void main(String[] args) {

frmFormulario form = new frmFormulario();

}

}

**frmFormulario.java**

package com.sys4soft;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JRadioButton;

public class frmFormulario extends JFrame {

JPanel painel;

JButton cmd;

ActionListener cmd\_click;

JRadioButton r1;

JRadioButton r2;

JRadioButton r3;

JRadioButton r4;

ButtonGroup grupo1;

// construtor

public frmFormulario(){

super("Formulário de Testes - Swing Components");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// JPanel

painel = new JPanel();

painel.setPreferredSize(new Dimension(600,400));

painel.setLayout(null);

add(painel);

pack();

// componente

r1 = new JRadioButton("Solteiro");

r1.setActionCommand("solteiro");

r1.setSelected(true);

r1.setBounds(20, 20, 200, 30);

r2 = new JRadioButton("Casado");

r2.setActionCommand("casado");

r2.setBounds(20, 40, 200, 30);

r3 = new JRadioButton("Divorcidado");

r3.setActionCommand("divorciado");

r3.setBounds(20, 60, 200, 30);

r4 = new JRadioButton("Viúvo");

r4.setActionCommand("viúvo");

r4.setBounds(20, 80, 200, 30);

painel.add(r1);

painel.add(r2);

painel.add(r3);

painel.add(r4);

// grupo1

grupo1 = new ButtonGroup();

grupo1.add(r1);

grupo1.add(r2);

grupo1.add(r3);

grupo1.add(r4);

// cmd

cmd\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// código a executar no evento click

String item = grupo1.getSelection().getActionCommand();

Console(item);

if(item == "casado"){

Console("Este indivíduo é casado.");

}

}

};

// criar cmd

cmd = new JButton("Clique aqui");

cmd.setBounds(400, 350, 120, 30);

cmd.addActionListener(cmd\_click);

painel.add(cmd);

setLocationRelativeTo(null);

setVisible(true);

}

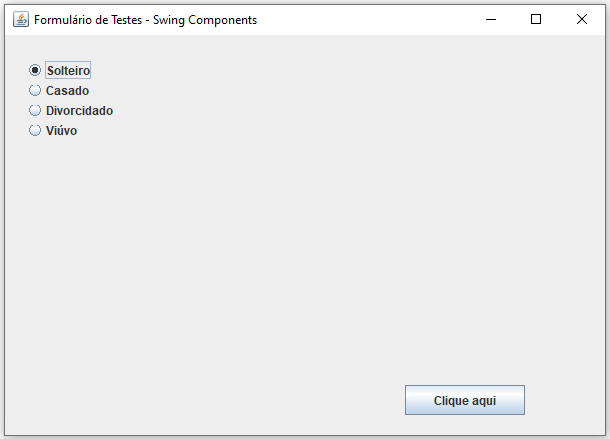
public void Console(String str){

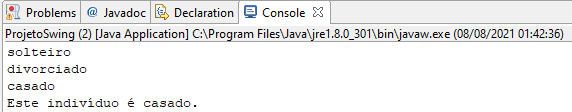
// apresenta texto no console

System.out.println(str);

}

}





# Aula 89 - Swing components - JComboBox

**ProjetoSwing.java**

package com.sys4soft;

public class ProjetoSwing {

public static void main(String[] args) {

frmFormulario form = new frmFormulario();

}

}

**frmFormulario.java**

package com.sys4soft;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class frmFormulario extends JFrame {

JPanel painel;

JButton cmd;

ActionListener cmd\_click;

ActionListener combo\_selected\_changed;

JComboBox combo;

// construtor

public frmFormulario(){

super("Formulário de Testes - Swing Components");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// JPanel

painel = new JPanel();

painel.setPreferredSize(new Dimension(600,400));

painel.setLayout(null);

add(painel);

pack();

combo\_selected\_changed = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JComboBox temp = (JComboBox)e.getSource();

String valor = temp.getSelectedItem().toString();

Console(valor);

}

};

// componente

String[] array\_itens = {"Primeiro", "Segundo", "Terceiro", "Quarto"};

combo = new JComboBox(array\_itens);

combo.setBounds(50, 50, 150, 20);

combo.addActionListener(combo\_selected\_changed);

// combo.setEditable(true);

// combo.setSelectedIndex(2);

painel.add(combo);

// cmd

cmd\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// código a executar no evento click

Console(combo.getSelectedItem().toString());

// Console(String.valueOf(combo.getSelectedIndex()));

}

};

// criar cmd

cmd = new JButton("Clique aqui");

cmd.setBounds(400, 350, 120, 30);

cmd.addActionListener(cmd\_click);

painel.add(cmd);

setLocationRelativeTo(null);

setVisible(true);

}

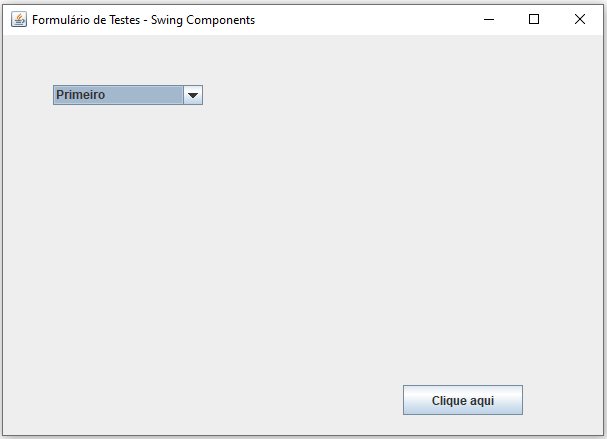
public void Console(String str){

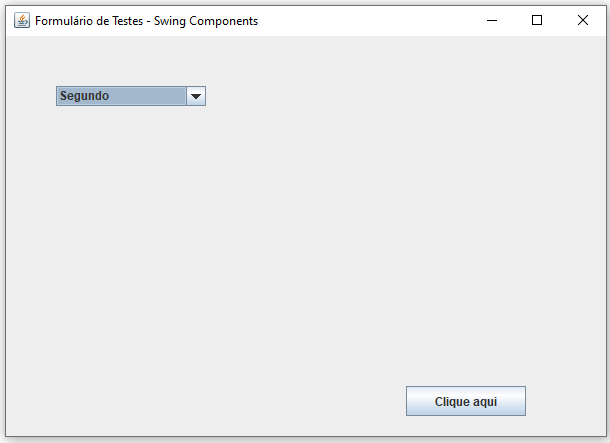
// apresenta texto no console

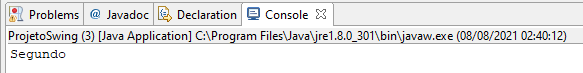
System.out.println(str);

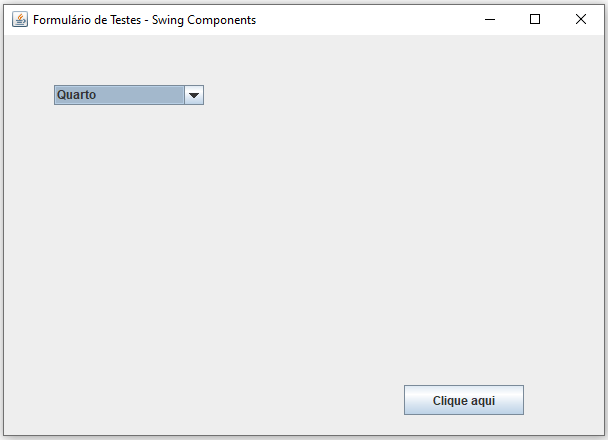
}

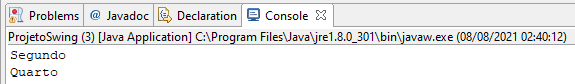
}











# Aula 90 - Swing components - JTextField

**ProjetoSwing.java**

package com.sys4soft;

public class ProjetoSwing {

public static void main(String[] args) {

frmFormulario form = new frmFormulario();

}

}

**frmFormulario.java**

package com.sys4soft;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class frmFormulario extends JFrame {

JPanel painel;

JButton cmd;

ActionListener cmd\_click;

JTextField textbox1;

JTextField textbox2;

// construtor

public frmFormulario(){

super("Formulário de Testes - Swing Components");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// JPanel

painel = new JPanel();

painel.setPreferredSize(new Dimension(600,400));

painel.setLayout(null);

add(painel);

pack();

// componente

textbox1 = new JTextField();

textbox1.setBounds(50, 50, 250, 30);

textbox1.setHorizontalAlignment(JTextField.LEFT);

textbox2 = new JTextField();

textbox2.setBounds(50, 100, 250, 30);

textbox2.setHorizontalAlignment(JTextField.LEFT);

// cmd

cmd\_click = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// código a executar no evento click

String texto = textbox1.getText() + " " + textbox2.getText();

Console(texto);

}

};

// criar cmd

cmd = new JButton("Clique aqui");

cmd.setBounds(400, 350, 120, 30);

cmd.addActionListener(cmd\_click);

painel.add(cmd);

painel.add(textbox1);

painel.add(textbox2);

setLocationRelativeTo(null);

setVisible(true);

}

public void Console(String str){

// apresenta texto no console

System.out.println(str);

}

}

