

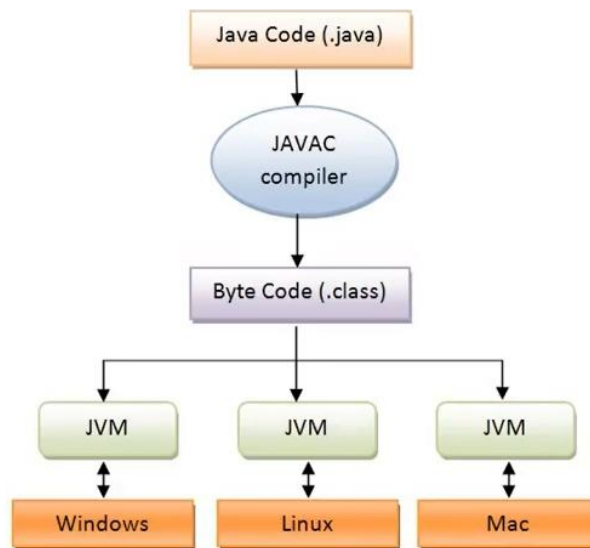
# Curso de Java

## João Ribeiro (Sys4Soft)

Vídeos: [https://www.youtube.com/watch?v=Hn3FPTNbjHA&list=PLXik\\_5Br-zO-9NPaurIpUFfjrhJIESJ6R](https://www.youtube.com/watch?v=Hn3FPTNbjHA&list=PLXik_5Br-zO-9NPaurIpUFfjrhJIESJ6R)

Resumo do curso feito por Roberto Pinheiro

### 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>

ORACLE

ProdutosSetoresRecursosSuporteEventosDesenvolvedor

Java / Technical Details / Java SE / Java SE Development Kit 8 Downloads

Java SE DownloadsJava SE Subscriptions

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Important Oracle JDK License Update

The Oracle JDK License has changed for releases starting April 16, 2019.

The new Oracle Technology Network License Agreement for Oracle Java SE is substantially different from prior Oracle JDK licenses. The new license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. An FAQ is available [here](#).

Commercial license and support is available with a low cost [Java SE Subscription](#).

Oracle also provides the latest OpenJDK release under the open source [GPL License](#) at [jdk.java.net](#).

### Java SE Development Kit 8u301

This software is licensed under the [Oracle Technology Network License Agreement](#) for Oracle Java SE

Product / File Description	File Size	Download
Linux ARM 64 RPM Package	59.15 MB	<a href="#">jdk-8u301-linux-aarch64.rpm</a>
Linux ARM 64 Compressed Archive	70.84 MB	<a href="#">jdk-8u301-linux-aarch64.tar.gz</a>
Linux ARM 32 Hard Float ABI	73.55 MB	<a href="#">jdk-8u301-linux-arm32-vfp-hflt.tar.gz</a>
Linux x86 RPM Package	109.49 MB	<a href="#">jdk-8u301-linux-i586.rpm</a>
Linux x86 Compressed Archive	138.48 MB	<a href="#">jdk-8u301-linux-i586.tar.gz</a>
Linux x64 RPM Package	109.24 MB	<a href="#">jdk-8u301-linux-x64.rpm</a>
Linux x64 Compressed Archive	138.78 MB	<a href="#">jdk-8u301-linux-x64.tar.gz</a>
macOS x64	197.26 MB	<a href="#">jdk-8u301-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	133.66 MB	<a href="#">jdk-8u301-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	94.8 MB	<a href="#">jdk-8u301-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	134.42 MB	<a href="#">jdk-8u301-solaris-x64.tar.Z</a>
Solaris x64	92.66 MB	<a href="#">jdk-8u301-solaris-x64.tar.gz</a>
Windows x86	156.45 MB	<a href="#">jdk-8u301-windows-i586.exe</a>
Windows x64	169.46 MB	<a href="#">jdk-8u301-windows-x64.exe</a>

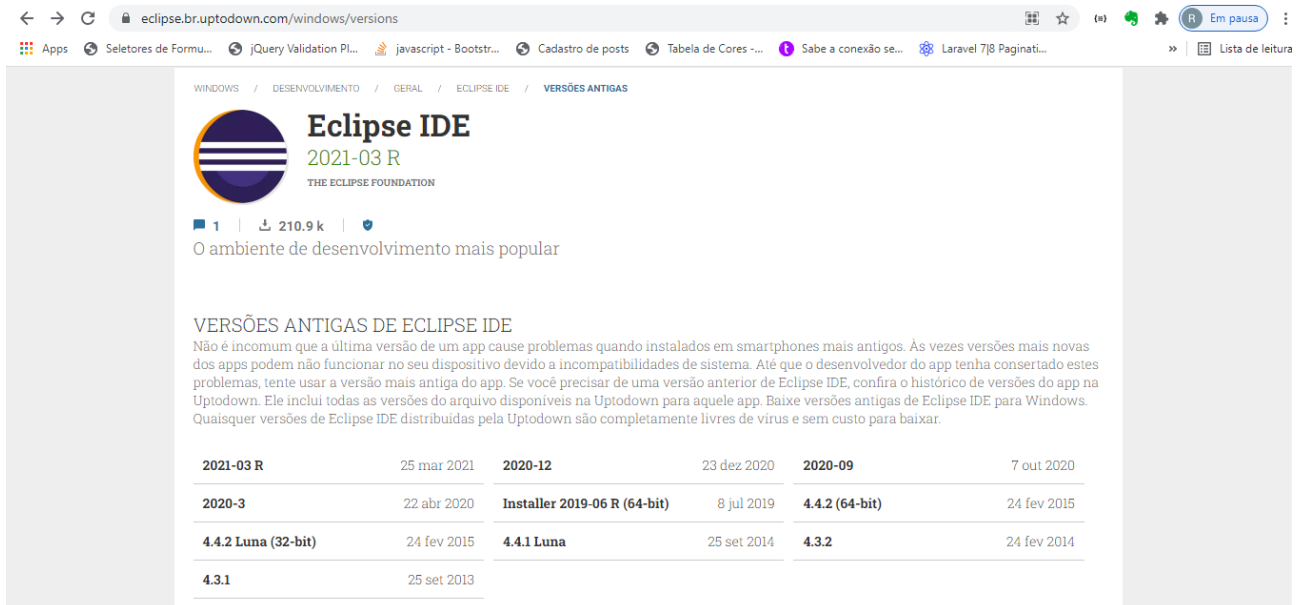
# Eclipse

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

[www.eclipse.org](http://www.eclipse.org)

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

WINDOWS / DESENVOLVIMENTO / GERAL / ECLIPSE IDE / **VERSÕES ANTIGAS**

 **Eclipse IDE**  
2021-03 R  
THE ECLIPSE FOUNDATION

1 | 210.9 k |

O ambiente de desenvolvimento mais popular

**VERSÕES ANTIGAS DE ECLIPSE IDE**

Não é incomum que a última versão de um app cause problemas quando instalados em smartphones mais antigos. Às vezes versões mais novas dos apps podem não funcionar no seu dispositivo devido a incompatibilidades de sistema. Até que o desenvolvedor do app tenha consertado estes problemas, tente usar a versão mais antiga do app. Se você precisar de uma versão anterior de Eclipse IDE, confira o histórico de versões do app na Uptodown. Ele inclui todas as versões do arquivo disponíveis na Uptodown para aquele app. Baixe versões antigas de Eclipse IDE para Windows. Quaisquer versões de Eclipse IDE distribuídas pela Uptodown são completamente livres de vírus e sem custo para baixar.

<b>2021-03 R</b>	25 mar 2021	<b>2020-12</b>	23 dez 2020	<b>2020-09</b>	7 out 2020
<b>2020-3</b>	22 abr 2020	<b>Installer 2019-06 R (64-bit)</b>	8 jul 2019	<b>4.4.2 (64-bit)</b>	24 fev 2015
<b>4.4.2 Luna (32-bit)</b>	24 fev 2015	<b>4.4.1 Luna</b>	25 set 2014	<b>4.3.2</b>	24 fev 2014
<b>4.3.1</b>	25 set 2013				

- 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

```
C:\java_sys4soft>javac EstouAqui.java  
  
C:\java_sys4soft>dir  
O volume na unidade C é W10-161  
O Número de Série do Volume é 76FE-D12F  
  
Pasta de C:\java_sys4soft  
  
01/08/2021  22:21    <DIR>          .  
01/08/2021  22:21    <DIR>          ..  
01/08/2021  22:21                423 EstouAqui.class  
01/08/2021  22:08                125 EstouAqui.java  
                2 arquivo(s)          548 bytes  
                2 pasta(s) 45.037.064.192 bytes disponíveis
```

- Para executá-lo, entre com:

```
java EstouAqui
```

```
C:\java_sys4soft>java EstouAqui  
Estou Aqui!
```

## 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**

```
C:\java_sys4soft>dir  
O volume na unidade C é W10-161  
O Número de Série do Volume é 76FE-D12F  
  
Pasta de C:\java_sys4soft  
  
01/08/2021  22:48    <DIR>          .  
01/08/2021  22:48    <DIR>          ..  
01/08/2021  22:21             423 EstouAqui.class  
01/08/2021  22:43             111 EstouAqui.java  
01/08/2021  22:48             441 MeuProjeto.class  
01/08/2021  22:45             128 MeuProjeto.java  
               4 arquivo(s)          1.103 bytes  
               2 pasta(s)  44.711.927.808 bytes disponíveis
```

- 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`

```
C:\java_sys4soft>jar cvfm MeuProjeto.jar manifest.txt MeuProjeto.class  
manifesto adicionado  
adicionando: MeuProjeto.class(entrada = 441) (saída= 309)(compactado 29%)
```

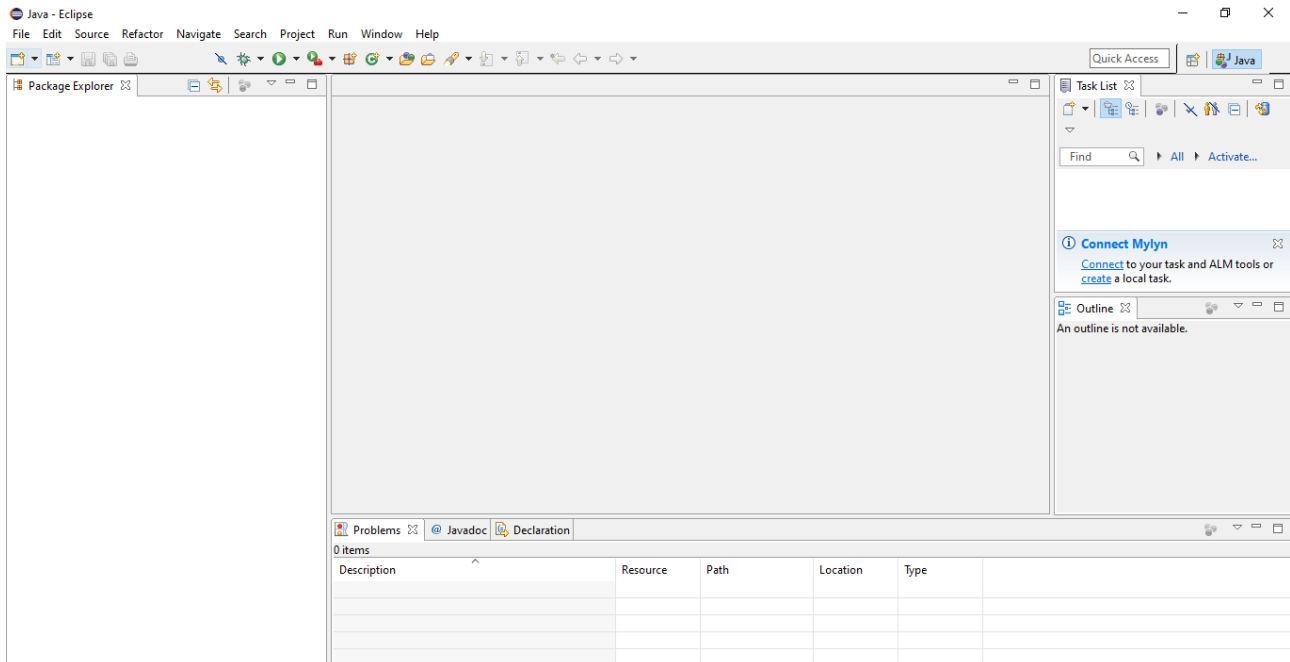
- Para executar o arquivo .JAR:

`java -jar MeuProjeto.jar`

```
C:\java_sys4soft>java -jar MeuProjeto.jar  
O MEU NOVO PROJETO EM JAVA!
```

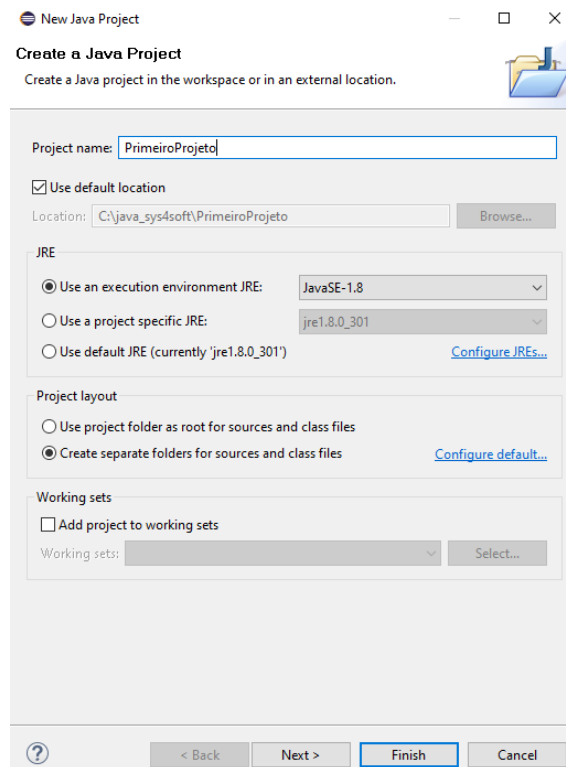
# Aula 06 - Criando o primeiro projeto Java no Eclipse

- Abra o Eclipse

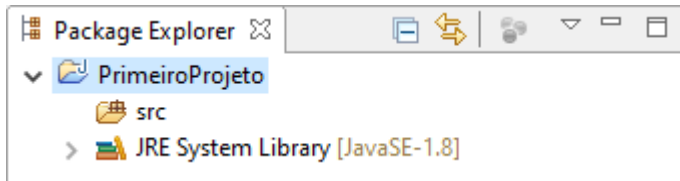


## Criando um novo projeto

File -> New -> Java Project



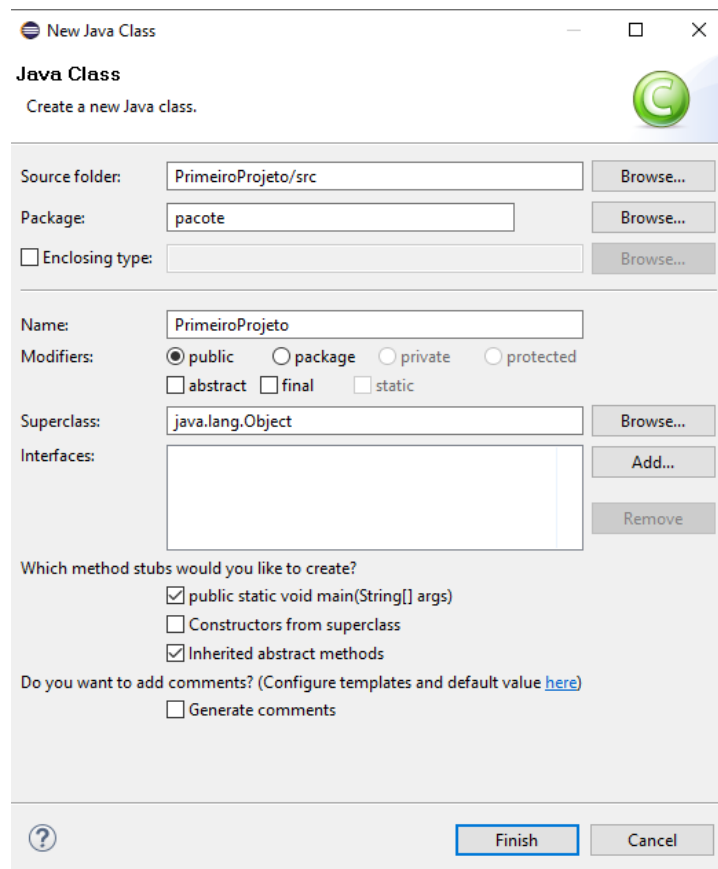
- Clique no botão "Finish"



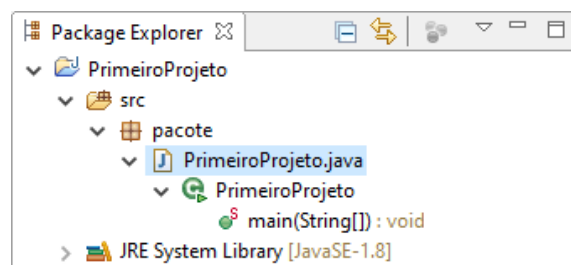
## Acrescentando uma classe ao projeto

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

New -> Class



- Clique no botão "Finish"



## PrimeiroProjeto.java

```
package pacote;

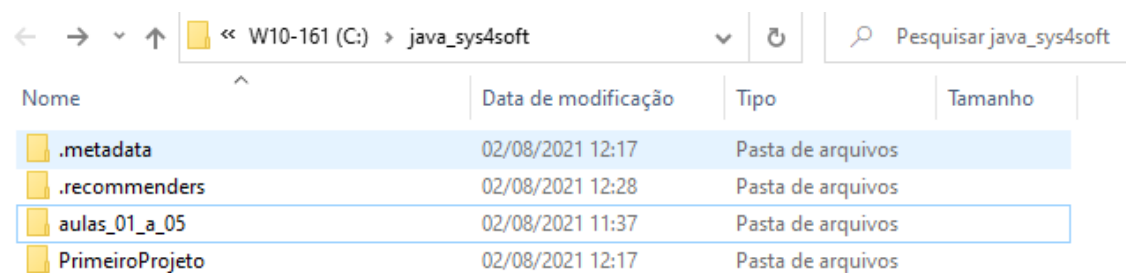
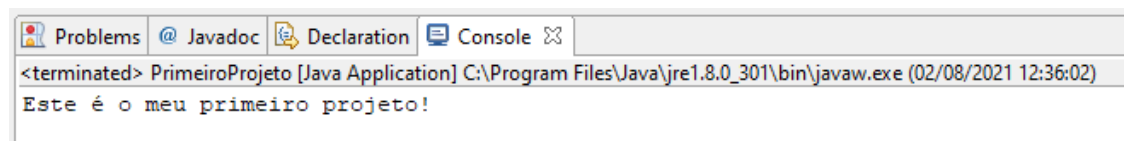
public class PrimeiroProjeto {

    public static void main(String[] args) {
        System.out.println("Este é o meu primeiro projeto!");
    }

}
```

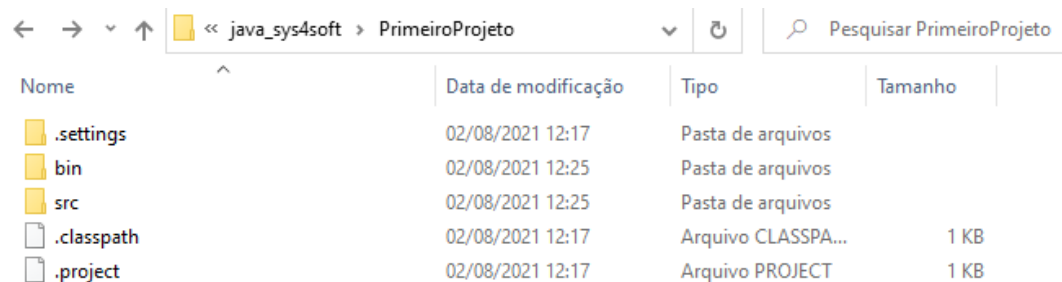
## Executando o projeto

- Clique no botão "Run": 



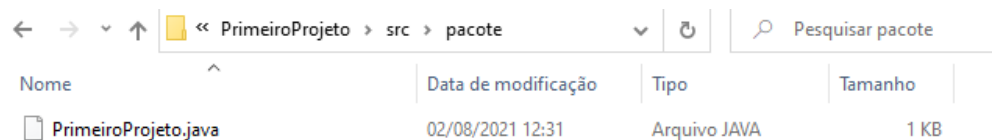
The screenshot shows the file explorer with the path `W10-161 (C:) > java_sys4soft`. The table below represents the contents of this directory.

Nome	Data de modificação	Tipo	Tamanho
.metadata	02/08/2021 12:17	Pasta de arquivos	
.recommenders	02/08/2021 12:28	Pasta de arquivos	
aulas_01_a_05	02/08/2021 11:37	Pasta de arquivos	
PrimeiroProjeto	02/08/2021 12:17	Pasta de arquivos	



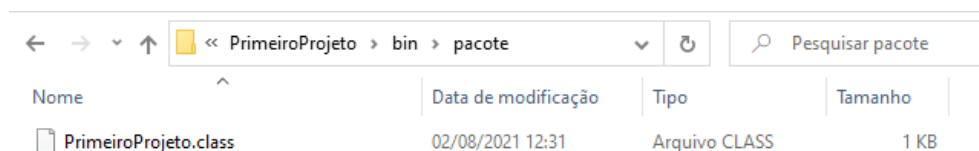
The screenshot shows the file explorer with the path `java_sys4soft > PrimeiroProjeto`. The table below represents the contents of this directory.

Nome	Data de modificação	Tipo	Tamanho
.settings	02/08/2021 12:17	Pasta de arquivos	
bin	02/08/2021 12:25	Pasta de arquivos	
src	02/08/2021 12:25	Pasta de arquivos	
.classpath	02/08/2021 12:17	Arquivo CLASSPA...	1 KB
.project	02/08/2021 12:17	Arquivo PROJECT	1 KB



The screenshot shows the file explorer with the path `PrimeiroProjeto > src > pacote`. The table below represents the contents of this directory.

Nome	Data de modificação	Tipo	Tamanho
PrimeiroProjeto.java	02/08/2021 12:31	Arquivo JAVA	1 KB



The screenshot shows the file explorer with the path `PrimeiroProjeto > bin > pacote`. The table below represents the contents of this directory.

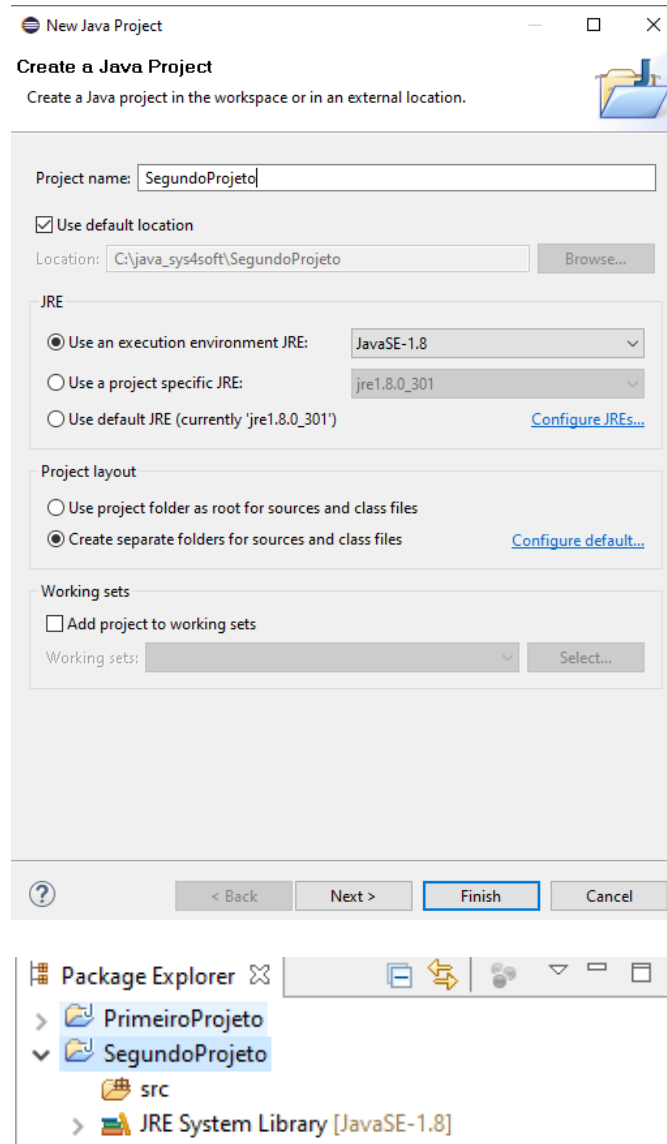
Nome	Data de modificação	Tipo	Tamanho
PrimeiroProjeto.class	02/08/2021 12:31	Arquivo CLASS	1 KB



## Aula 07 - Criando o segundo projeto Java no Eclipse

- Crie um novo projeto:

File -> New -> Java Project



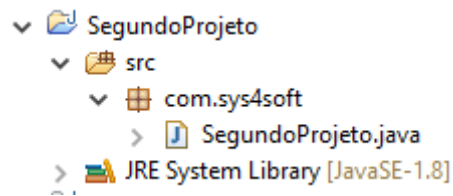
- Clique com o botão direito do mouse sobre a pasta **src** e crie uma nova classe:

New -> Class

Package: com.sys4soft

Name: SegundoProjeto

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



### SegundoProjeto.java

```
package com.sys4soft;
```

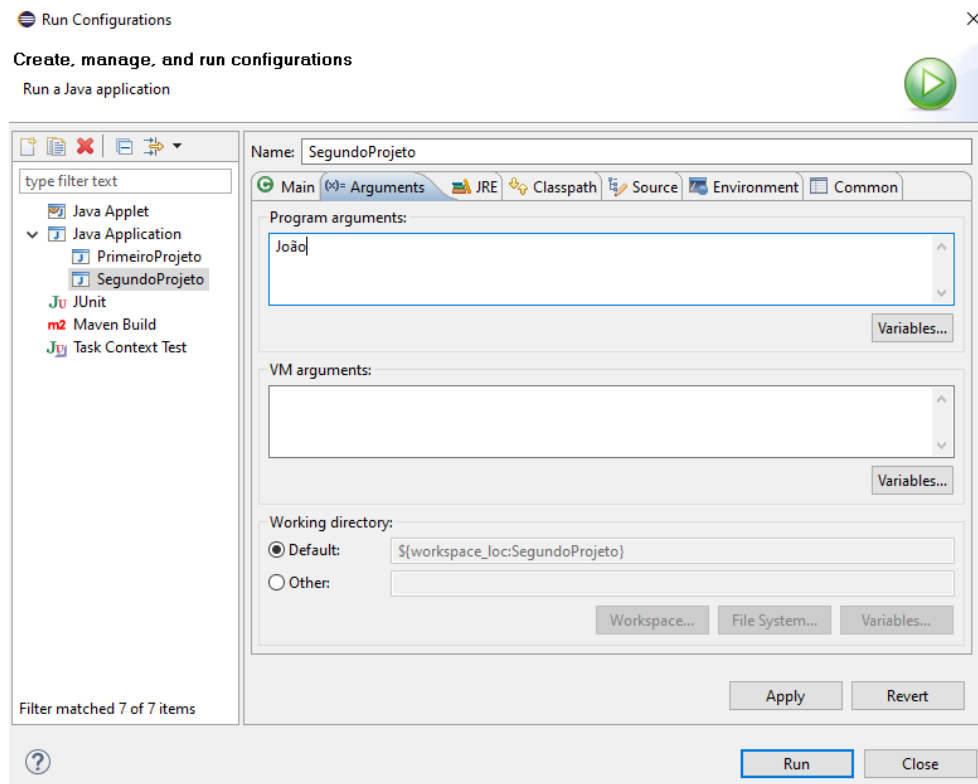
```
public class SegundoProjeto {
```

```
    public static void main(String[] args) {  
        System.out.println("Bom dia, " + args[0] + "!");  
    }
```

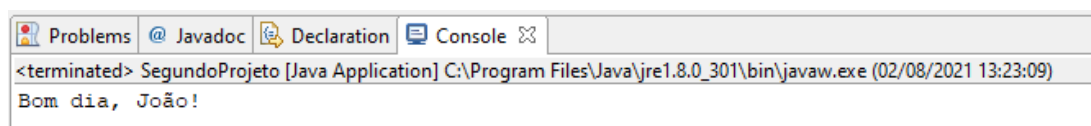
```
}
```

- 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.



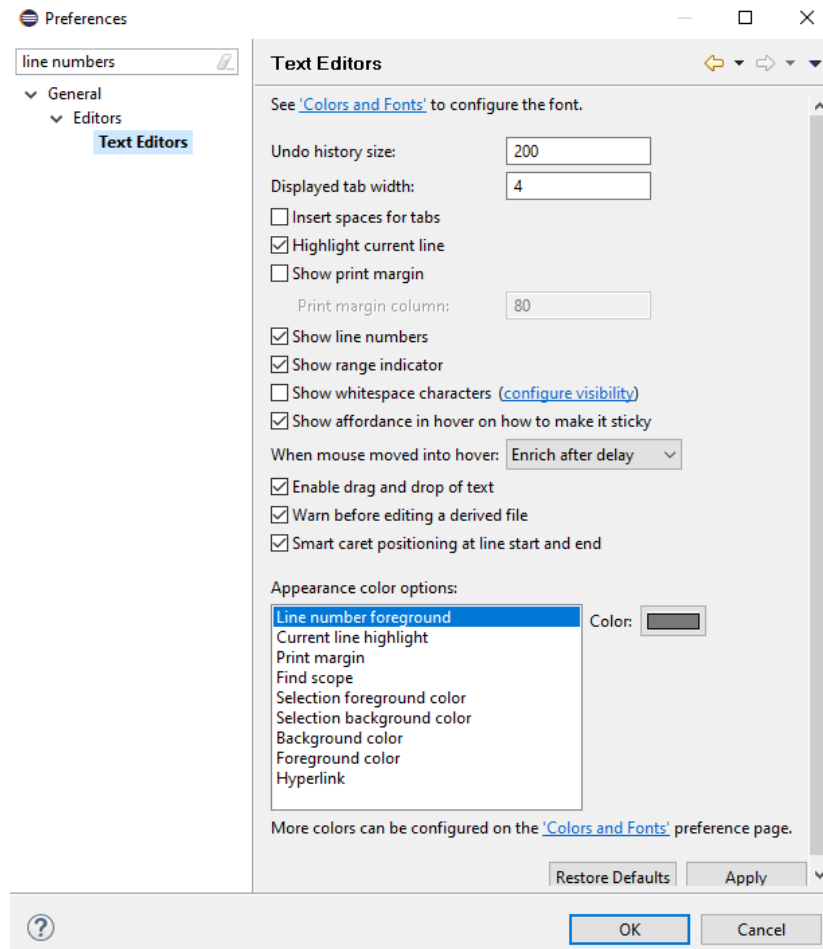
- Em seguida clique no botão "Run"



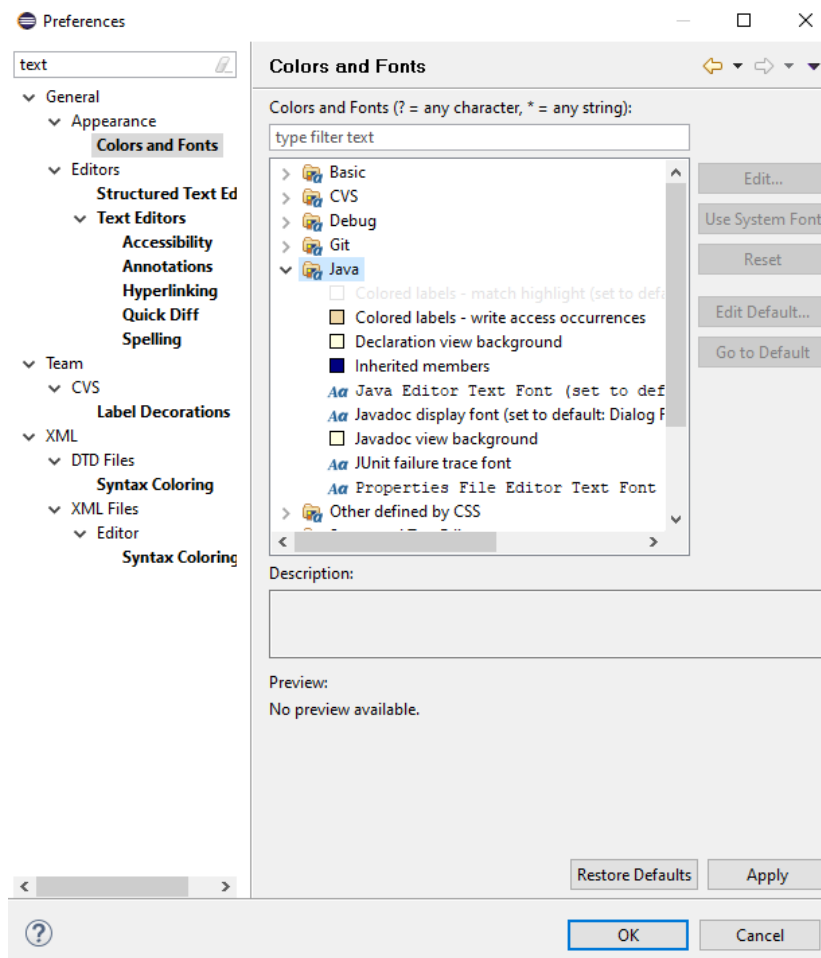
# Aula 08 - Configurando o Eclipse

## Exibindo o número das linhas

Window -> Preferences

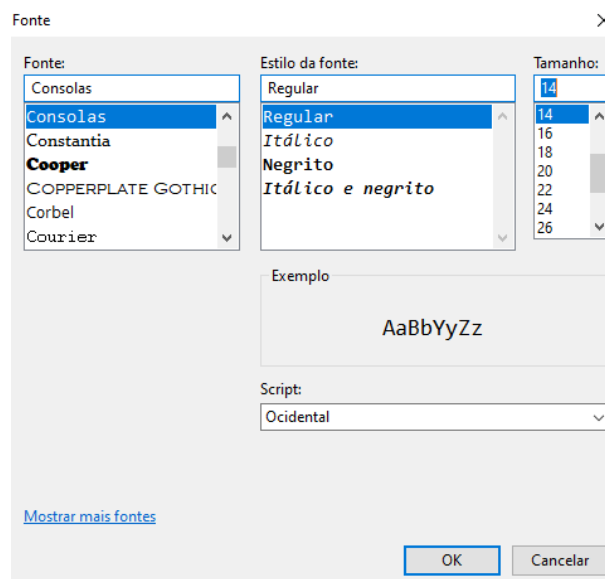


## Alterando a fonte



- Clique em "Java Editor Text Font"

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

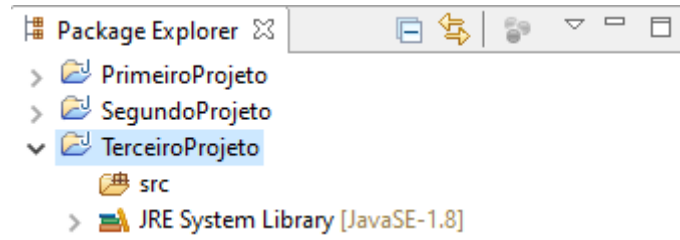


## Aula 09 - Terceiro projeto Java no Eclipse

- Crie um novo projeto

File -> New -> Java Project

- Nomeie como "TerceiroProjeto"



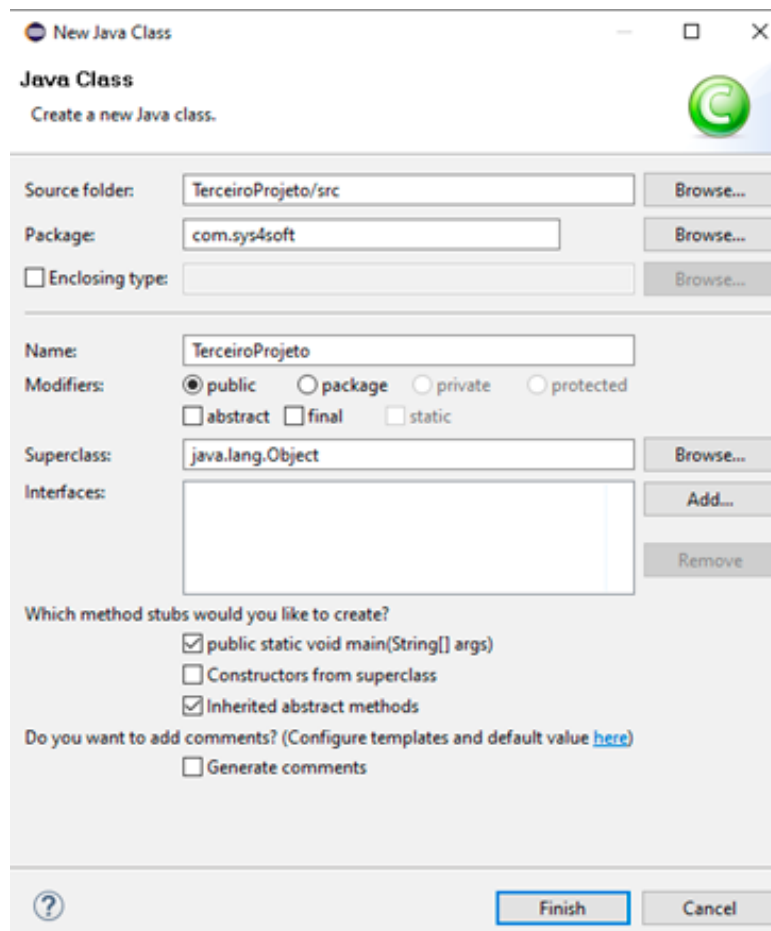
- Clique com o botão direito do mouse sobre a pasta `src` e crie uma nova classe:

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);

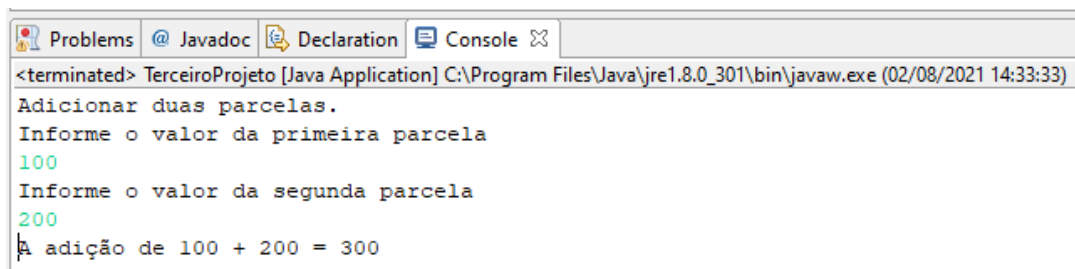
        // pedido da primeira parcela
        System.out.println("Informe o valor da primeira parcela");
        parcela1 = input.nextInt();

        // 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):



```
<terminated> TerceiroProjeto [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (02/08/2021 14:33:33)
Adicionar duas parcelas.
Informe o valor da primeira parcela
100
Informe o valor da segunda parcela
200
A adição de 100 + 200 = 300
```

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

## Documentação Java SE8

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

The screenshot shows the Java Platform, Standard Edition 8 API Specification page. The browser address bar displays <https://docs.oracle.com/javase/8/docs/api/>. The page has a navigation bar with tabs: OVERVIEW, PACKAGE, CLASS, USE, TREE, DEPRECATED, INDEX, and HELP. The left sidebar contains a tree view with 'All Classes' and 'All Profiles'. The main content area is titled 'Java™ Platform, Standard Edition 8 API Specification' and includes a description: 'This document is the API specification for the Java™ Platform, Standard Edition.' Below this, there is a 'Profiles' section with a list of profiles: compact1, compact2, and compact3. A 'Packages' section is also present, showing a table with columns 'Package' and 'Description'. The table lists several packages: java.applet, java.awt, java.awt.color, and java.awt.datatransfer.

Package	Description
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.

- 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
```

```
*/
```

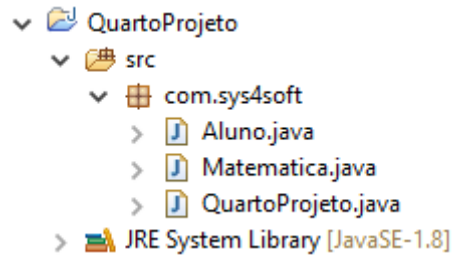
## Aula 13 - Breve introdução às classes em Java

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();
```

```
    }
```

```
}
```

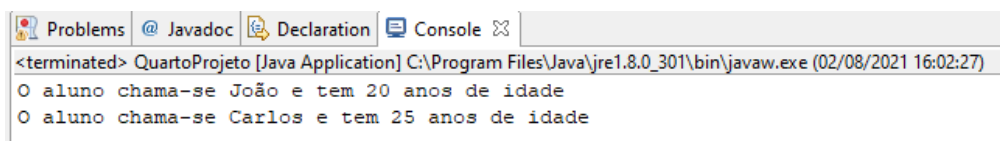
## Aluno.java

```
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):



The screenshot shows an IDE window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of a Java application. The output consists of two lines: "O aluno chama-se João e tem 20 anos de idade" and "O aluno chama-se Carlos e tem 25 anos de idade". The window title bar indicates the application is "QuartoProjeto [Java Application]" and the path is "C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe" with a timestamp of "02/08/2021 16:02:27".

```
<terminated> QuartoProjeto [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (02/08/2021 16:02:27)
O aluno chama-se João e tem 20 anos de idade
O aluno chama-se Carlos e tem 25 anos de idade
```

## 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);
    }

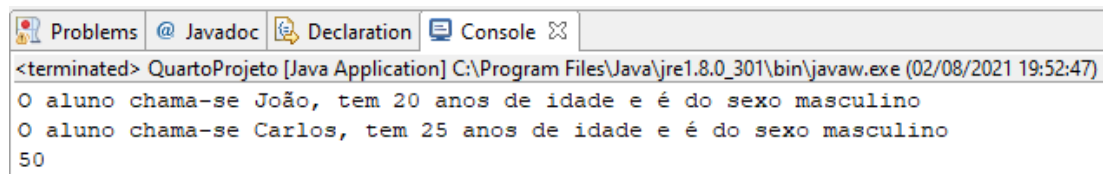
    public void DefineSexo(String s){
        // Define o valor de sexo do aluno
        sexo = s;
    }

}
```

## Matemática.java

```
package com.sys4soft;  
  
public class Matematica {  
  
    public static void Adicao(int parcela1, int parcela2){  
        System.out.println(parcela1 + parcela2);  
    }  
}
```

Rodando (Run):



The screenshot shows the 'Console' tab of a Java IDE. The title bar indicates the application is 'QuartoProjeto [Java Application]' running at 'C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe (02/08/2021 19:52:47)'. The console output displays the results of the 'Adicao' method calls: 'O aluno chama-se João, tem 20 anos de idade e é do sexo masculino' followed by the sum '50', and then 'O aluno chama-se Carlos, tem 25 anos de idade e é do sexo masculino' followed by the sum '50'.

```
<terminated> QuartoProjeto [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (02/08/2021 19:52:47)  
O aluno chama-se João, tem 20 anos de idade e é do sexo masculino  
50  
O aluno chama-se Carlos, tem 25 anos de idade e é do sexo masculino  
50
```

## 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 character = 'c';

String palavra = "texto";

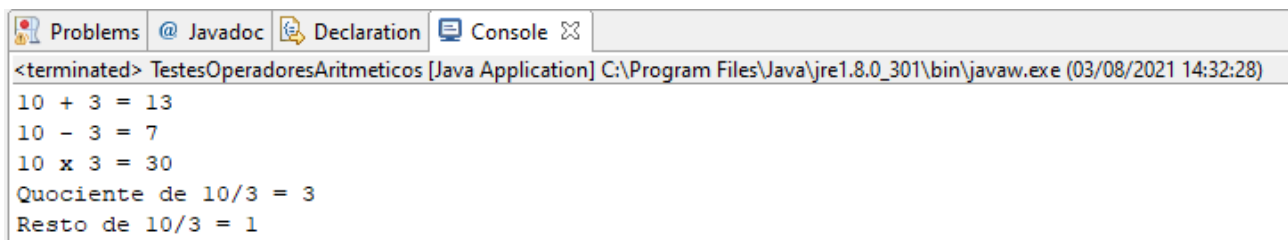
OBS.: String é uma classe e tem vários métodos associados.

## 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);  
  
    }  
}
```



The screenshot shows the IDE's console window with the following output:

```
<terminated> TestesOperadoresAritmeticos [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 14:32:28)  
10 + 3 = 13  
10 - 3 = 7  
10 x 3 = 30  
Quociente de 10/3 = 3  
Resto de 10/3 = 1
```

## 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);
```

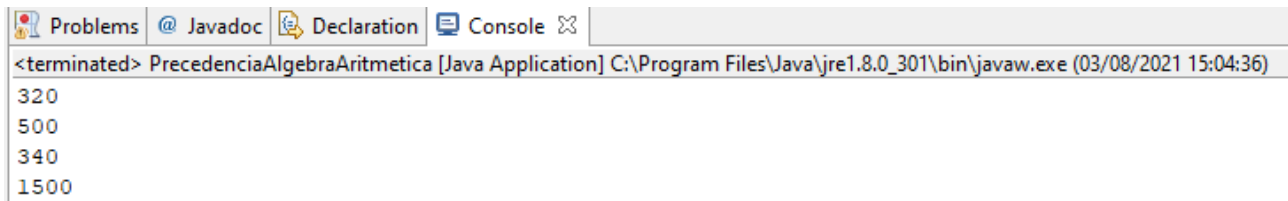
```
        calculo = (20 + 30) * 10;  
        System.out.println(calculo);
```

```
        calculo = 20 + 30 * 10 + 20;  
        System.out.println(calculo);
```

```
        calculo = (20 + 30) * (10 + 20);  
        System.out.println(calculo);
```

```
    }
```

```
}
```



The screenshot shows a Java IDE interface with a console window. The console title bar reads "<terminated> PrecedenciaAlgebraAritmetica [Java Application] C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe (03/08/2021 15:04:36)". The console output displays four lines of numbers: 320, 500, 340, and 1500, which correspond to the results of the arithmetic expressions in the code above.

```
<terminated> PrecedenciaAlgebraAritmetica [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 15:04:36)  
320  
500  
340  
1500
```



## Aula 20 - Mecanismos de decisão - condição If

Condicaolf.java

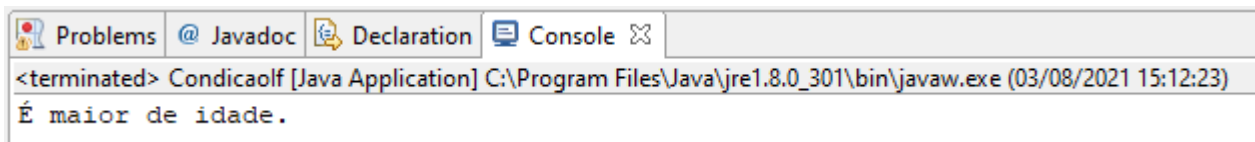
```
package com.sys4soft;

public class Condicaolf {

    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;
    }
}
```

 Problems	 Javadoc	 Declaration	 Console 
--	---	---	---

```
<terminated> CondicaoIfElse [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 15:52:51)  
15 anos: Não pode votar!  
17 anos: O voto é opcional!  
30 anos: O voto é obrigatório!  
70 anos: O voto é opcional!  
Idade muito alta para votar
```

## Aula 25 - Operador condicional / operador ternário

### TestesOperadorTernario.java

```
package com.sys4soft;

public class TestesOperadorTernario {

    public static void main(String[] args) {

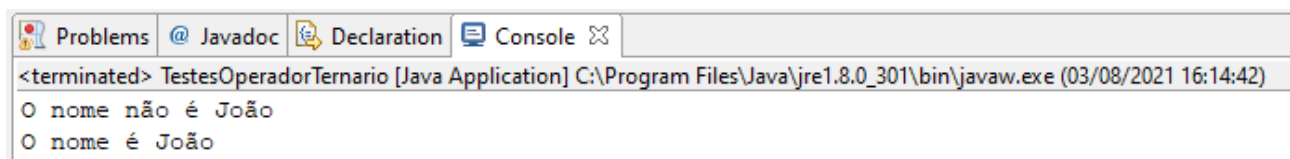
        String nome = "";

        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);
    }
}
```

```
<terminated> TestesOperadoresAtribuicao [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 19:01:38)
60
30
90
18
4
```

## Aula 27 - Noção de ciclo e ciclos While

### TestesWhile.java

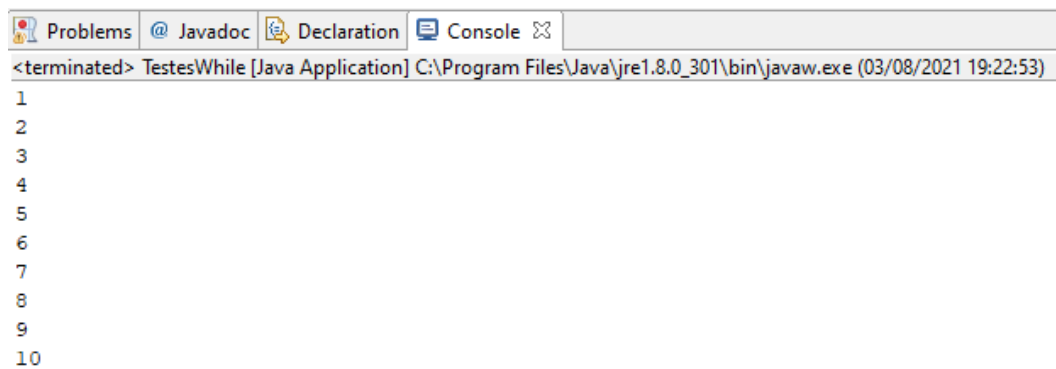
```
package com.sys4soft;

public class TestesWhile {

    public static void main(String[] args) {

        int valor = 1;

        while(valor <= 10){
            System.out.println(valor);
            valor++;
        }
    }
}
```



```
<terminated> TestesWhile [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 19:22:53)
1
2
3
4
5
6
7
8
9
10
```

## Aula 28 - Mais detalhes sobre o ciclo while

### InterrompendoCicloWhile.java

```
package com.sys4soft;

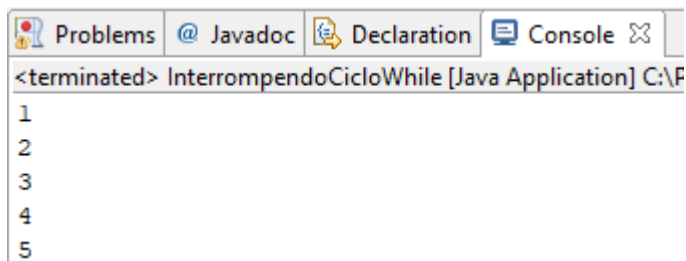
public class InterrompendoCicloWhile {

    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;

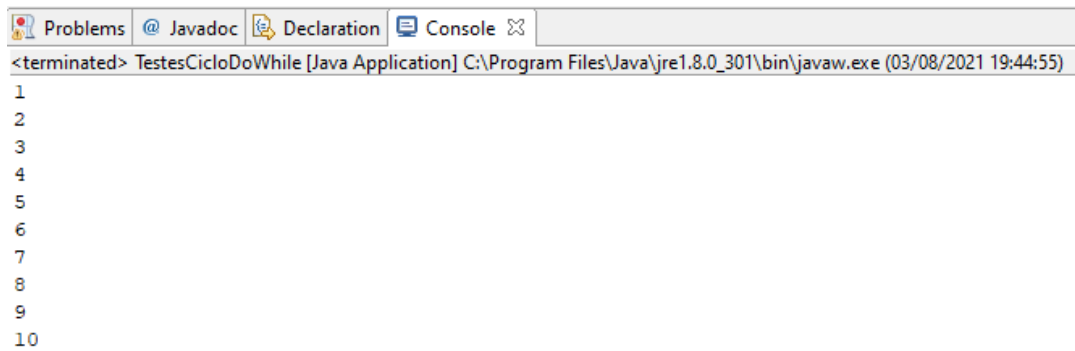
public class TestesCicloDoWhile {

    public static void main(String[] args) {

        int valor = 1;

        do {
            System.out.println(valor);
            valor++;
        } while(valor <= 10);

    }
}
```



The screenshot shows an IDE console window with the following tabs: Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the Java application. The output consists of the numbers 1 through 10, each on a new line, indicating that the do-while loop executed successfully for 10 iterations.

```
<terminated> TestesCicloDoWhile [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 19:44:55)
1
2
3
4
5
6
7
8
9
10
```



## Aula 30 - Ciclo For

### TestesCicloFor.java

```
package com.sys4soft;

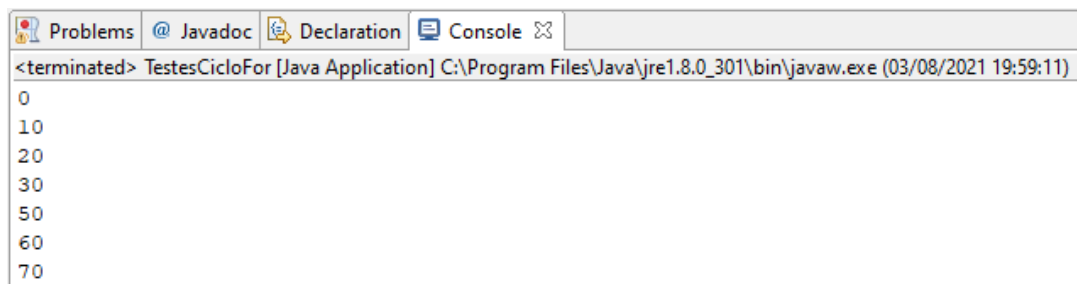
public class TestesCicloFor {

    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);
        }
    }
}
```



The screenshot shows an IDE window with a tab labeled "Console". The console output displays the numbers 0, 10, 20, 30, 50, 60, and 70, each on a new line. The output is preceded by the text "<terminated> TestesCicloFor [Java Application] C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe (03/08/2021 19:59:11)".

```
<terminated> TestesCicloFor [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 19:59:11)
0
10
20
30
50
60
70
```

## Aula 31 - Estrutura condicional Switch

EstruturaCondicionalSwitch.java

```
package com.sys4soft;

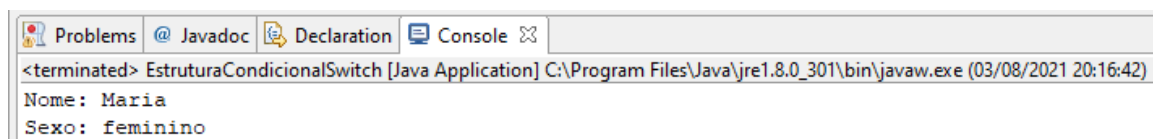
public class EstruturaCondicionalSwitch {

    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);
    }
}
```



The screenshot shows an IDE window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the Java application. The output is as follows:

```
<terminated> EstruturaCondicionalSwitch [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 20:16:42)
Nome: Maria
Sexo: feminino
```

## Aula 32 - Utilização do operador lógico de negação

**TestesOperadorLogicoNegacao.java**

```
package com.sys4soft;

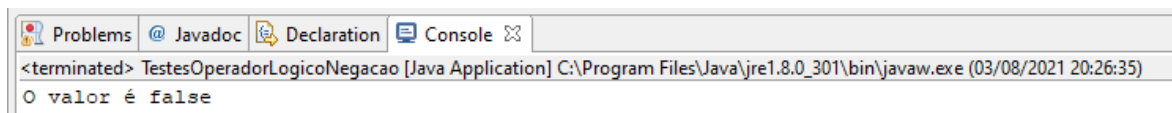
public class TestesOperadorLogicoNegacao {

    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("=====");

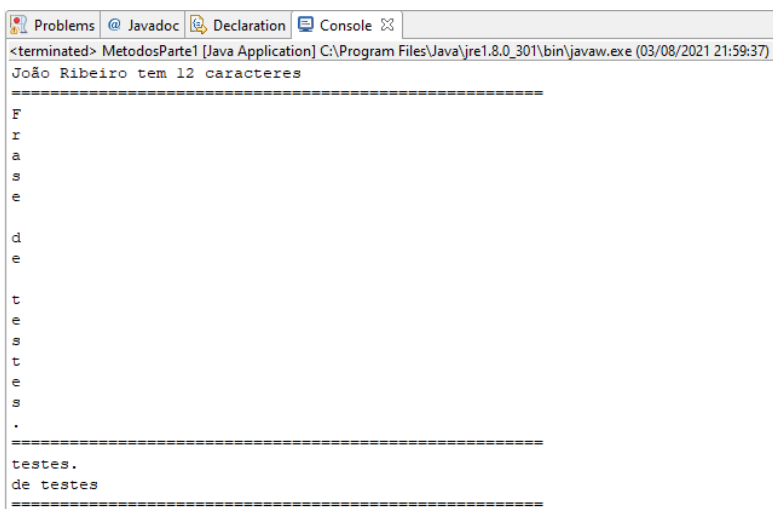
        String teste_substring1 = texto.substring(9);
        String teste_substring2 = texto.substring(6, 15);

        System.out.println(teste_substring1);
        System.out.println(teste_substring2);

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

    }

}
```



```
<terminated> MetodosParte1 [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 21:59:37)
João Ribeiro tem 12 caracteres
=====
F
r
a
s
e

d
e

t
e
s
t
e
s
.
=====
testes.
de testes
=====
```

## 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');

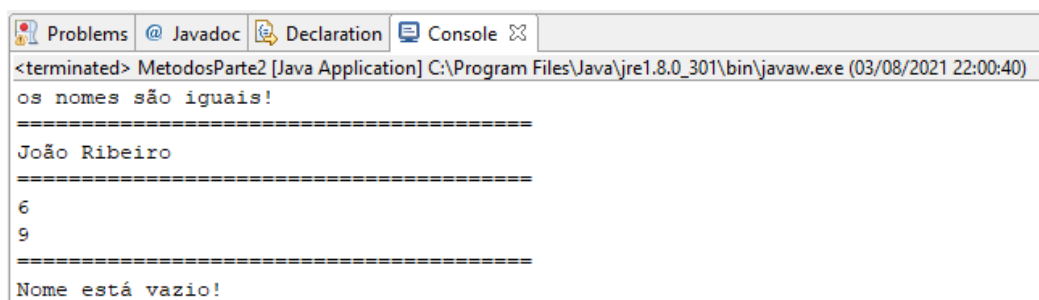
        System.out.println(posicao1);
        System.out.println(posicao2);

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

        if(nome3.isEmpty()){
            System.out.println("Nome está vazio!");
        }

    }

}
```



```
<terminated> MetodosParte2 [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 22:00:40)
os nomes são iguais!
=====
João Ribeiro
=====
6
9
=====
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("=====");

    }
}
```

```
Problems @ Javadoc Declaration Console
<terminated> MetodosParte3 [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 22:03:27)
Frase de testes de strings no Java ==> CONTÉM a string testes
Frase de testes de strings no Java ==> NÃO CONTÉM a string java
frase de testes de strings no java ==> CONTÉM a string java
FRASE DE TESTES DE STRINGS NO JAVA ==> CONTÉM a string JAVA
=====
Frase de testes de strings no java
=====
Textes com strings
=====
```

## 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");
```

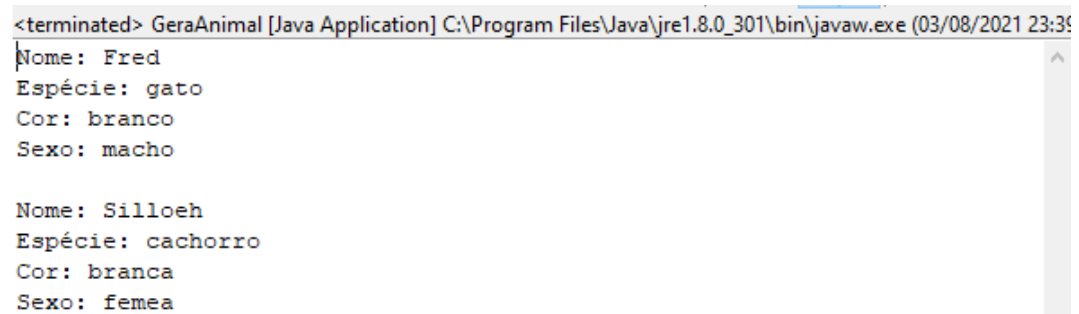
```
    }
```

```
}
```



## Animal.java

```
package com.sys4soft;  
  
public class Animal {  
    public String nome;  
    public String especie;  
    public String cor;  
    public String sexo;  
}
```



```
<terminated> GeraAnimal [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 23:39)  
Nome: Fred  
Espécie: gato  
Cor: branco  
Sexo: macho  
  
Nome: Silloeh  
Espécie: cachorro  
Cor: branca  
Sexo: femea
```

## 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!");  
    }  
  
}
```

```
<terminated> GeraAnimal [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 23:44  
Nome: Fred  
Espécie: gato  
Cor: branco  
Sexo: macho  
  
Nome: Silloeh  
Espécie: cachorro  
Cor: branca  
Sexo: femea  
  
O gato mia  
O cachorro late  
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();

    }

}
```

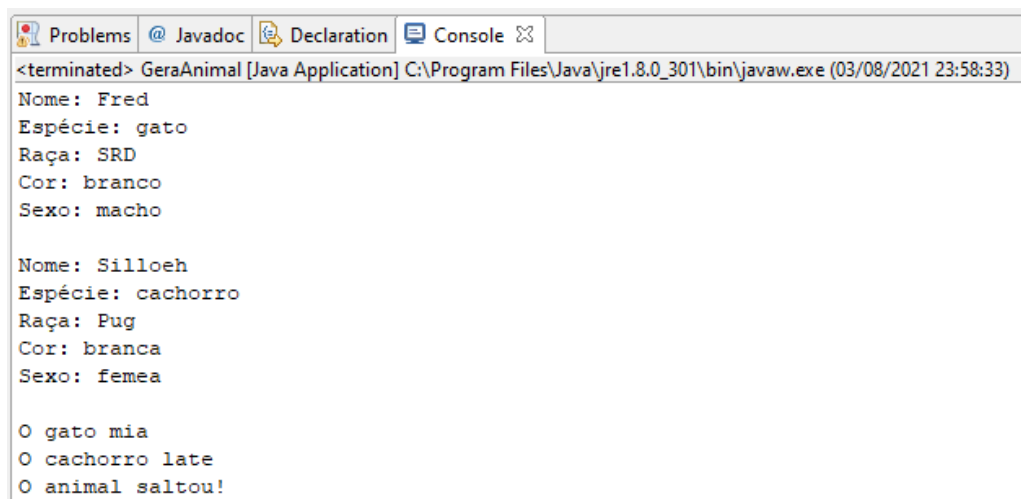
## Animal.java

```
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;
    }
}
```



The screenshot shows a Java IDE window with the 'Console' tab selected. The title bar indicates the application is 'GeraAnimal [Java Application]' running on 'C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe' at '03/08/2021 23:58:33'. The console output displays the results of two animal objects and three actions:

```
<terminated> GeraAnimal [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (03/08/2021 23:58:33)
Nome: Fred
Espécie: gato
Raça: SRD
Cor: branco
Sexo: macho

Nome: Silloeh
Espécie: cachorro
Raça: Pug
Cor: branca
Sexo: femea

O gato mia
O cachorro late
O animal saltou!
```

## 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

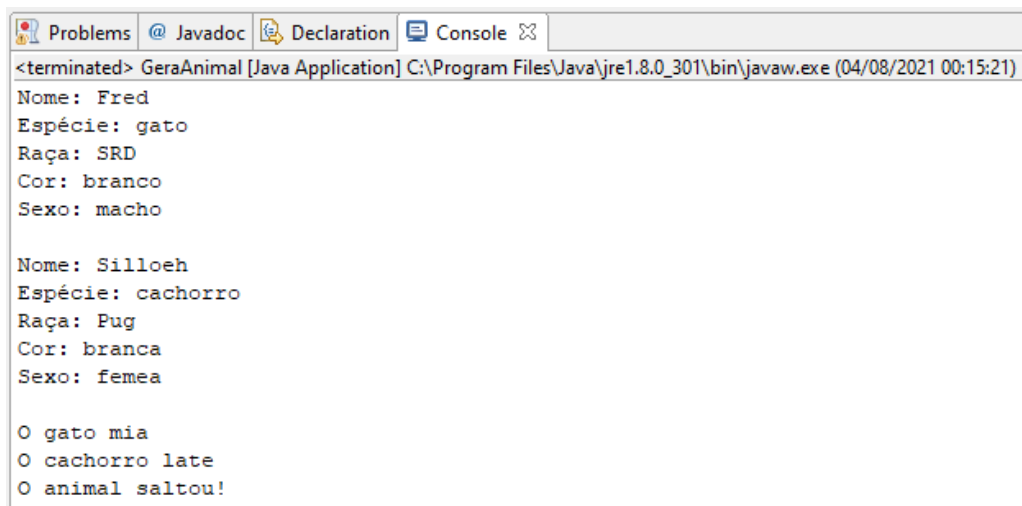
```
package com.sys4soft;

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;
    }
}
```



```
<terminated> GeraAnimal [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (04/08/2021 00:15:21)
Nome: Fred
Espécie: gato
Raça: SRD
Cor: branco
Sexo: macho

Nome: Silloeh
Espécie: cachorro
Raça: Pug
Cor: branca
Sexo: femea

O gato mia
O cachorro late
O animal saltou!
```

## 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

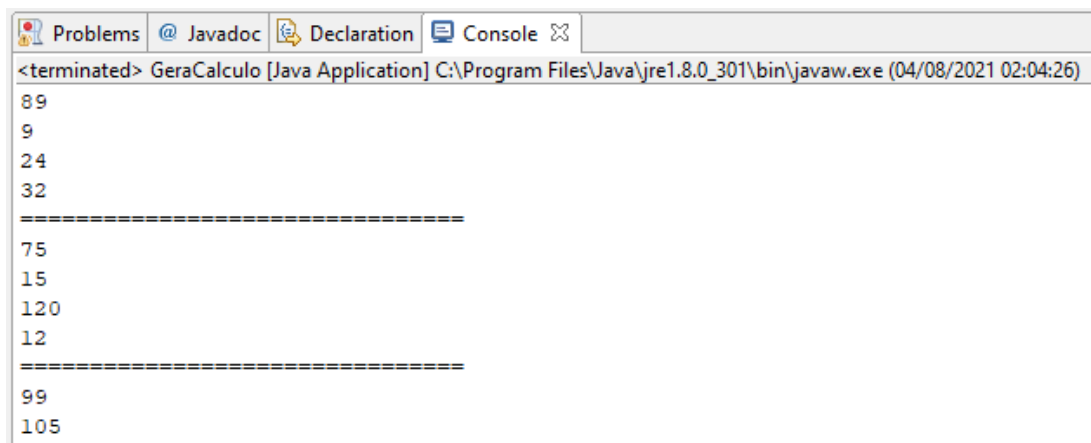
```
package com.sys4soft;

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);
    }

}
```



```
<terminated> GeraCalculo [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (04/08/2021 02:04:26)
89
9
24
32
=====
75
15
120
12
=====
99
105
```

## 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");

    }
}
```

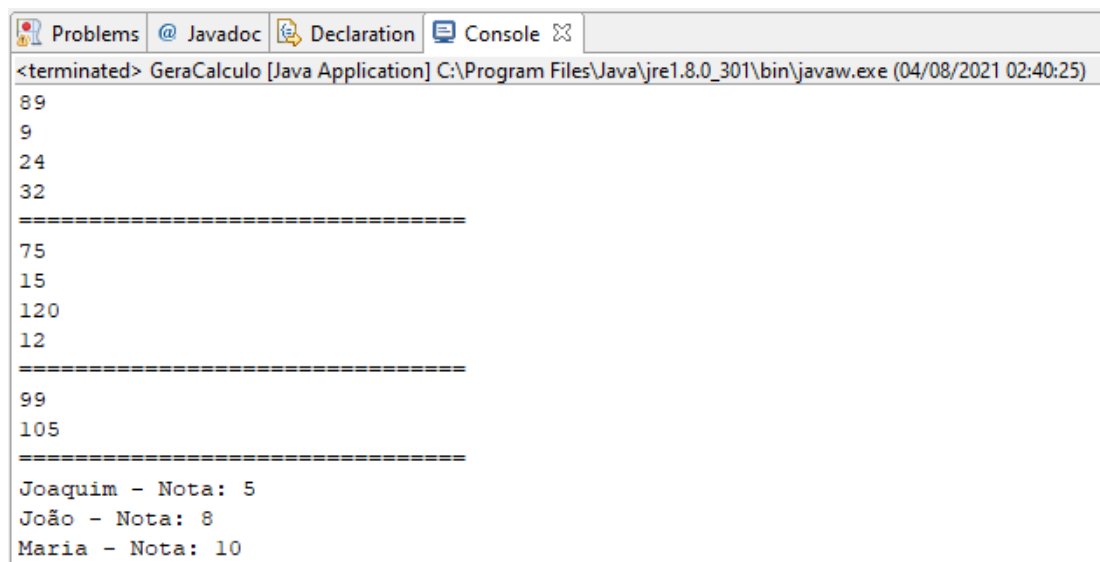
## PassagemParametros.java

```
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);
    }
}
```



```
<terminated> GeraCalculo [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (04/08/2021 02:40:25)
89
9
24
32
=====
75
15
120
12
=====
99
105
=====
Joaquim - Nota: 5
João - Nota: 8
Maria - Nota: 10
```

## 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();
    }
}
```

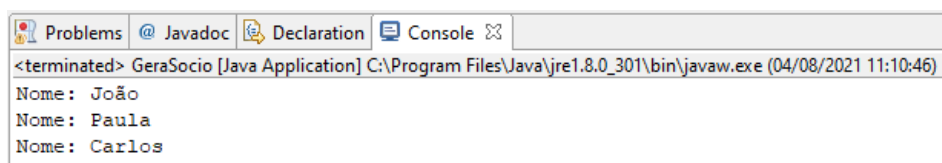
### Socio.java

```
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);
    }
}
```



The screenshot shows the 'Console' tab of an IDE. The title bar indicates the application is 'GeraSocio [Java Application]' running at 'C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe' on '04/08/2021 11:10:46'. The output text in the console is:

```
<terminated> GeraSocio [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (04/08/2021 11:10:46)
Nome: João
Nome: Paula
Nome: Carlos
```

## 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);
    }
}
```

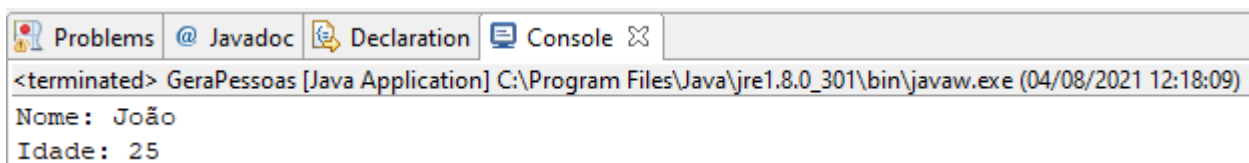
#### Pessoas.java

```
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);
    }
}
```



The screenshot shows the IDE's console window with tabs for Problems, Javadoc, Declaration, and Console. The console output indicates the application has terminated successfully and displays the results of the static method calls: 'Nome: João' and 'Idade: 25'.

```
<terminated> GeraPessoas [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (04/08/2021 12:18:09)
Nome: João
Idade: 25
```

## 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);
    }
}
```

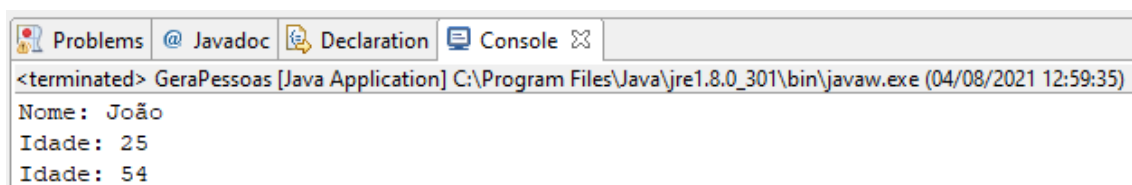
### Pessoas.java

```
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);
    }

}
```



```
Problems @ Javadoc Declaration Console 
<terminated> GeraSocio [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (04/08/2021 13:29:44)
Nome: João
Nome: Paula
Nome: Carlos
=====
Construtor com parâmetro String Carlos

Construtor com parâmetro String Roberto
Construtor sem parâmetros
```

## 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_batom = true;

        h1.ApresentarHumano();
        h1.info_barba(h1.tem_barba);
        m1.ApresentarHumano();
        m1.info_batom(m1.usa_batom);

    }

}
```

### 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;

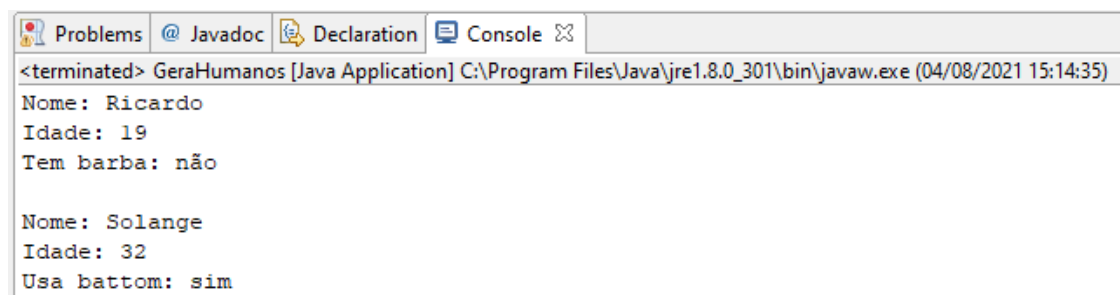
public class Mulher extends Humano {
    boolean usa_batom;

    public void info_batom(boolean usa_batom){

        String resposta;

        if(usa_batom){
            resposta = "sim";
        } else {
            resposta = "não";
        }

        System.out.println("Usa batom: " + resposta + "\n");
    }
}
```



```
<terminated> GeraHumanos [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (04/08/2021 15:14:35)
Nome: Ricardo
Idade: 19
Tem barba: não

Nome: Solange
Idade: 32
Usa batom: sim
```

## 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");
    }

}
```

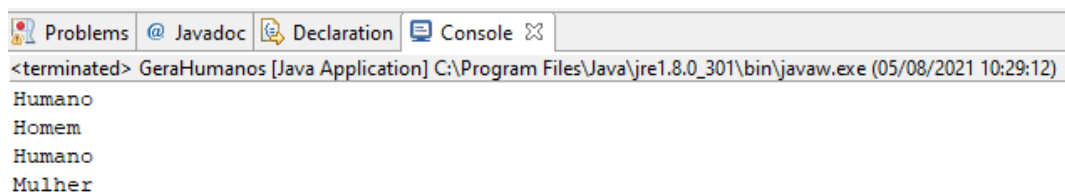
### Mulher.java

```
package com.sys4soft;

public class Mulher extends Humano {
    boolean usa_batom;

    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

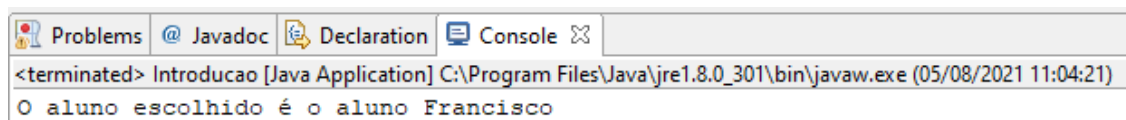
        String[] alunos = new String[6];

        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 {

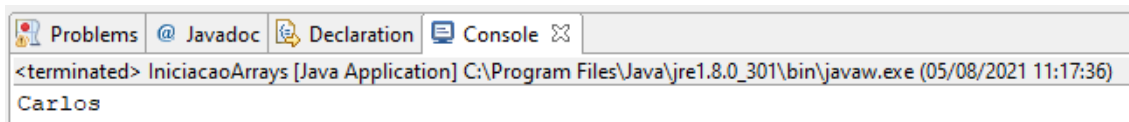
    public static void main(String[] args) {

        // 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"
        };

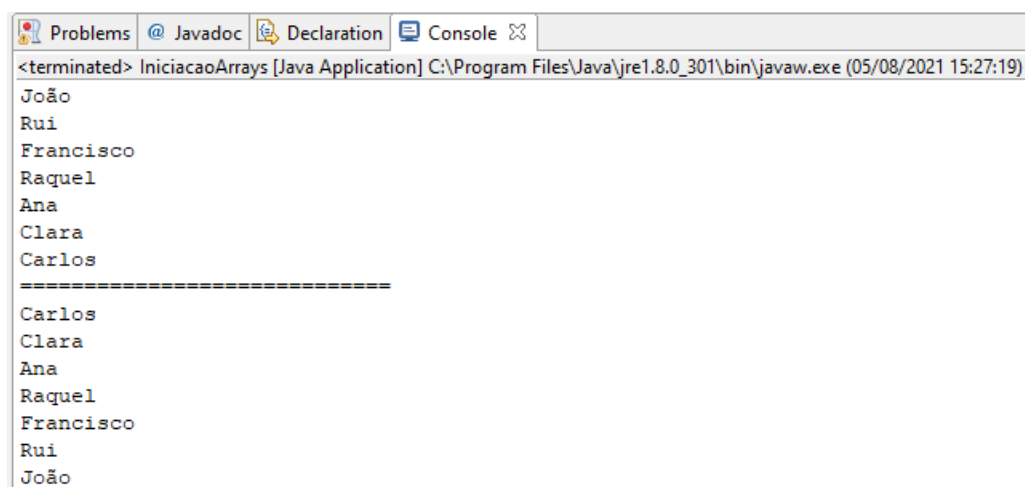
        for(int i = 0; i < alunos.length; i++){
            System.out.println(alunos[i]);
        }

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

        for(int i = alunos.length-1; i >= 0; i--){
            System.out.println(alunos[i]);
        }

    }

}
```



```
<terminated> IniciacaoArrays [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (05/08/2021 15:27:19)
João
Rui
Francisco
Raquel
Ana
Clara
Carlos
=====
Carlos
Clara
Ana
Raquel
Francisco
Rui
João
```



## 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"
        };

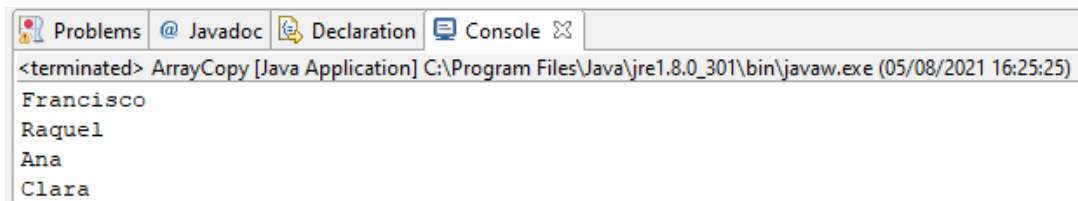
        String[] novo = new String[4];

        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);
        };

    }

}
```

```
Problems @ Javadoc Declaration Console
<terminated> OrdenandoDados [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (06/08/2021 09:47:35)
Ana
Carlos
Clara
Francisco
João
Raquel
Rui
=====
23
42
45
67
68
87
=====
Rui
Raquel
João
Francisco
Clara
Carlos
Ana
```

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

### ReversaoArrayNumerico.java

```
package com.sys4soft;

public class ReversaoArrayNumerico {

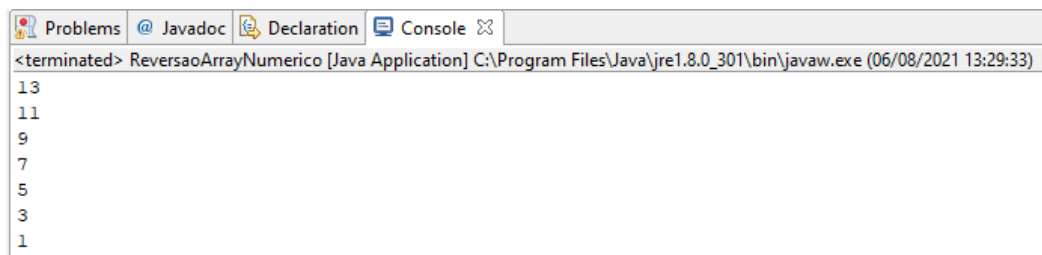
    public static void main(String[] args) {

        // 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]);
        }

    }

}
```



The screenshot shows a Java IDE window with the 'Console' tab selected. The title bar indicates the application is 'ReversaoArrayNumerico [Java Application]' running at 'C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe' on '06/08/2021 13:29:33'. The console output displays the numbers 13, 11, 9, 7, 5, 3, and 1, each on a new line, representing the reversed array.

```
<terminated> ReversaoArrayNumerico [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (06/08/2021 13:29:33)
13
11
9
7
5
3
1
```

## 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){

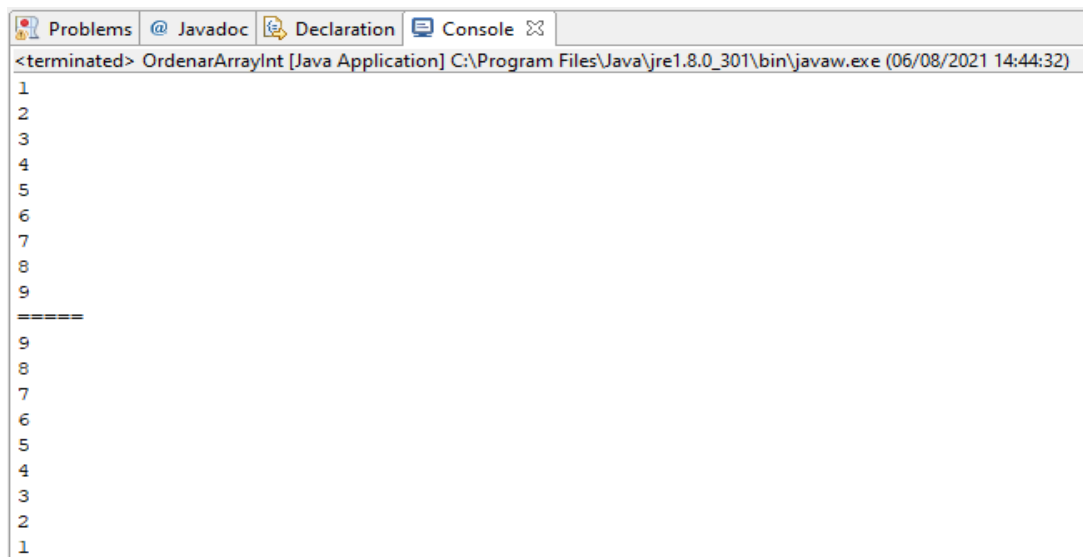
        Arrays.sort(array_int);

        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;
    }
}
```



```
<terminated> OrdenarArrayInt [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (06/08/2021 14:44:32)
1
2
3
4
5
6
7
8
9
=====
9
8
7
6
5
4
3
2
1
```

## Aula 57 - Exemplo de um Array multidimensional

### ArrayMultidimensional.java

```
package com.sys4soft;

public class ArrayMultidimensional {

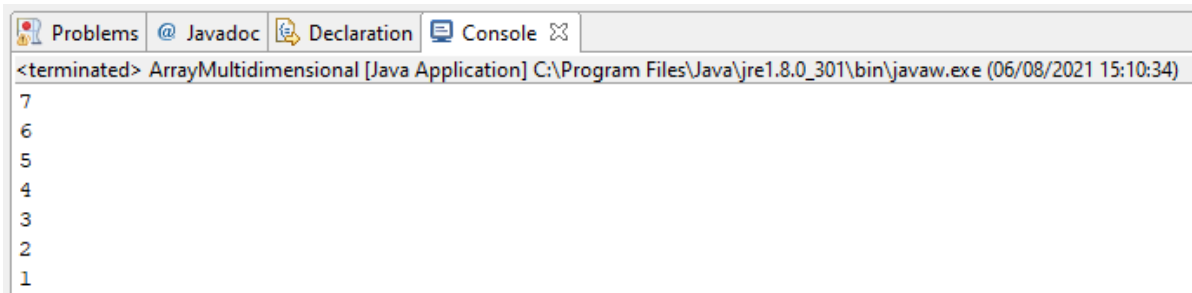
    public static void main(String[] args) {

        // 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]);
        }

    }

}
```



```
<terminated> ArrayMultidimensional [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (06/08/2021 15:10:34)
7
6
5
4
3
2
1
```

## 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

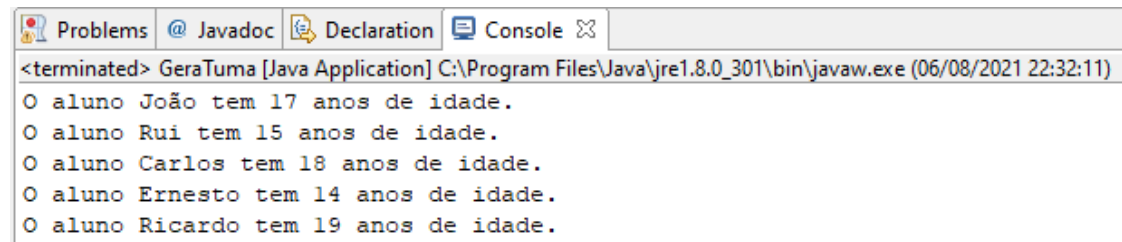
```
package com.sys4soft;
```

```
public class Aluno {
```

```
    String nome;
```

```
    int idade;
```

```
}
```



The screenshot shows an IDE console window with the following tabs: Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of a Java application. The output consists of five lines, each starting with "O aluno" followed by a name and age. The application has terminated, as indicated by the "<terminated>" message at the top of the console output.

```
<terminated> GeraTuma [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (06/08/2021 22:32:11)  
O aluno João tem 17 anos de idade.  
O aluno Rui tem 15 anos de idade.  
O aluno Carlos tem 18 anos de idade.  
O aluno Ernesto tem 14 anos de idade.  
O aluno Ricardo tem 19 anos de 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!");
```

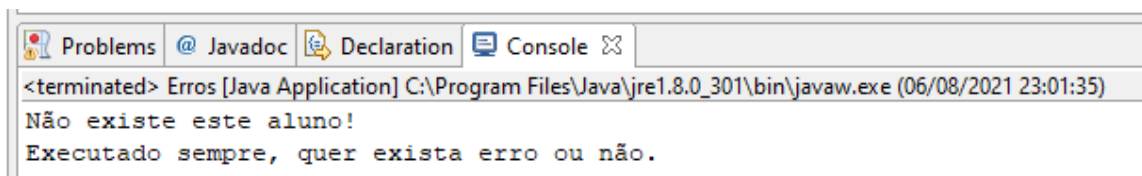
```
        } finally {
```

```
            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);

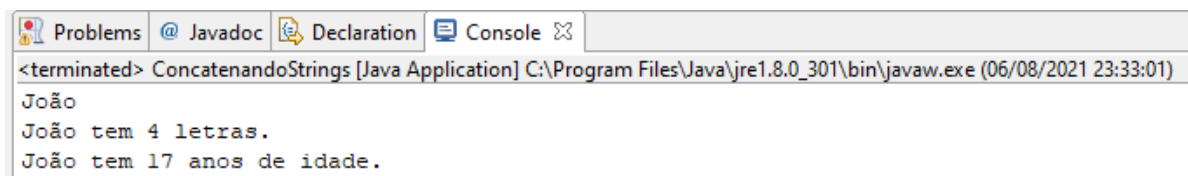
        System.out.println(nome);
        System.out.println(nome + " tem " + nome.length() + " letras.");

        StringBuilder str = new StringBuilder();
        str.append("João");
        str.append(" tem ");
        str.append(17);
        str.append(" anos de idade.");

        System.out.println(str);

    }

}
```



The screenshot shows an IDE window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the Java application. The output consists of three lines: "João", "João tem 4 letras.", and "João tem 17 anos de idade." The window title bar indicates the application is "ConcatenandoStrings [Java Application]" and the command used is "C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe (06/08/2021 23:33:01)".

```
<terminated> ConcatenandoStrings [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (06/08/2021 23:33:01)
João
João tem 4 letras.
João tem 17 anos de idade.
```

## 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) {

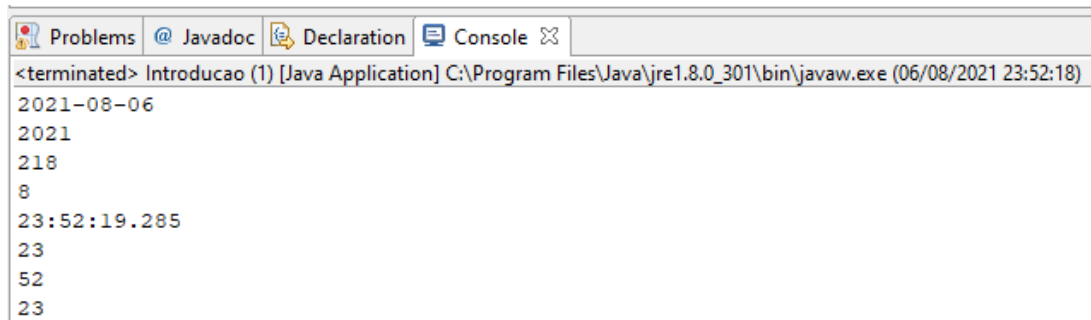
        System.out.println(LocalDate.now().toString());
        System.out.println(LocalDate.now().getYear());
        System.out.println(LocalDate.now().getDayOfYear());
        System.out.println(LocalDate.now().getMonthValue());

        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);

    }

}
```



The screenshot shows an IDE console window with the following tabs: Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the Java application. The output consists of seven lines: the full date and time in ISO format, the year, the day of the year, the month value, the full time in ISO format, the hour, and the minute. The output matches the code provided above.

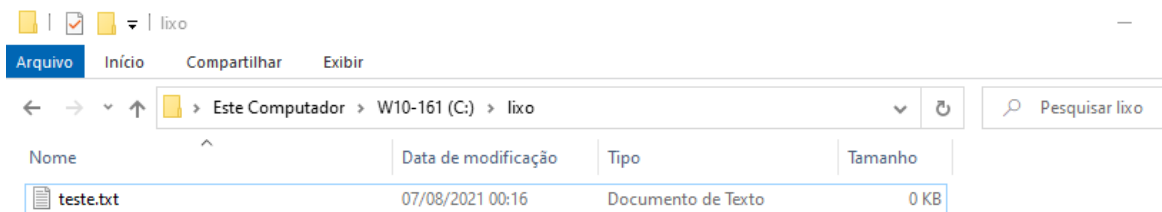
```
<terminated> Introducao (1) [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (06/08/2021 23:52:18)
2021-08-06
2021
218
8
23:52:19.285
23
52
23
```

# 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) {

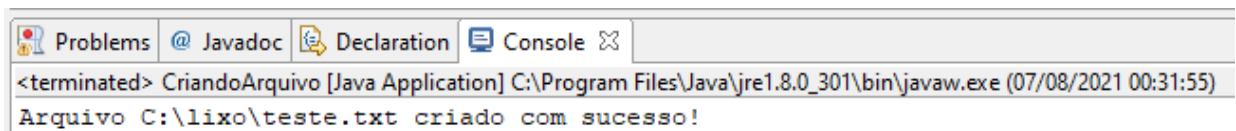
        // String nomeArquivo = "C:\\\\lixo\\\\teste.txt";
        String nomeArquivo = "C:" + File.separator + "lixo" + File.separator + "teste.txt";

        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) {

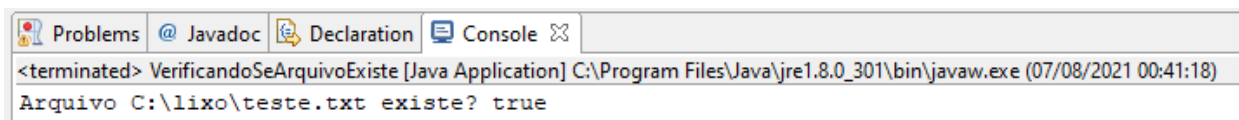
        String nomeArquivo = "C:" + File.separator + "lixo" + File.separator + "teste.txt";

        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 {

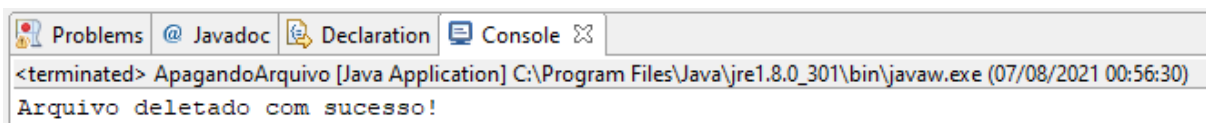
    public static void main(String[] args) {

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

        File arquivo = new File(nomeArquivo);

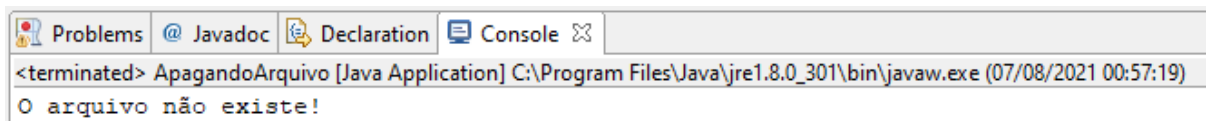
        boolean existe = arquivo.exists();

        if(existe){
            arquivo.delete();
            System.out.println("Arquivo deletado com sucesso!");
        } else {
            System.out.println("O arquivo não existe!");
        }
    }
}
```



```
Problems | @ Javadoc | Declaration | Console |
<terminated> ApagandoArquivo [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (07/08/2021 00:56:30)
Arquivo deletado com sucesso!
```

- Rodando novamente:



```
Problems | @ Javadoc | Declaration | Console |
<terminated> ApagandoArquivo [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (07/08/2021 00:57:19)
O arquivo não existe!
```

## 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) {

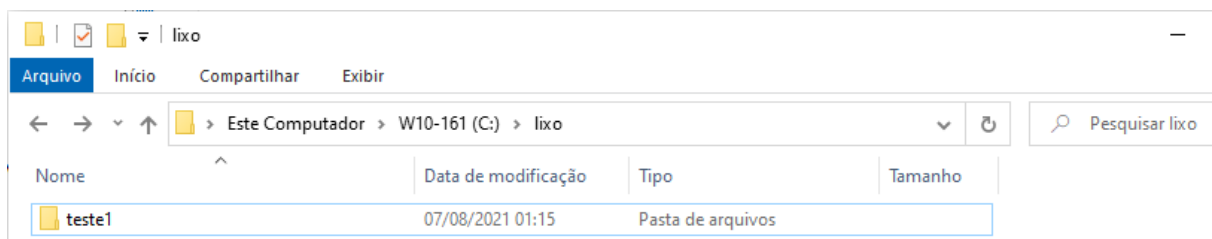
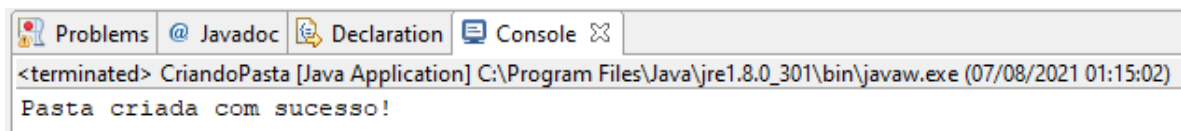
        String nomePasta = "C:" + File.separator + "lixo" + File.separator + "teste1";

        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) {

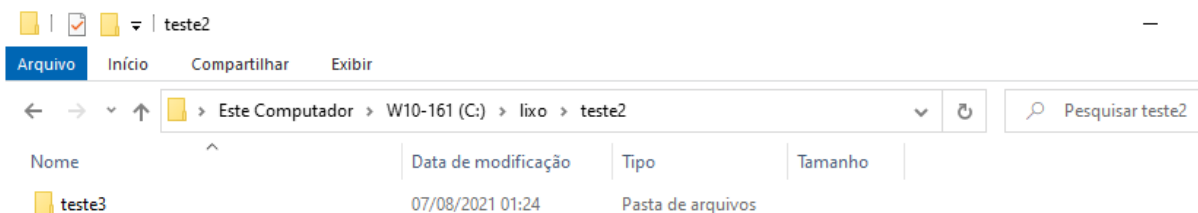
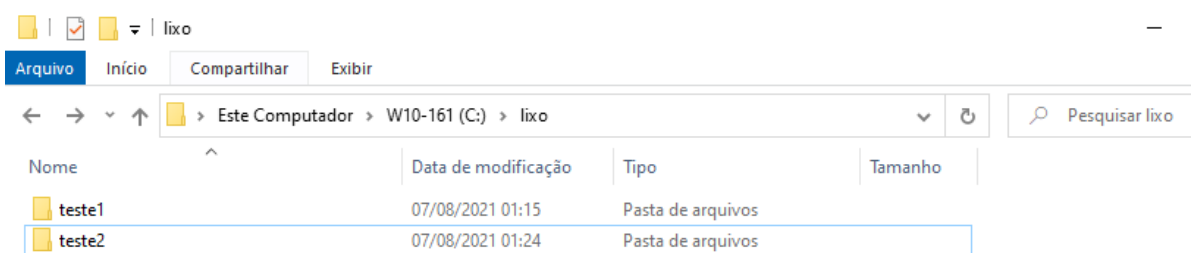
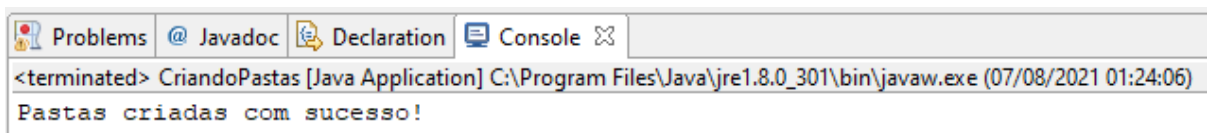
        String nomePasta = "C:" + File.separator + "lixo" + File.separator + "teste2" + File.separator + "teste3";

        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!");
        }

    }

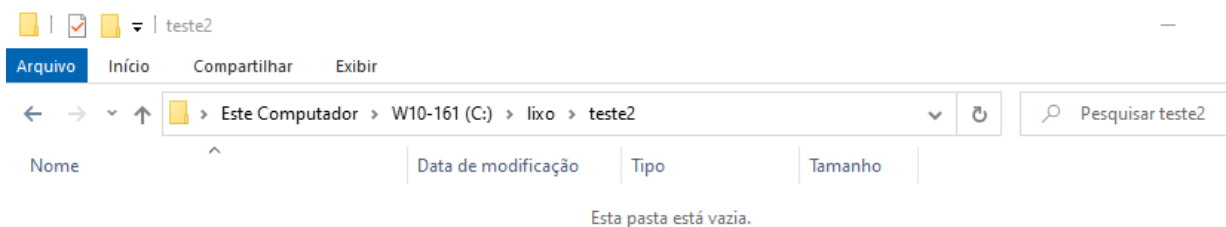
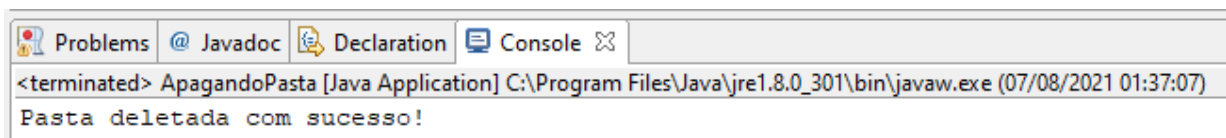
}
```



## 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);
```

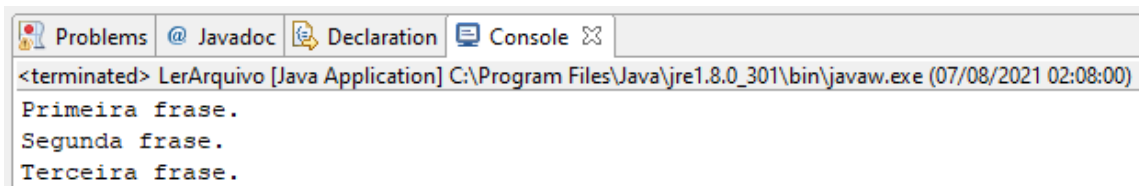
```
        } catch(Exception e) {
```

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

```
        }
```

```
    }
```

```
}
```



The screenshot shows an IDE window with a tab labeled 'Console'. The console output displays the text from the file 'teste.txt' as follows:

```
<terminated> LerArquivo [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (07/08/2021 02:08:00)
Primeira frase.
Segunda frase.
Terceira frase.
```

## Aula 69 - Escrever conteúdos num arquivos de texto

### InserirTexto.java

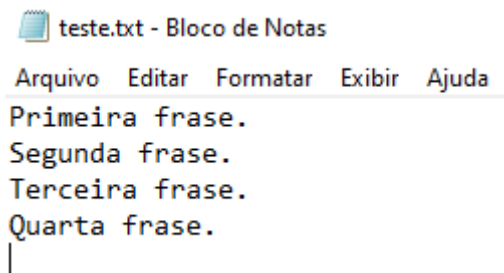
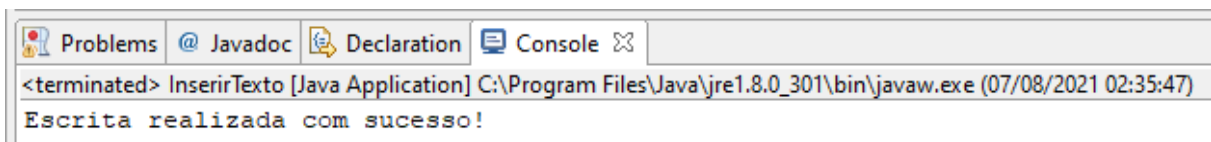
```
package com.sys4soft;

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);
        }
    }
}
```



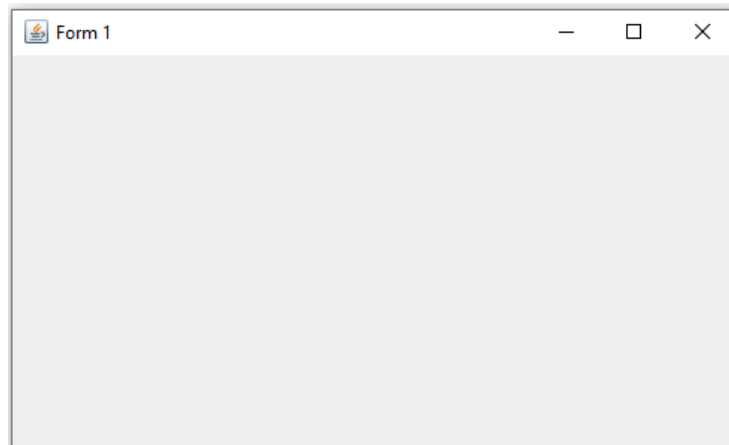
## Aula 70 - Introdução a GUI com classes Swing

### 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");

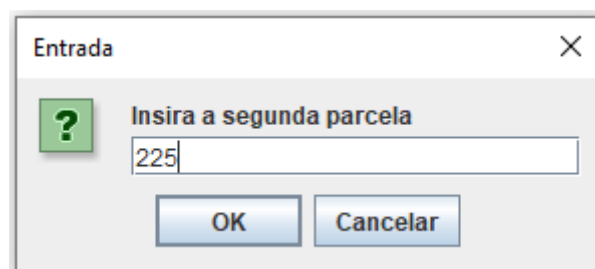
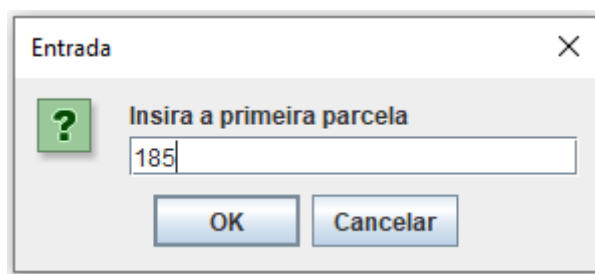
        // converter as variáveis string em int
        int parcela1 = Integer.parseInt(primeira_parcela);
        int parcela2 = Integer.parseInt(segunda_parcela);

        int resultado = parcela1 + parcela2;

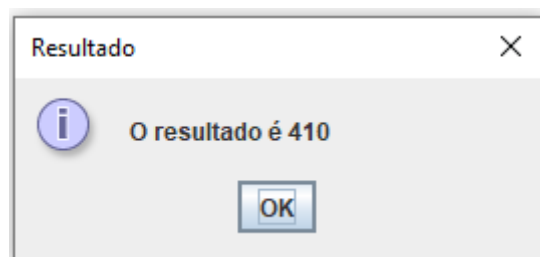
        JOptionPane.showMessageDialog(null,
                                    "O resultado é " + resultado, "Resultado",
                                    JOptionPane.INFORMATION_MESSAGE
                                    );

    }

}
```









## 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;

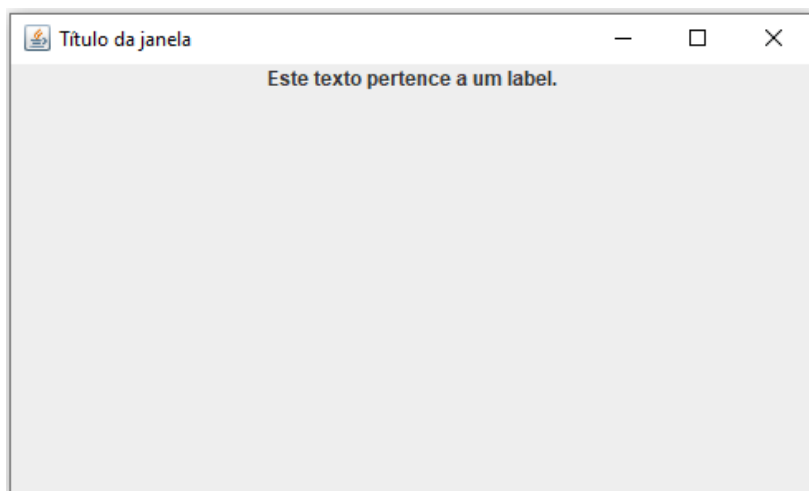
public class SwingJLabel {

    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);

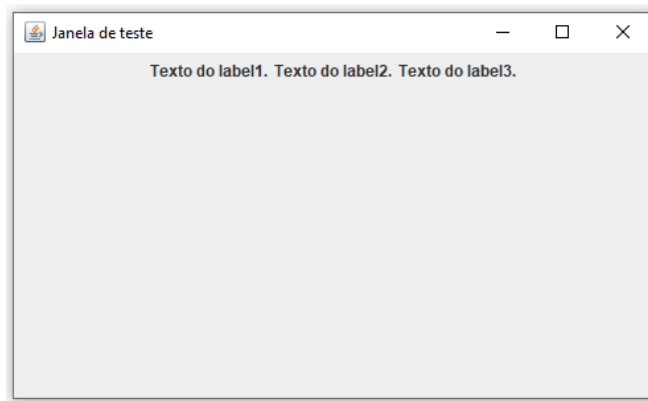
        // label1
        JLabel label1 = new JLabel("Texto do label1.");

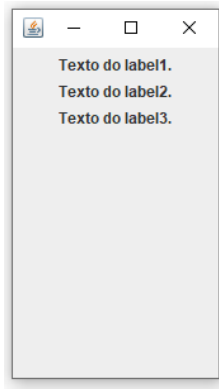
        // label2
        JLabel label2 = new JLabel("Texto do label2.");

        // 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);

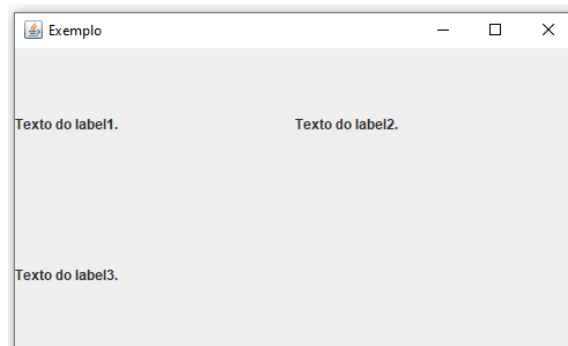
        // label2
        JLabel label2 = new JLabel("Texto do label2.");

        // 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>

The screenshot shows the Oracle Java Documentation website. The browser's address bar displays the URL: `docs.oracle.com/javase/tutorial/uiswing/layout/visual.html`. The page header includes the Oracle logo and the text "The Java™ Tutorials". A search bar is located in the top right corner. The main content area is titled "A Visual Guide to Layout Managers" and includes a list of layout managers: BorderLayout, BoxLayout, CardLayout, FlowLayout, GridBagLayout, GridLayout, GroupLayout, and SpringLayout. A sidebar on the left contains a navigation menu with links to various layout manager topics. A note at the top of the main content area states: "The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases and might use technology no longer available. See Java Language Changes for a summary of updated language features in Java SE 9 and subsequent releases. See JDK Release Notes for information about new features, enhancements, and removed or deprecated options for all JDK releases."

## 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");
```

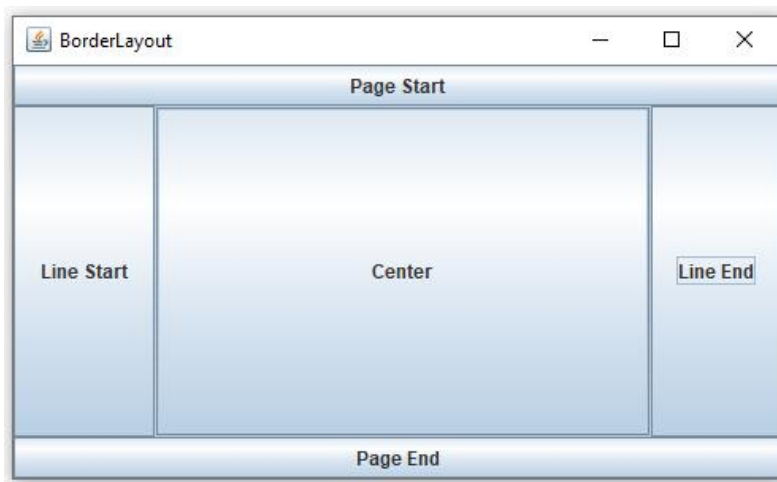
```
// layout de janela
janela.setLayout(new BorderLayout());

// 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);

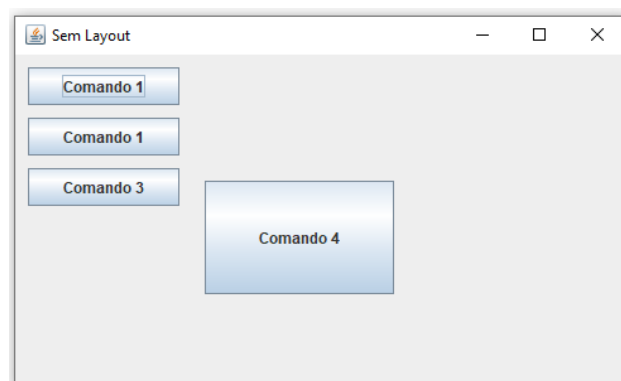
        // comandos
        JButton cmd1 = new JButton("Comando 1");
        cmd1.setBounds(10, 10, 120, 30);

        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) {

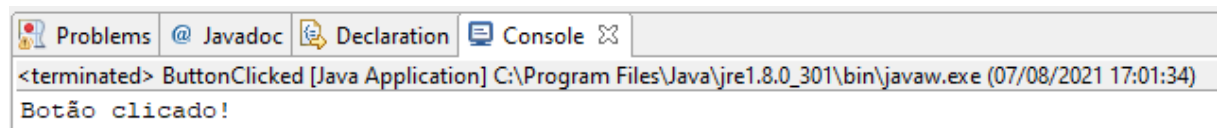
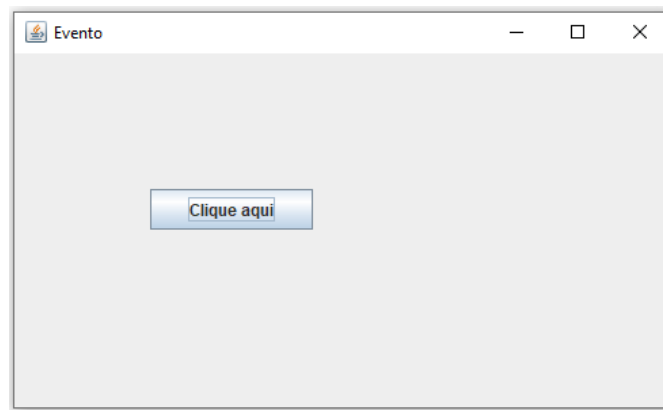
        // janela
        JFrame janela = new JFrame("Evento");
        janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        janela.setSize(500,300);
        janela.setLayout(null);

        // 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

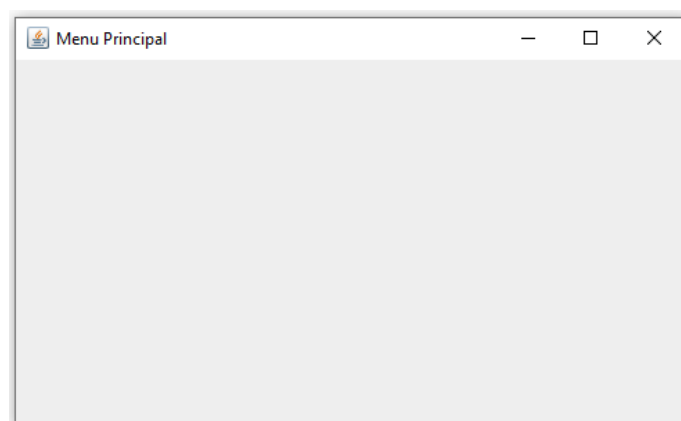
```
package com.sys4soft;

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;

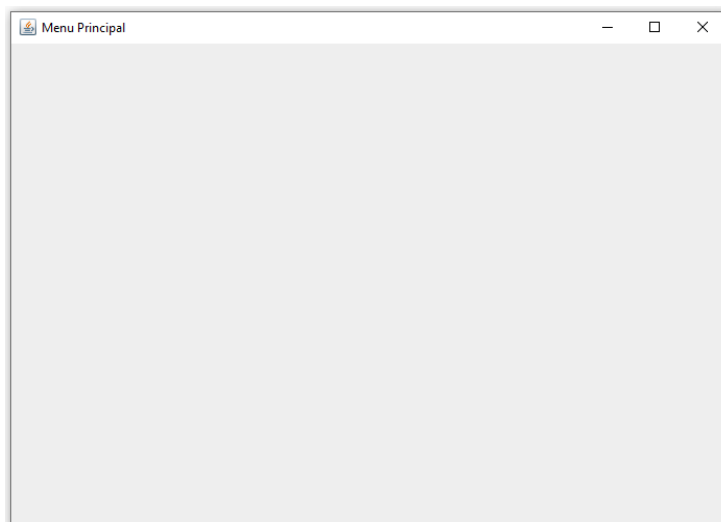
import java.awt.Dimension;

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

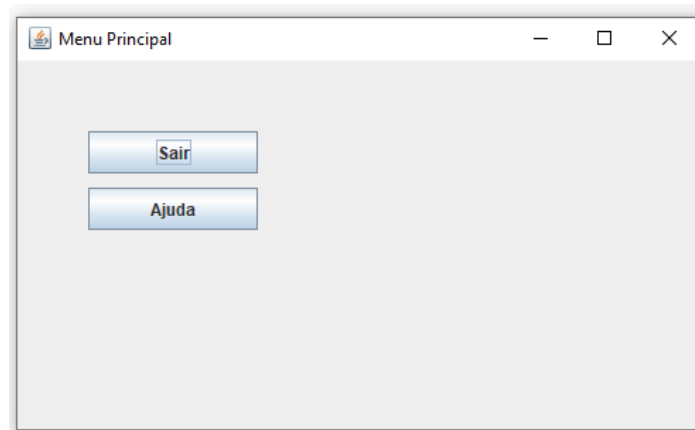
// cmd_sair_click

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_monitor.width/2,
            dimensao_monitor.height/2 - dimensao_monitor.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();

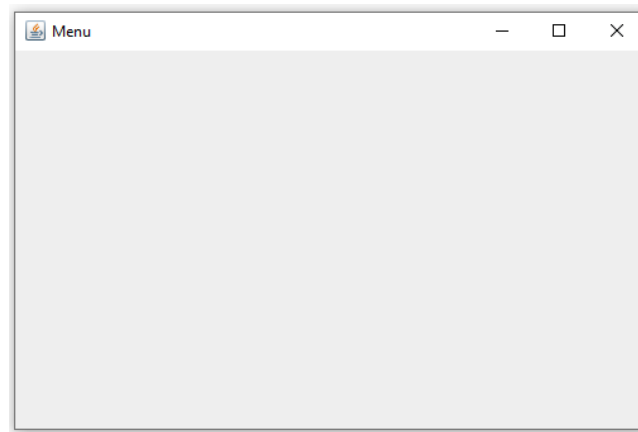
        // centralizar formulário
        setLocationRelativeTo(null);
        setVisible(true);

        // 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);
    }

}
```



Problems	@ Javadoc	Declaration	Console
SwingJFrame2 [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (07/08/2021 21:59:10)			
java.awt.Dimension[width=516,height=339]			
Dimensão exterior = 516 x 339			
Dimensão interior = 500 x 300			

## 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();

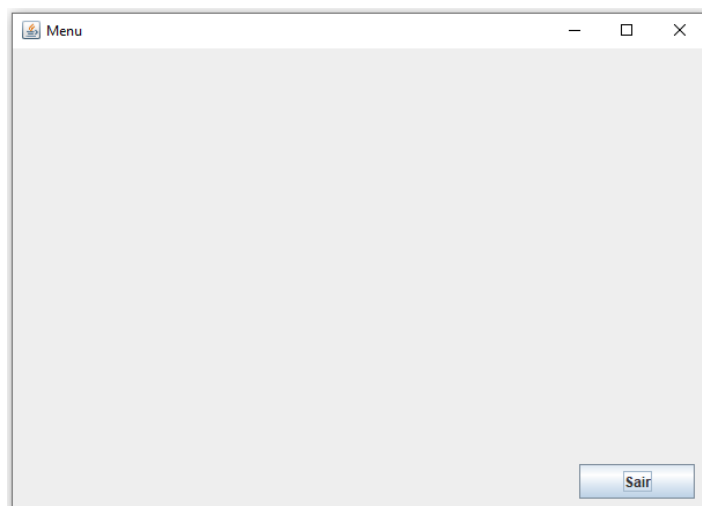
        // centralizar formulário
        setLocationRelativeTo(null);

        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();

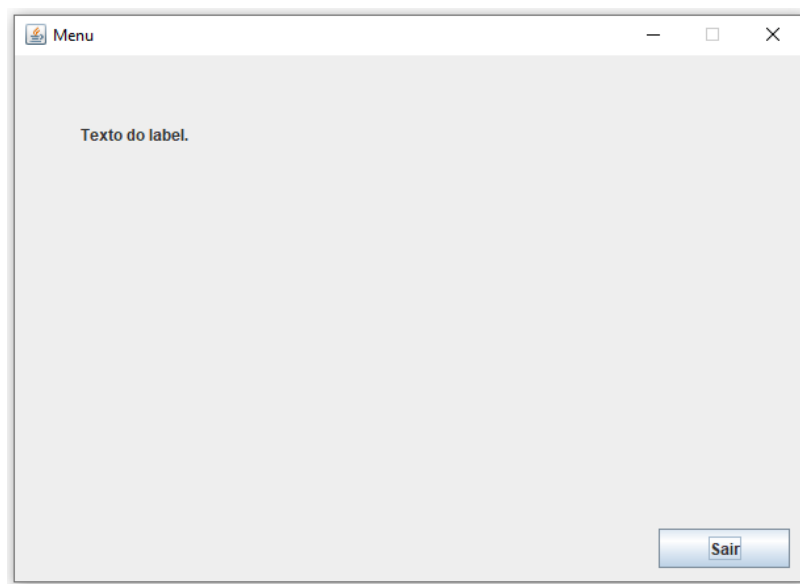
        // centralizar formulário
        setLocationRelativeTo(null);

        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);
    }
};

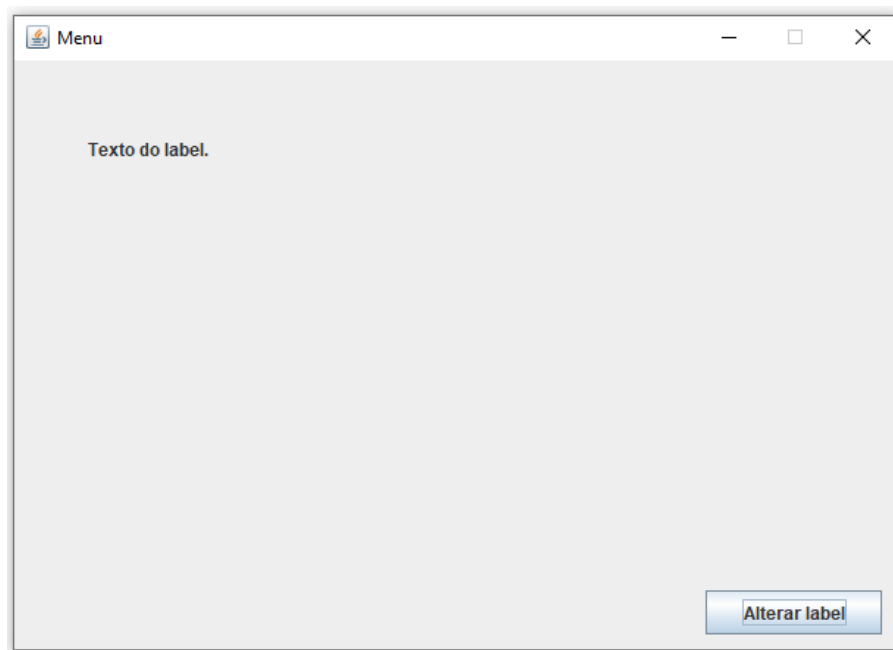
// botão
cmd1 = new JButton("Alterar label");
cmd1.setBounds(largura - 130, altura - 40, 120, 30);
cmd1.addActionListener(cmd1_click);
painel.add(cmd1);

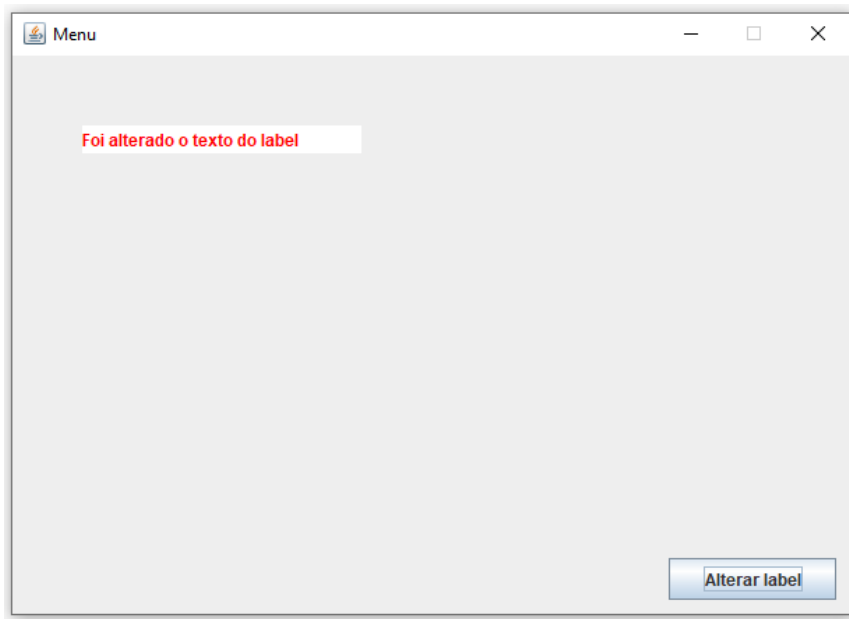
// label
label1 = new JLabel("Texto do label.");
label1.setBounds(50, 50, 200, 20);
painel.add(label1);

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
            }
        };
    }
}
```

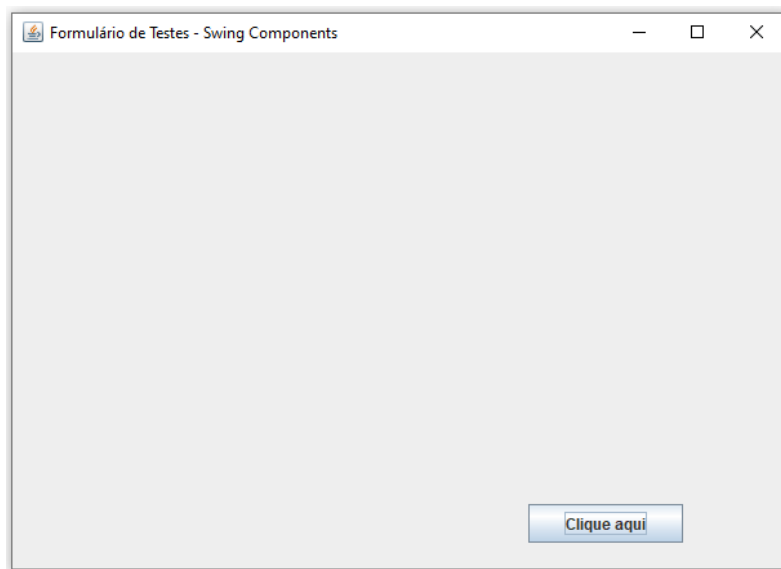
```
// criar cmd
cmd = new JButton("Clique aqui");
cmd.setBounds(400, 350, 120, 30);
cmd.addActionListener(cmd_click);
painel.add(cmd);

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);

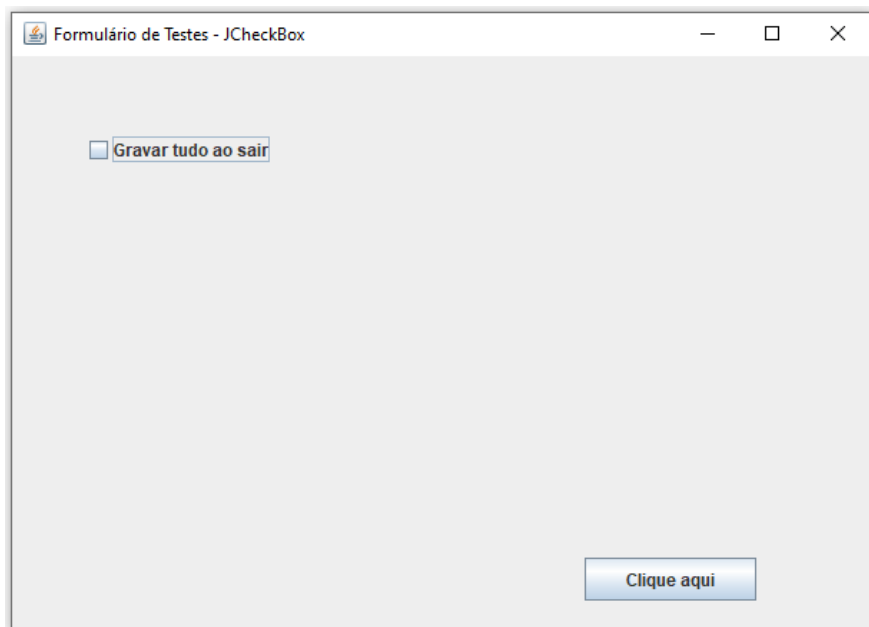
setLocationRelativeTo(null);
setVisible(true);

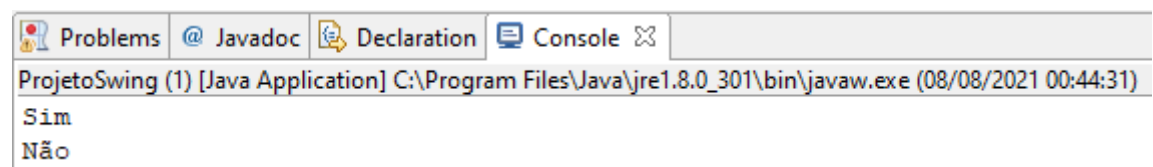
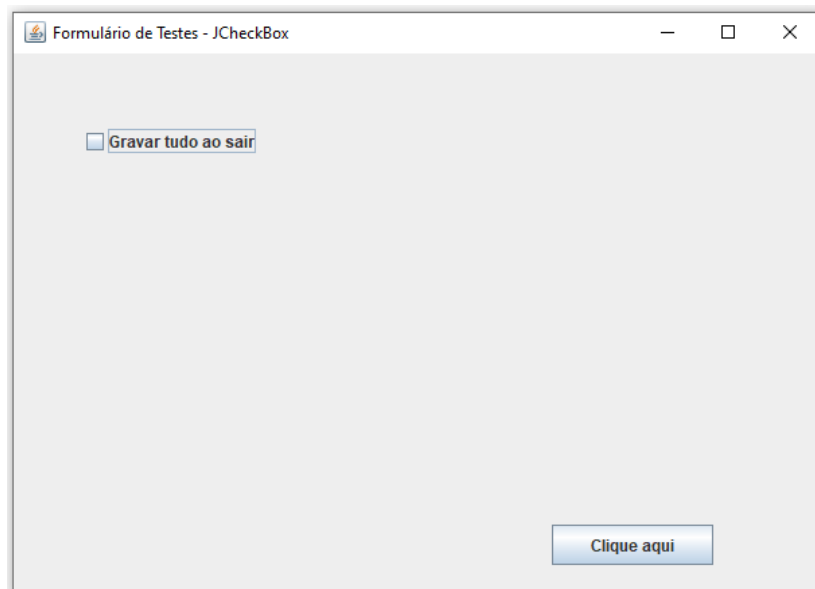
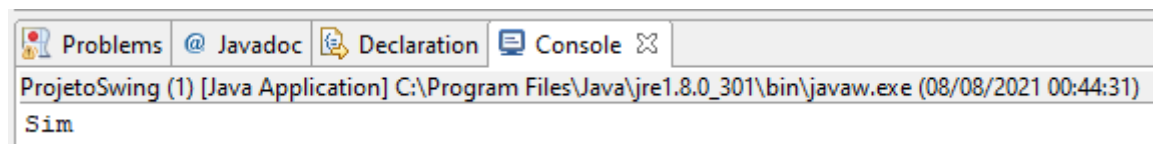
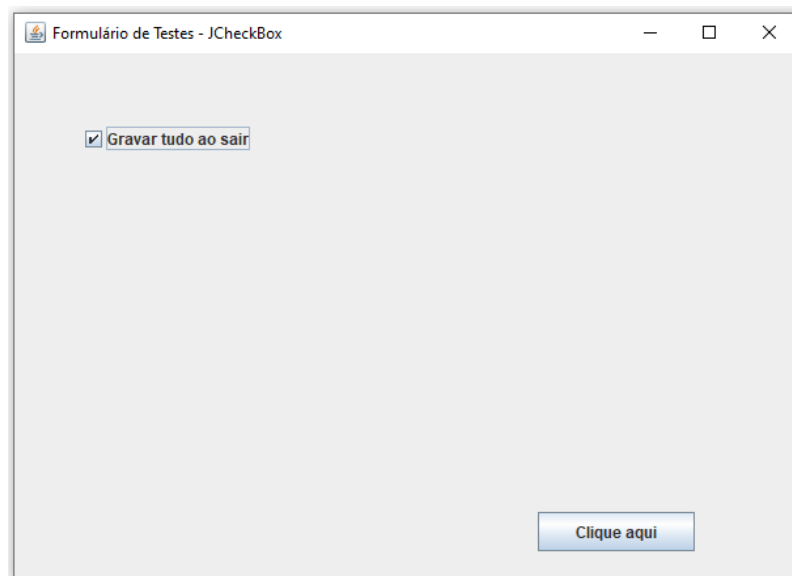
}

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("Divorciado");
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);
}

}

```

Formulário de Testes - Swing Components

☒ Solteiro  
☐ Casado  
☐ Divorciado  
☐ Viúvo

[Clique aqui](#)

Problems Javadoc Declaration Console

ProjetoSwing (2) [Java Application] C:\Program Files\Java\jre1.8.0\_301\bin\javaw.exe (08/08/2021 01:42:36)

```
solteiro  
divorciado  
casado  
Este indivíduo é casado.
```

## 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);

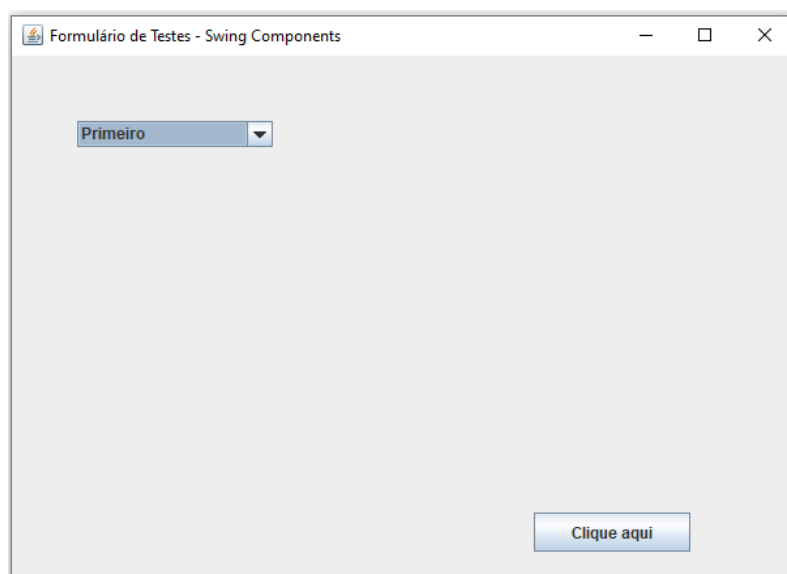
setLocationRelativeTo(null);
setVisible(true);

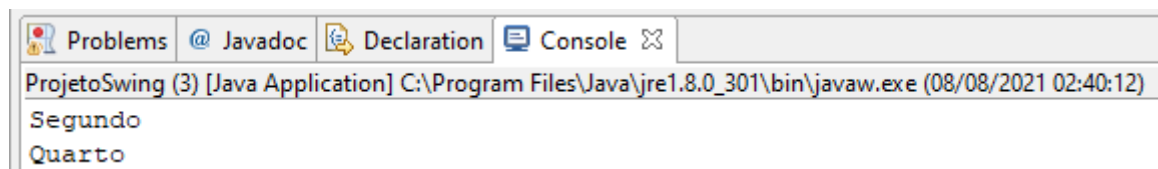
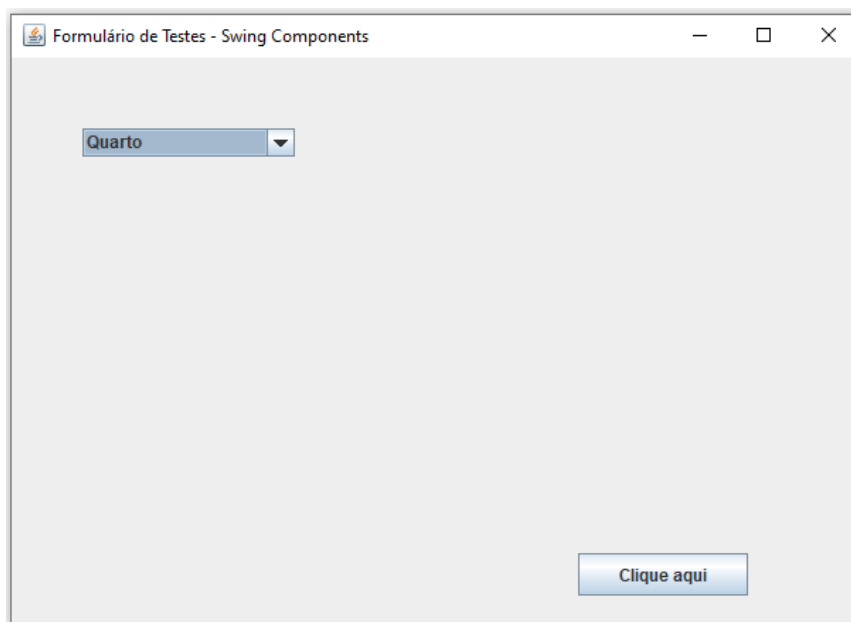
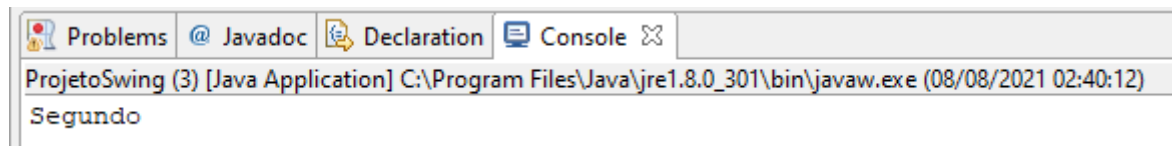
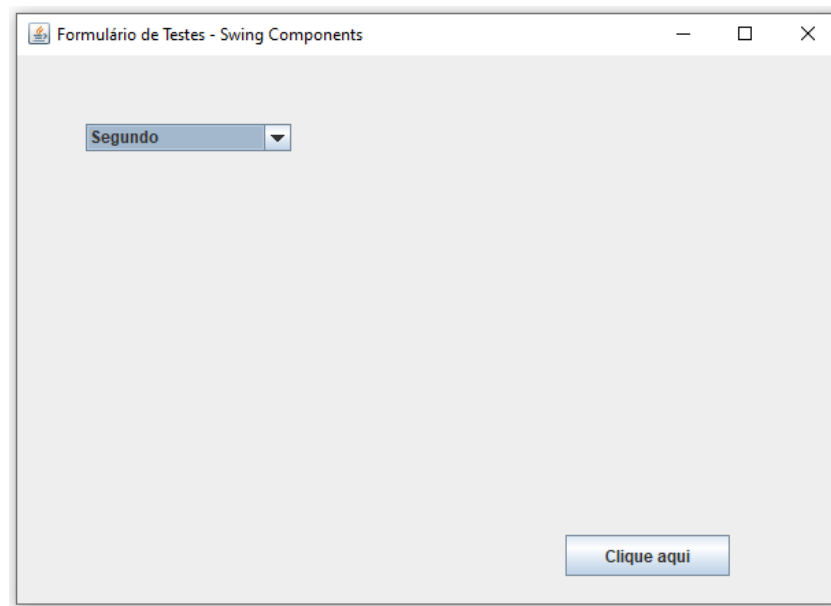
}

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);
}
}

```

