

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Instituto de Informática  
Departamento de Informática Aplicada

## Aula 4: Programação orientada a objetos [2] (Laboratório 1)

Prof. Dennis Giovani Balreira  
(Material adaptado do Prof. Thiago L. T. da Silveira)



INF01120 - Técnicas de Construção de Programas



# Instalação JRE e JDK

# Instalação JRE/JDK (Ambos)

- Os passos a seguir são [sugestões](#) de instalação
- Em caso de [dúvidas ou problemas](#) durante as instalações, independentemente de seu Sistema Operacional, poste no [Fórum de Dúvidas](#)
  - Tente ser objetivo e dê o máximo de [informações](#) possível sobre a dúvida, problema ou erro;
  - Os colegas que tiverem soluções, podem (e são convidados a) compartilhá-las pelo [fórum de dúvidas](#);
- Ademais, lembre-se que o Google é seu amigo! A maioria das soluções para problemas de instalação do JRE e JDK são facilmente encontradas por lá

# Instalação JRE/JDK (Windows)

- Para **desenvolvermos** em Java, precisamos instalar o JRE e o JDK
  - Os downloads via **Oracle** se dão por JRE\* e JDK\*
  - Escolher versão adequada para sua máquina (para Windows 11, por exemplo, geralmente se utiliza “Windows x64” com extensão “.exe”)
  - Criar uma **conta Oracle gratuita** para realizar os downloads
- Execute os instaladores (arquivos “.exe”) e clique em “Instalar” e, ao fim, “Fechar” (ou “Next”, “Next”, “Close”)
- **Recomenda-se instalar primeiro o JRE e depois o JDK**

\*<https://www.oracle.com/java/technologies/downloads/#java8>

# Instalação JRE/JDK (Windows)

- Para testar as instalações do JRE, abra um terminal (“Prompt de Comando”) e digite “**java -version**”; Deve-se ter uma saída diferente de “*java is not recognized as an internal or external command*”
- Para testar as instalações do JDK, abra um terminal (“Prompt de Comando”) e digite “**javac -version**”; Deve-se ter uma saída diferente de “*javac is not recognized as an internal or external command*”
- O comando “**java**” deve funcionar diretamente
- O comando “**javac**” pode falhar a depender das configurações do Windows. Nesse caso, siga as instruções do próximo slide

# Instalação JRE/JDK (Windows)

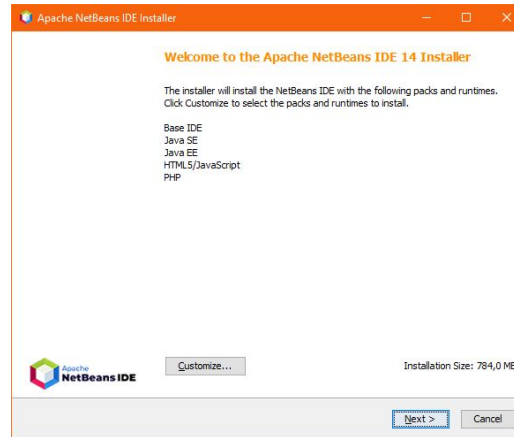
- Se o comando “**javac**” falhar, temos que adicionar e configurar duas variáveis de ambiente do/no Windows
  - Na barra de procura do Windows (10, nesse exemplo) escreva “Exibir configurações avançadas do sistema” e abra a opção selecionada; Clique em “Variáveis de Ambiente...”;
  - Na lista “Variáveis do sistema”, clique em “Novo...”; Em “Nome da variável” escreva “JAVA\_HOME” (sem as aspas) e no “Valor da variável” escreva o caminho de instalação do JDK ([Verifique qual foi utilizado em seu sistema](#)). Um exemplo é “C:\Program Files\Java\jdk.1.8.0\_281\”. Clique em OK;
  - Na lista “Variáveis do sistema”, selecione a entrada “Path” e clique em “Editar...”; Depois, clique em “Novo” e escreva “%JAVA\_HOME%\bin” (sem as aspas). Clique em OK.
  - Vá clicando em OK até que tudo se feche;
  - Teste o comando “**javac -version**” no terminal novamente.

# Instalação JRE/JDK (Ubuntu)

- Para **desenvolvermos** em Java, precisamos instalar o JRE e o JDK;
- Para instalação, abra um terminal e digite
  - **`sudo apt update`** para atualizar o índice de pacotes;
  - **`sudo apt install default-jre`** para instalar o JRE;
  - **`sudo apt install default-jdk`** para instalar o JDK;
- Para testar a instalação do JRE, digite **`java -version`**; Deve-se ter uma saída diferente de *`java: command not found`*;
- Para testar a instalação do JDK, digite **`javac -version`**; Deve-se ter uma saída diferente de *`javac: command not found`*;

# Instalação Java com NetBeans

- Instalar o **JRE e JDK** conforme passos anteriores
- Baixar instalador do **NetBeans** em <https://netbeans.apache.org/download/index.html>
  - Seguir os passos de instalação normalmente





# Instalação Java com VSCode

- Instalar o **JRE** conforme passos anteriores
- Seguir os passos descritos em <https://code.visualstudio.com/docs/languages/java>
  - “Pacote” de suporte Java para quem **não tem** VsCode instalado
  - “Pacote” de suporte Java para quem **já tem** VsCode instalado

## Install Visual Studio Code for Java

To help you set up quickly, we recommend you use the **Coding Pack for Java**, which is the bundle of VS Code, the Java Development Kit (JDK), and a collection of suggested extensions by Microsoft. The Coding Pack can also be used to fix an existing development environment.

Install the Coding Pack for Java - Windows

Install the Coding Pack for Java - macOS

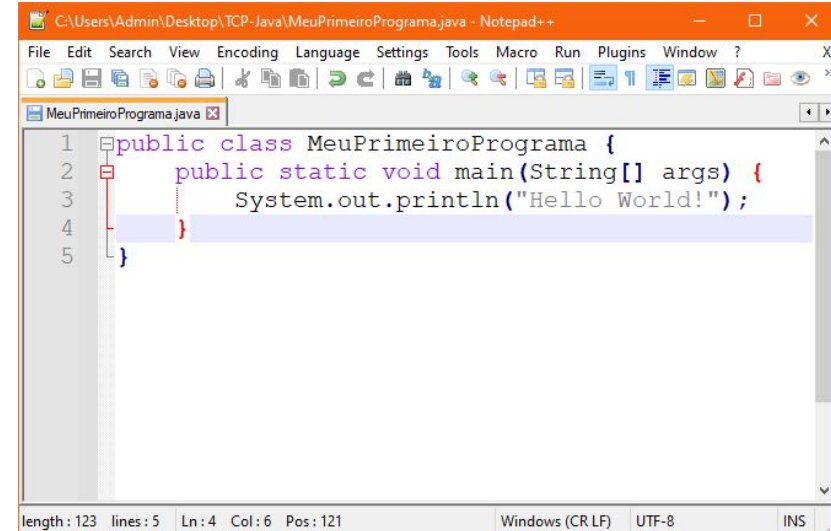
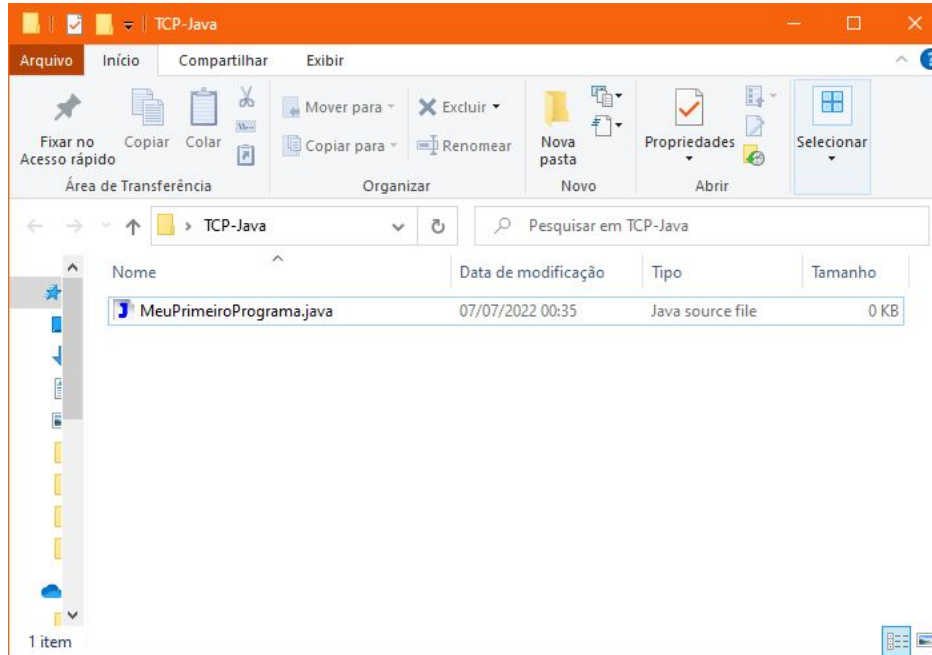
**Note:** The Coding Pack for Java is only available for Windows and macOS. For other operating systems, you will need to manually install a JDK, VS Code, and Java extensions.

# Compiladores de Java Online

- <https://replit.com/languages/java10>
- <https://www.jdoodle.com/online-java-compiler/>
- <https://www.programiz.com/java-programming/online-compiler/>
- [https://www.onlinegdb.com/online\\_java\\_compiler](https://www.onlinegdb.com/online_java_compiler)

# Rodando um programa Java

# Programando, salvando, compilando, executando...



# Programando, salvando, compilando, executando...

The image illustrates the workflow of a Java program:

- File Explorer (Top Left):** Shows a folder named 'TCP-Java' containing a file named 'MeuPrimeiroPrograma.java'.
- Code Editor (Bottom Left):** Displays the source code for 'MeuPrimeiroPrograma.java':

```
1 public class MeuPrimeiroPrograma {  
2     public static void main(String[] args) {  
3         System.out.println("Hello World!");  
4     }  
5 }
```
- File Explorer (Middle):** Shows the same folder 'TCP-Java' after compilation, now containing two files: 'MeuPrimeiroPrograma.class' (Arquivo CLASS, 1 KB) and 'MeuPrimeiroPrograma.java' (Java source file, 1 KB).
- Windows PowerShell (Right):** Shows the execution of the program:

```
PS C:\Users\Admin\Desktop\TCP-Java> javac MeuPrimeiroPrograma.j  
ava  
PS C:\Users\Admin\Desktop\TCP-Java> java MeuPrimeiroProgram  
Hello World! :)  
PS C:\Users\Admin\Desktop\TCP-Java>
```

Red arrows indicate the flow: from the source file to the compilation command, from the source file to the execution command, from the code editor to the execution command, and from the compiled class file to the execution command.

# Laboratório 1

# Exercício 1 - Parte 0

- Uma imobiliária quer identificar oportunidades de lucro em suas futuras locações e vendas. Há uma taxa fixa aplicada sobre o valor de um imóvel, um imposto, a qual o empresário não quer considerar em suas contas. Um imóvel também ganha valor agregado de tempos em tempos. Ajude o empresário a **modelar uma aplicação** que visa entender melhor essas duas questões (**incidência de imposto e valorização**)



# Exercício 1 - Parte 1

- Construir uma classe em Java que representa um **imóvel**, chame-a de **Imovel1**, conforme a especificação a seguir (foque nos **atributos**, por enquanto):



Imovel
<div>int nQuartos</div> <div>int nBanheiros</div> <div>boolean temGaragem</div> <div>String endereco</div> <div>boolean aluguel</div> <div>double valorVenda</div>
<div>void imprimeDados()</div> <div>double calculaImposto()</div> <div>void valoriza(double perc)</div>



# Exercício 1 - Parte 1

- Construir uma classe em Java que representa um **imóvel**, chame-a de **Imovel1**, conforme a especificação a seguir (foque nos **atributos**, por enquanto):

```
public class Imovel{  
  
    int nQuartos;  
    int nBanheiros;  
    // outros atributos  
  
}
```

Imovel
int nQuartos int nBanheiros boolean temGaragem String endereco boolean aluguel double valorVenda
void imprimeDados() double calculaImposto() void valoriza(double perc)

## Exercício 1 - Parte 2

- Construir uma classe em Java que representa o programa principal e contém o método `main`. Chame-a de **ProgramaPrincipal**:

```
public class ProgramaPrincipal{  
  
    public static void main (String[] args){  
        // implementação  
    }  
}
```

## Exercício 1 - Parte 2

- O método **main** de **ProgramaPrincipal** deve declarar e instanciar um objeto do tipo **Imovel** e atribuir valores a cada um de seus atributos. Por exemplo:

```
// ...  
Imovel im1; // declaracao do objeto  
im1 = new Imovel(); // instanciacao do objeto  
  
im1.nQuartos = 2;  
im1.nBanheiros = 1;  
im1.temGaragem = false;  
im1.endereco = "Rua dos Fulanos, n. 0";  
im1.aluguel = true;  
im1.valorVenda = 142000;
```

# Exercício 1 - Parte 3

- Implementar os **métodos** de **Imovel**, conforme a especificação a seguir:

```
public tipoRetorno nomeMetodo (tipoArgumento argumento){  
  
    // implementação do método  
  
}
```

- **tipoRetorno** deve ser **void** se não houver retorno ou especificar o tipo de dados de retorno;
- **tipoArgumento** deve ser **void** se não houver argumentos (opcional) ou especificar um tipo de dados. Há um **tipoArgumento** por **argumento**.

# Exercício 1 - Parte 3

- Implementar os **métodos** de **Imovel**, conforme a especificação a seguir:
- **void imprimeDados()** mostra na tela todos os dados do imóvel (use **System.out** para impressão); Formate a saída (Ex.: “Quartos: 2”; “Garagem: sim”, etc.);
- **void valoriza(double perc)** aumenta **perc** por cento (%) do atributo **valorVenda**, simulando a valorização de um imóvel;

Imovel
int nQuartos int nBanheiros boolean temGaragem String endereco boolean aluguel double valorVenda
void imprimeDados() double calculaImposto() void valoriza(double perc)

# Exercício 1 - Parte 3

- Implementar os **métodos** de **Imovel**, conforme a especificação a seguir:
- **double calculaImposto()** retorna o **valor dos impostos incidentes** dependendo do **valorVenda** e se o imóvel está disponível para **aluguel** ou venda:
  - venda: 12.0% do valor de venda;
  - aluguel: 8.0% do valor de venda;

Imovel
int nQuartos int nBanheiros boolean temGaragem String endereco boolean aluguel double valorVenda
void imprimeDados() double calculaImposto() void valoriza(double perc)

# Exercício 1 - Parte 4

- Testar os métodos da instância **im1** já criada em **ProgramaPrincipal**;
  - Imprima os dados de **im1** com o método **imprimeDados**;
  - Valorize o imóvel **im1** em 3.2% com o método **valoriza**;
  - Imprima novamente os dados de **im1** com o método **imprimeDados**;
  - Imprima o valor do imposto incidente sobre **im1** ao chamar o método **calculaImposto**. A impressão do valor do imposto deve ocorrer no método **main** de **ProgramaPrincipal**.

# Exercício 1 - Parte 5

- Crie uma nova instância de **Imovel**, **im2**, e atribua os mesmos valores que aqueles atribuídos aos atributos de **im1** (atributo por atributo);
- Compare **im1** e **im2** utilizando o código abaixo:

```
if (im1 == im2){  
    System.out.println("Mesmos imoveis!");  
}else{  
    System.out.println("Imoveis distintos!");  
}
```

- **O que acontece?**
  - Pesquise\* porquê isto acontece, respondendo com comentário no próprio código

\* <https://www.javatpoint.com/how-to-compare-two-objects-in-java>



## Exercício 1 - Parte 6

- Declare uma variável **im3** do “tipo” **Imovel** e atribua **im1** a ela;
- Modifique **valorVenda** de **im3** e imprima os dados de **im1** e **im3**;

```
Imovel im3 = im1;  
im3.valorVenda = 0; // imovel com perda total  
im3.imprimeDados();  
im1.imprimeDados();
```

- **O que acontece?**
  - Pesquise\* porquê isto acontece, respondendo com comentário no próprio código

\* <https://sites.google.com/view/javacommiza/home/capitulo-01/destruindo-objetos/coleta-de-lixo/objetos-vs-referencias?pli=1>

# Exercício 1 - Entrega

- Envie um único arquivo no formato **ZIP** contendo a implementação do **Laboratório 1 - Exercício 1** (Imovel.java e ProgramaPrincipal.java)
  - Indique no próprio código com um comentário (“//” ou “/\* \*/”) **o que acontece** nas partes 5 e 6 do Exercício 1
- O nome do arquivo deve seguir o formato:  
"l1-<primeiro\_nome\_do\_aluno>-<último\_nome\_do\_aluno>.zip"
- Atente ao prazo de entrega do trabalho especificado no Moodle!
- Bom trabalho! Qualquer dúvida contate o professor

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Instituto de Informática  
Departamento de Informática Aplicada

**Obrigado pela atenção!**  
**Dúvidas?**

Prof. Dennis Giovani Balreira  
(Material adaptado do Prof. Thiago L. T. da Silveira)



INF01120 - Técnicas de Construção de Programas

