Linguagem de Programação II – IMD0040

Aula 06 – Introdução ao Eclipse





Integrated Development Environment

- ☐ Ferramenta/Ambiente para desenvolvimento.
- Um só programa que permita fazer "tudo"
 - * Busca-se um aumento na produtividade.
- □ Facilita a configuração do ambiente de desenvolvimento.
 - *É mais rápido aprender a usar a ferramenta do que configurar tudo na mão.

IDEs para Java

☐ NetBeans



Eclipse



☐ JDeveloper





- Disponibilização de ferramentas para auxiliar no desenvolvimento de aplicações.
- ☐ Mantido pela Eclipse Foundation.
- 🖵 Multiplataforma.
- Disponível para Windows, Linux, Solaris, MacOS
- Gratuito e de código aberto
- ☐ Linguagens: Java, C/C++, PHP, HTML...
- ☐ Simplifica desenvolvimento e aumenta a produtividade
- ☐ Distribuído em diversos idiomas (PT-BR inclusive)
- Uma das mais poderosas ferramentas para o desenvolvimento de aplicações



- Instalar o JDK (Java Development Kit)
- Baixar uma versão do Eclipse

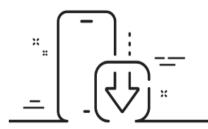


https://www.eclipse.org





Projects Supporters Collaborations Resources The Foundation



Download Eclipse Technology that is right for you





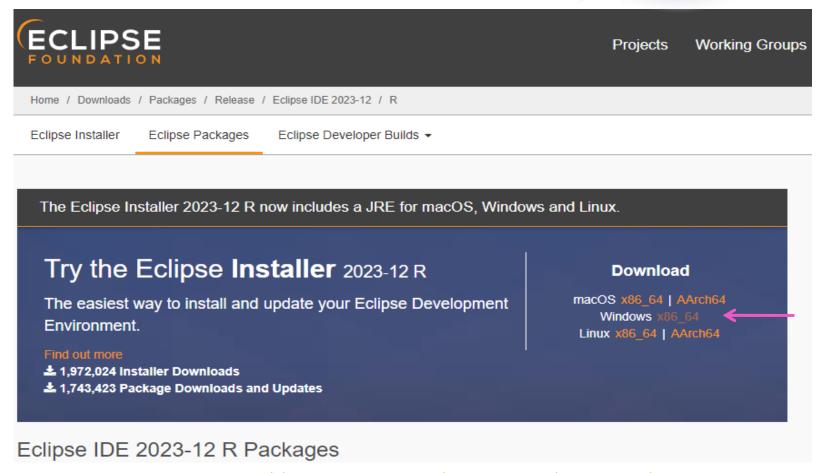
https://www.eclipse.org/downloads/



List of Projects				
	Search	Technology Types - Any -	State - Any -	Include Archived Projects Apply
	Eclipse EMFStore Modeling Eclipse EMFStore is a model report versioning of models. Read more Report an Issue 0	sitory (server) for EMF featuring o	collaborative editing a	and Latest release: 1.10 Download ▼
epsilon	Eclipse Epsilon™ Modeling Eclipse Epsilon™ is a family of lan transformation, model validation, c with EMF and Read more Report an Issue 0	guages and tools for code genera omparison, migration and refacto		

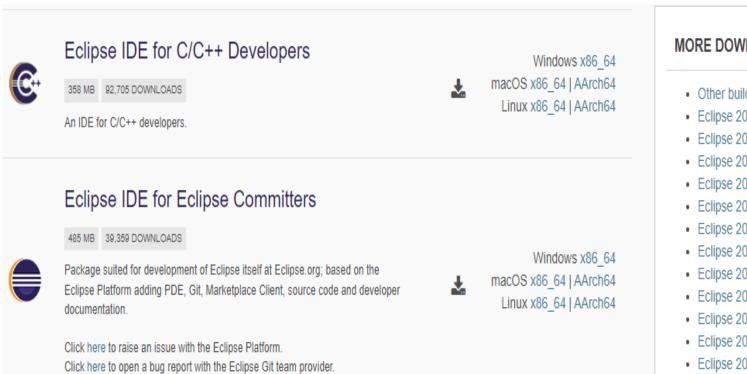
https://projects.eclipse.org/





https://www.eclipse.org/downloads/packages/



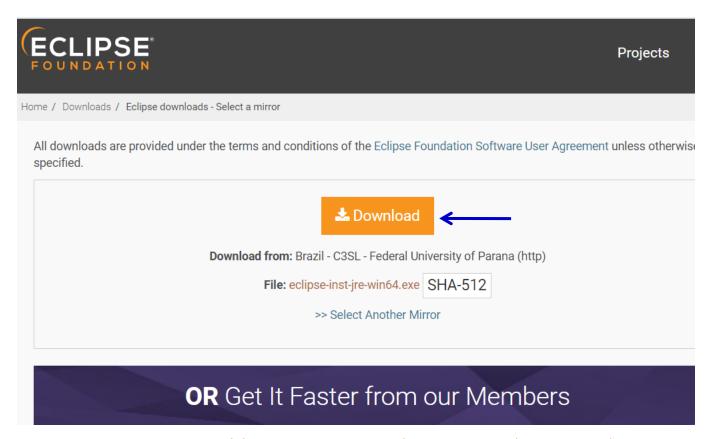


MORE DOWNLOADS

- Other builds
- Eclipse 2023-12 (4.30)
- Eclipse 2023-09 (4.29)
- Eclipse 2023-06 (4.28)
- Eclipse 2023-03 (4.27)
- Eclipse 2022-12 (4.26)
- Eclipse 2022-09 (4.25)
- Eclipse 2022-06 (4.24)
- Eclipse 2022-03 (4.23)
- Eclipse 2021-12 (4.22)
- Eclipse 2021-09 (4.21)
- Eclipse 2021-06 (4.20)
- Eclipse 2021-03 (4.19)
- Older Versions

https://www.eclipse.org/downloads/packages/

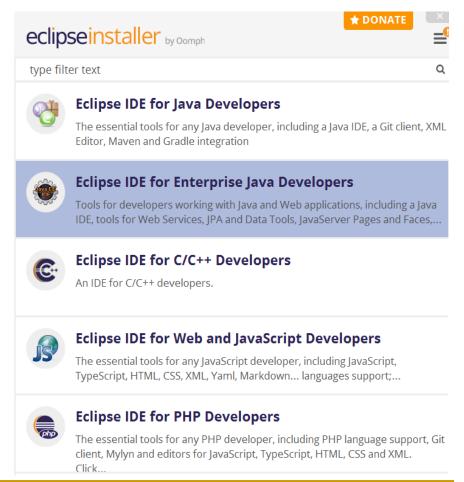




https://www.eclipse.org/downloads/packages/

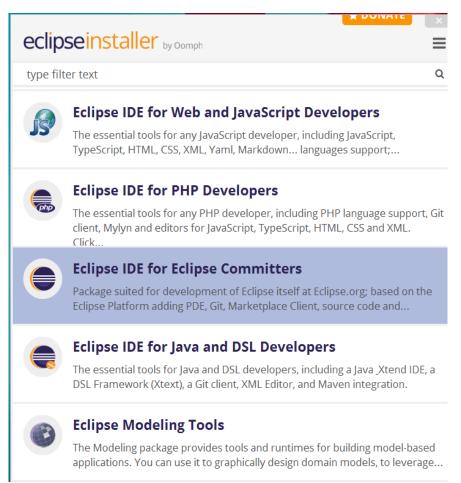




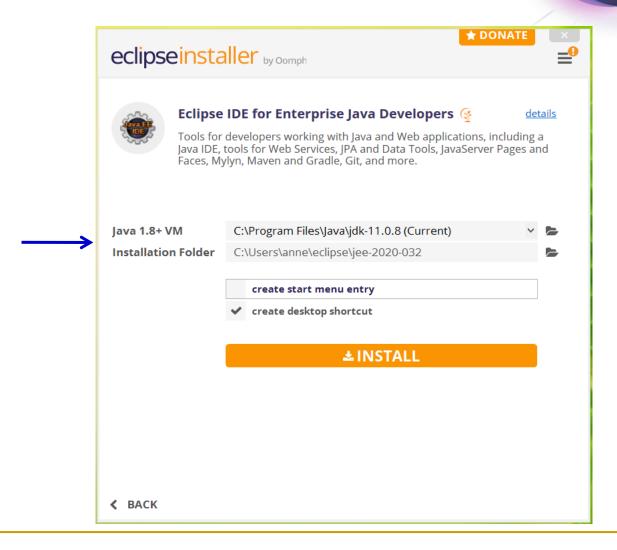




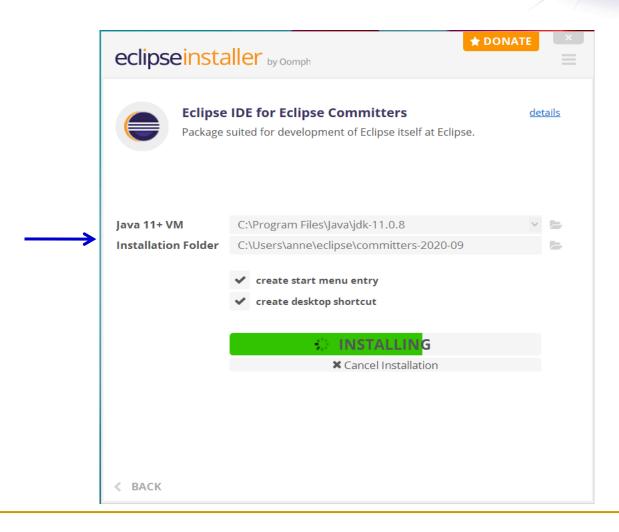




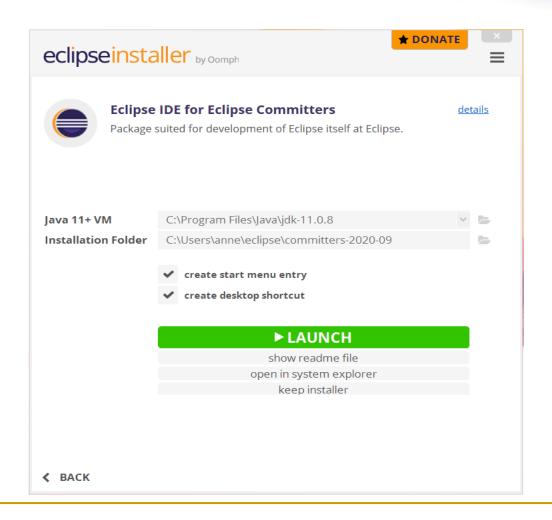




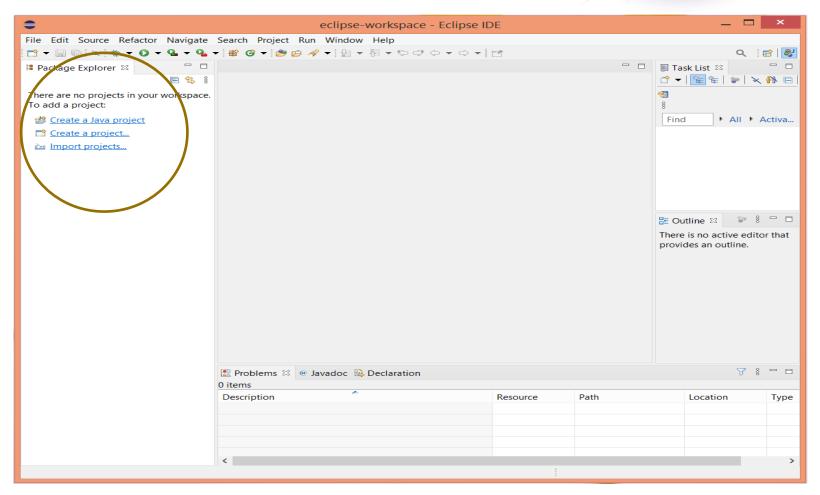




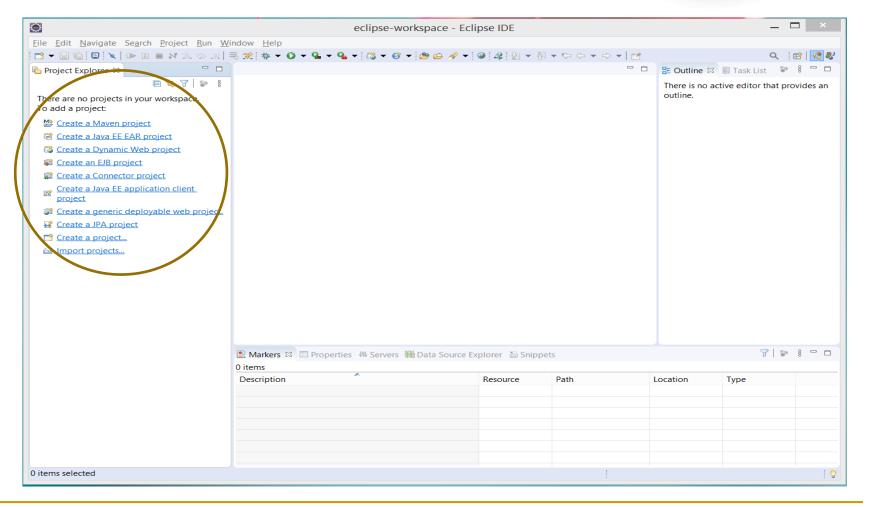






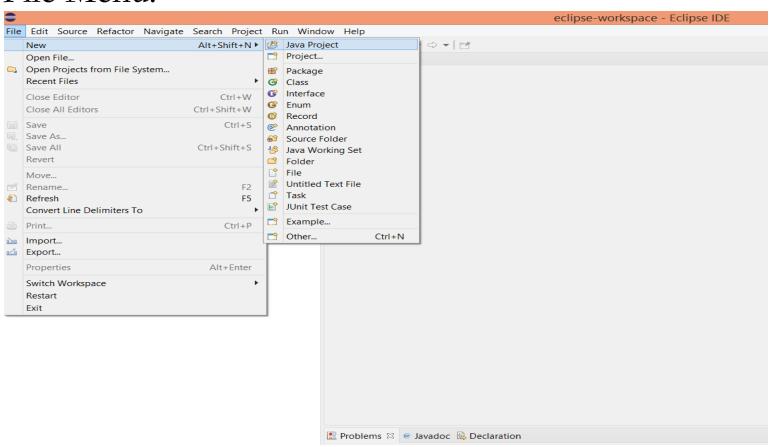






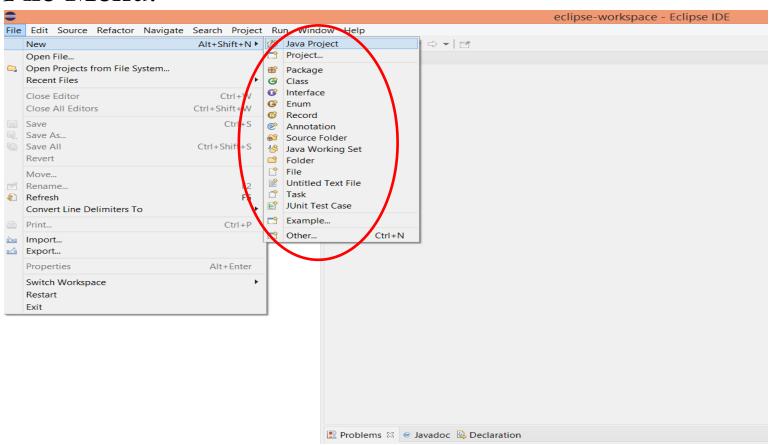


☐ File Menu:



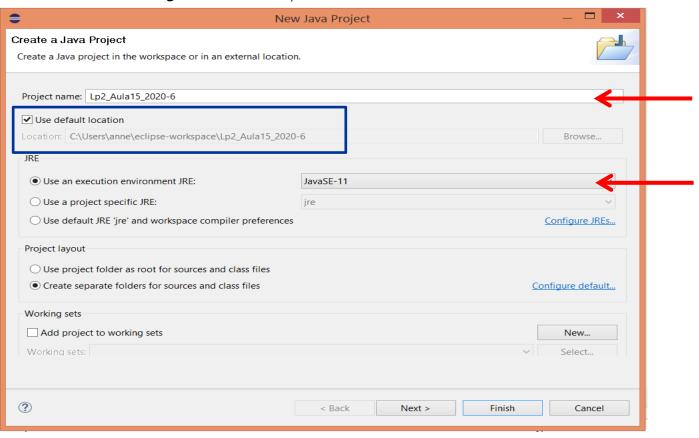


☐ File Menu:



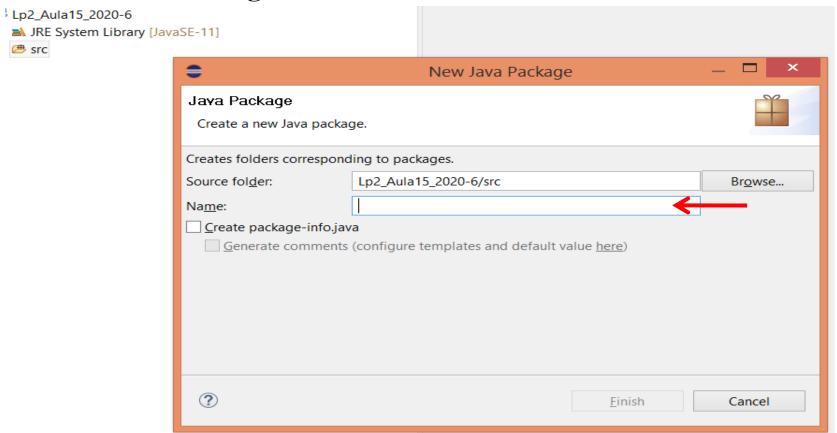


 \square File \rightarrow New \rightarrow Java Project



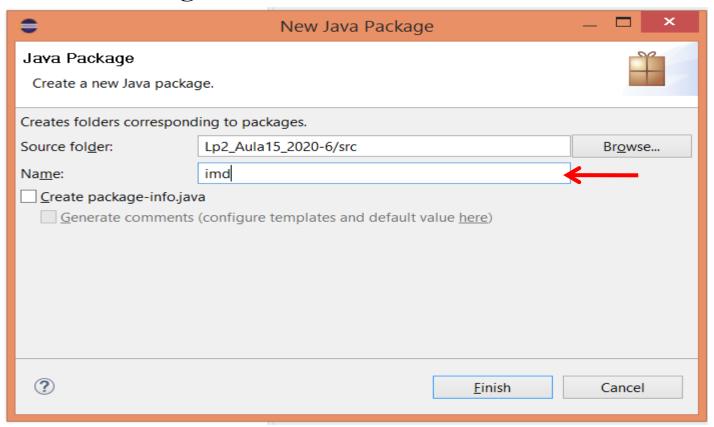


\square New \rightarrow Package:

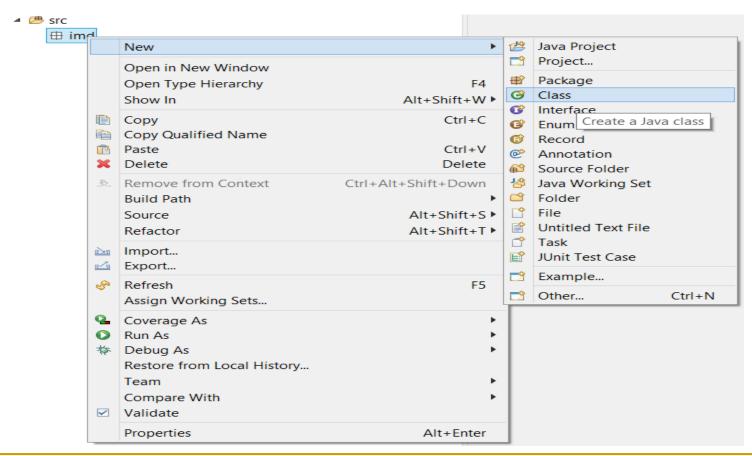




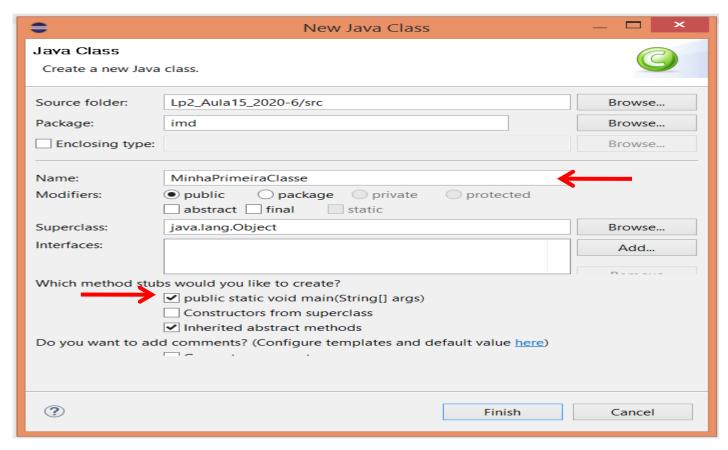
 \square New \rightarrow Package:



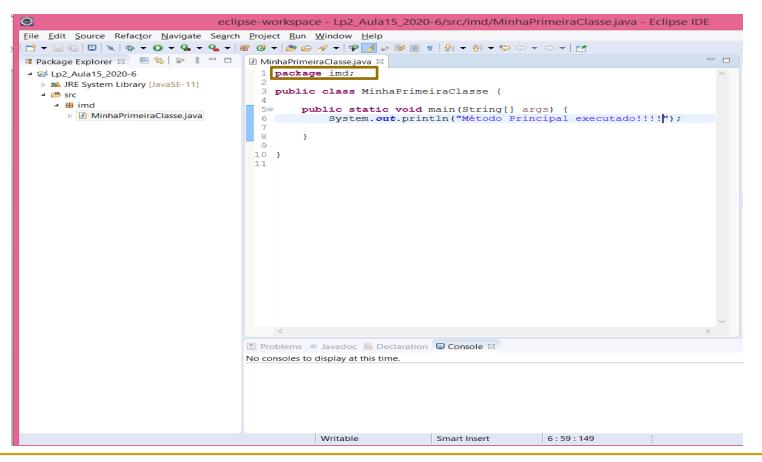




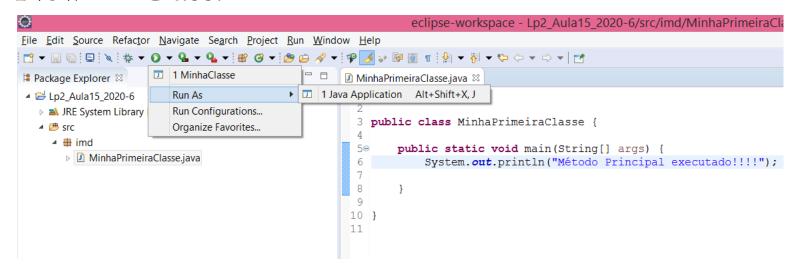










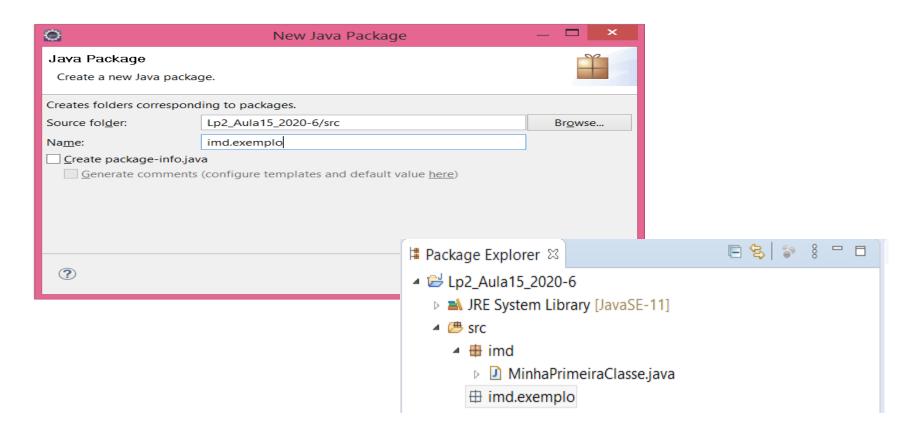


```
▼ Problems @ Javadoc Declaration Console Console Console MinhaPrimeiraClasse [Java Application] C:\Program Files\Java\jdk-11.0.8\bin\javaw.exe (26 de set de 2020 19:38:03 – 19:38:04)

Método Principal executado!!!!
```

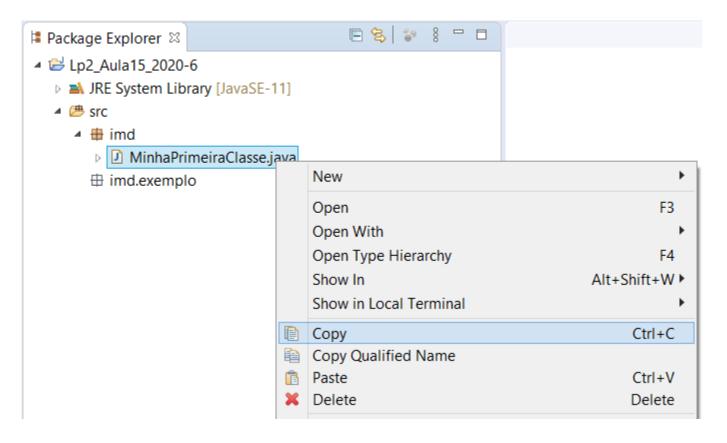


☐ Copiando e Movendo Classes:



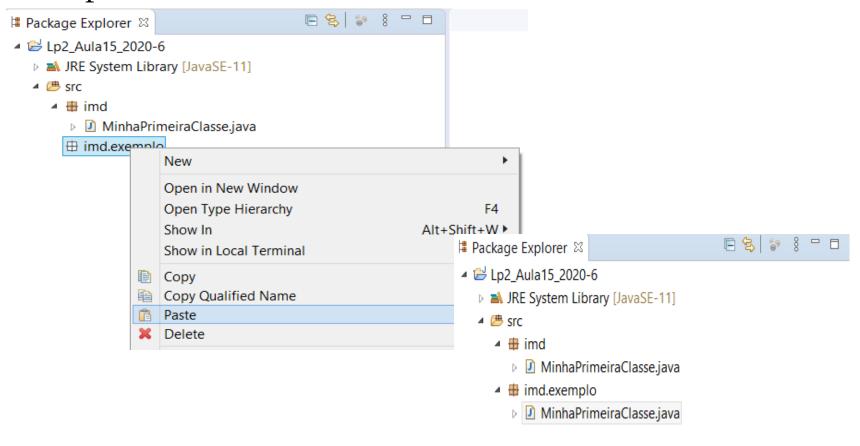


Copiando e Movendo Classes:



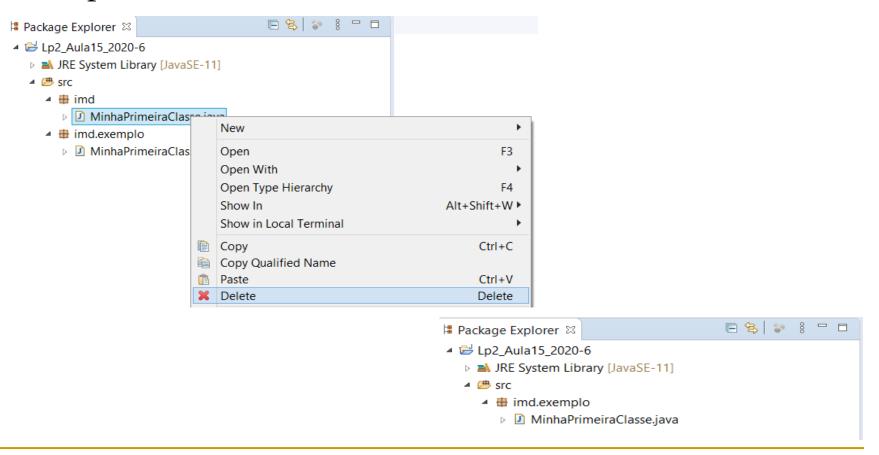


Copiando e Removendo Classes:





Copiando e Removendo Classes:





Copiando e Removendo Classes:

```
Package Explorer ⋈

▲ B Lp2_Aula15_2020-6

                                          package imd.exemplo;
 3 public class MinhaPrimeiraClasse {

♣ imd.exemplo

                                              public static void main(String[] args) {
     MinhaPrimeiraClasse.java
                                                 System.out.println("Método Principal executado!!!!");
```



```
☐ Package Explorer 
☐

                                      🔬 ContaBancaria.java 🛭

♣ B Lp2_Aula15_2020-6

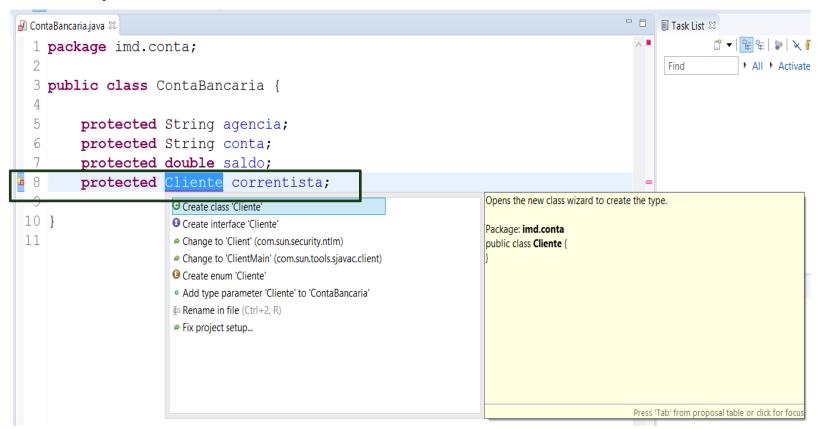
                                         1 package imd.conta;
 3 public class ContaBancaria {
   imd.conta
     De La ContaBancaria.java
                                               protected String agencia;

→ 

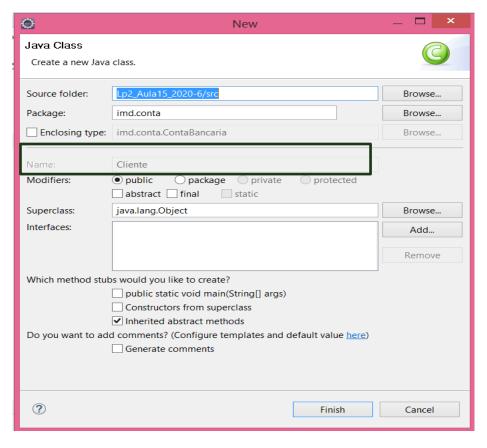
    imd.exemplo

                                               protected String conta;
                                               protected double saldo;
                                               protected Cliente correntista;
                                       10 }
```







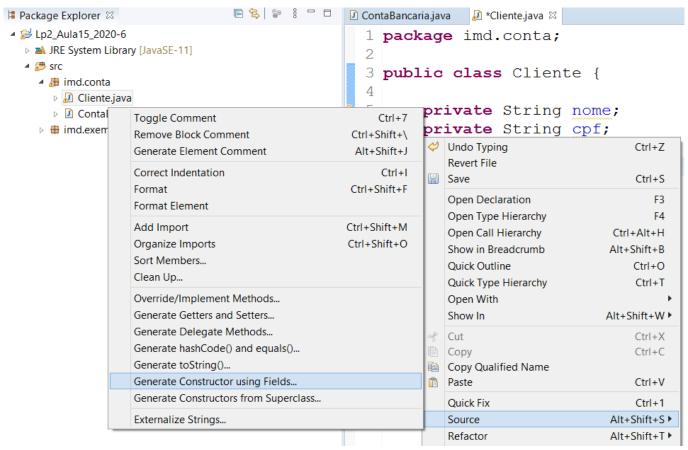




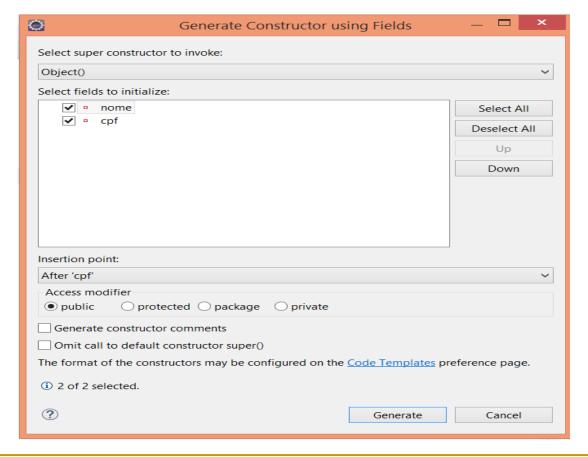
```
# Package Explorer 
□ ContaBancaria.java

□ Cliente.java
□ Lp2_Aula15_2020-6
□ M JRE System Library [JavaSE-11]
□ M B src
□ ContaBancaria.java
□ Cliente.java
□ D ContaBancaria.java
□ D ContaBancaria.java
□ D ContaBancaria.java
□ D ContaBancaria.java
□ ContaBancaria.java
□ D ContaBancaria.java
□ D
```











```
ContaBancaria.java

∠ Cliente.java 

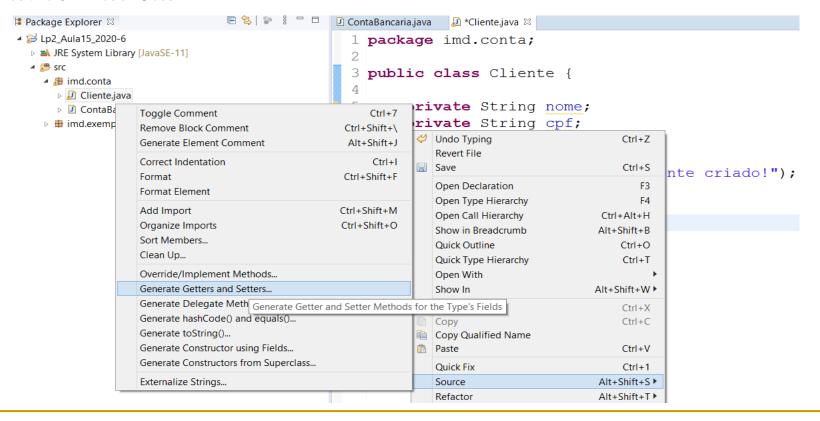
□
 1 package imd.conta;
 3 public class Cliente {
 5
        private String nome;
        private String cpf;
 8⊜
        public Cliente(String nome, String cpf)
             super();
10
             this.nome = nome;
11
             this.cpf = cpf;
12
13
14
15 }
16
```



```
ContaBancaria.java
             💹 Cliente.java 🛭
 1 package imd.conta;
 3 public class Cliente {
        private String nome;
        private String cpf;
        public Cliente() {
             System.out.println("Cliente criado!");
10
11
12 }
13
```

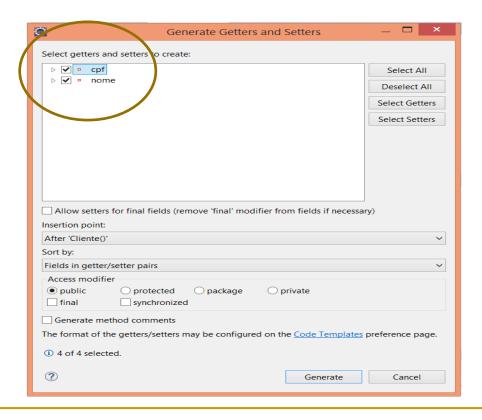


Gerando métodos Getters e Setters de forma automática:





Gerando métodos Getters e Setters de forma automática:





☐ Gerando métodos Getters e Setters de forma automática:

```
public String getNome() {
    return nome;
public void setNome(String nome) {
    this.nome = nome;
public String getCpf() {
    return cpf;
public void setCpf(String cpf) {
    this.cpf = cpf;
```



Refatoração de código de forma automática:

```
3 public class Cliente {
 4
           private String nom &
                                              Undo Typing
                                                                                Ctrl+Z
           private String cpf
 6
                                              Revert File
                                              Save
                                                                                Ctrl+S
 80
           public Cliente() {
                                              Open Declaration
                                                                                   F3
 9
                  System.out.pri
                                              Open Type Hierarchy
                                              Open Call Hierarchy
                                                                            Ctrl+Alt+H
10
                                              Show in Breadcrumb
                                                                            Alt+Shift+B
11
                                              Quick Outline
                                                                                Ctrl+O
120
           public String getN
                                              Quick Type Hierarchy
                                                                                Ctrl+T
13
                  return nome;
                                              Open With
14
                                              Show In
                                                                           Alt+Shift+W ▶
1.5
                                                                                Ctrl+X
           public void setNom
1 6⊜
                                              Copy
                                                                                Ctrl+C
17
                  this.nome = no
                                              Copy Qualified Name
                                              Paste
                                                                                Ctrl+V
18
19
                                              Quick Fix
                                                                                Ctrl+1
                                              Source
                                                                            Alt+Shift+S ▶
200
           public String getC
Rename...
                              Alt+Shift+R
                                              Refactor
                                                                            Alt+Shift+T ▶
Move...
                              Alt+Shift+V
                                              Surround With
                                                                            Alt+Shift+Z ▶
                                              Local History
Extract Interface...
Extract Superclass...
                                              References
Use Supertype Where Possible...
                                              Declarations
Pull Up...
                                           Add to Snippets...
Push Down...
                                              Coverage As
Extract Class...
                                              Run As
                                              Debug As
Encapsulate Field...
                                              Profile As
Generalize Declared Type...
                                              Team
```



Refatoração de código de forma automática:

```
public class Cliente {
     private String nomeCliente;
    private String Press Enter to refactor. Options... 

Press Enter to refactor. Options...
     public Cliente() {
          System.out.println("Cliente criado!");
     public String getNome() {
          return nomeCliente;
     public void setNome(String nome) {
          this.nomeCliente = nome;
```



Utilizando auto complete do Eclipse (CTL + space):

```
■ *Portifolio.java 

□
ContaBancaria.java
                    Cliente.java
     import java.util.ArrayList;
     public class Portifolio {
            public static ArrayList<ContaBancaria> contas;
            public static double totalPassivo;
            public static void main(String[] args) {
  9
 10
                   contas = new ArrayList<ContaBancaria>();
 11
 12
                   ContaBancaria c1 = new ContaBancaria();
213
                   c1.
 14

    agencia : String - ContaBancaria

 15

    conta : String - ContaBancaria

 16

    correntista : Cliente - ContaBancaria

    saldo: double - ContaBancaria

    equals(Object obj): boolean - Object

🔝 Problems @ Javadoc 🖳 D
                         getAgencia(): String - ContaBancaria
No consoles to display at this ti
                         getClass(): Class<?> - Object

    getConta(): String - ContaBancaria

    getCorrentista() : Cliente - ContaBancaria

    getSaldo(): double - ContaBancaria

                         hashCode(): int - Object
                         notifv(): void - Object
                                          Press 'Ctrl+Space' to show Template Proposals
```

Perguntas ...

