

# PROGRAMAÇÃO ORIENTADA A OBJETOS

## AULA 1 - APRESENTAÇÃO DA DISCIPLINA E PARADIGMAS DE LINGUAGEM

**WALTER TRAVASSOS SARINHO**

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR



**OLÁ!**  
**EU SOU WALTER TRAVASSOS**

ENTUSIASTA DE NOVAS TECNOLOGIAS E  
METODOLOGIAS INOVADORAS DE ENSINO  
APRENDIZAGEM.

VOCÊ PODE ME ENCONTRAR:

- INSTAGRAM: @PROF.WALTERTRAVASSOS
- E-MAIL: TRAVASSOSWALTER@GMAIL.COM
- WHATSAPP: (83) 98882-0007

♦ OLÁ! MEU NOME É ♦  
**WALTER  
TRAVASSOS**



# Psicólogo Clínico

**Abordagem Centrada na Pessoa – ACP**

**Eye Movement Desensitization and Reprocessing – EMDR**

**Terapia de Aceitação e Compromisso - ACT**

**CRP 13/10.495**

**Você pode me encontrar:**

- Instagram: **@psi.waltertravassos**
- WhatsApp: **(83) 98882-0007**

# Mini Currículo

- Graduado em Sistemas para Internet (IFPB - 2012)
- Mestre em Ciência da Computação (UFPE - 2014)
- Especialista em Tecnologias e Educação a Distância (Unidoctum – 2020)
- Graduado em Psicologia (Unipê - 2021)
- Especialista em Psicologia Comportamental e Cognitiva (Dom Alberto – 2021)
- Especialista em Tratamento dos Transtornos de Ansiedade e Síndrome do Pânico (Facuminas – 2021)
- Formação na Abordagem Centrada na Pessoa (ACPBR – 2022), Focalização (Focalização Brasil - 2023), EMDR (Trauma Clinic - 2023) e Terapia de Aceitação e Compromisso (Tiago Ferreira, 2025).
- Especialista em Psicologia Humanista: Abordagem Centrada na Pessoa (Facuminas – 2023)
- Especialista em Desenvolvimento de Sistemas com Java (Facuminas – 2025)
- Especialista em Tecnologias Digitais para Sala de Aula (Facuminas – 2025)
- Especialista em Autismo – (Facuminas – 2025)

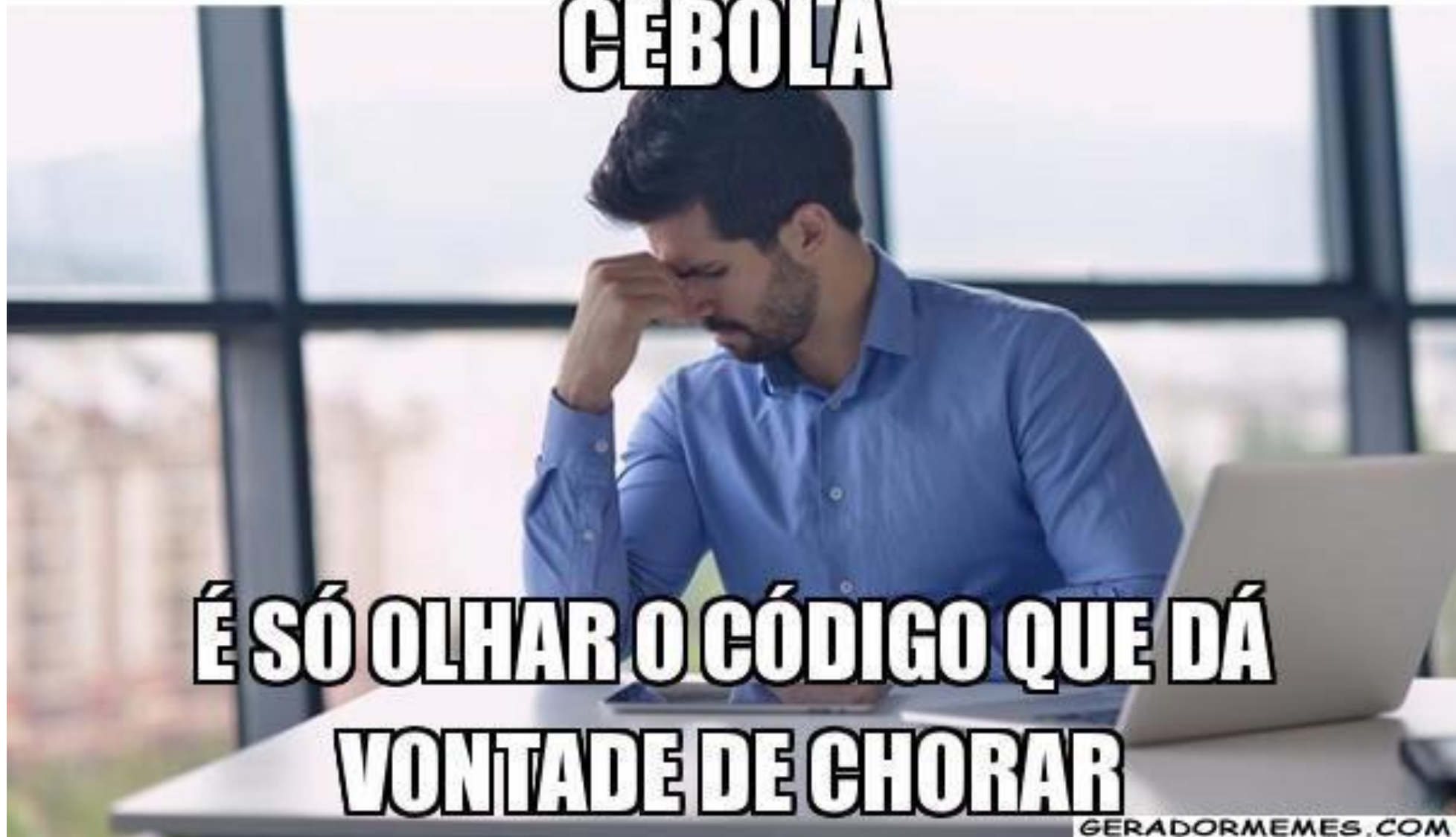
Professor de cursos técnicos desde 2007, na graduação desde 2012. Ministrei mais de 30 disciplinas diferentes na graduação. Professor de alguns módulos na pós-graduação.



QUAIS SÃO SUAS  
EXPECTATIVAS  
PARA ESTA  
DISCIPLINA?



**PROGRAMAÇÃO ORIENTADA A  
CÉBOLA**



**É SÓ OLHAR O CÓDIGO QUE DÁ  
VONTADE DE CHORAR**





Pensem nos objetos,  
como se fossem  
personagens de um RPG



# Conteúdo Programático



# Ementa

Introdução à orientação a objetos. Linguagens típicas orientadas a objetos. Programação Orientada a Objeto em Java. Conceitos básicos e terminologias de Programação orientada a objetos. Classe, Objetos, Atributos, Métodos, Construtores e sobrecarga. Instanciação e Referência de objetos. Envio de mensagens. Ciclo de vida de um objeto. Abstração e encapsulamento. Herança. Criação e uso de hierarquia de classes. Classes abstratas e Interfaces. Relacionamento entre classes. Polimorfismo. Ligação dinâmica (dynamic binding). Tratamento de exceções.

| Aulas | Conteúdo                                                | Listas         |
|-------|---------------------------------------------------------|----------------|
| 1     | Apresentação e Paradigmas de Linguagem                  | Padawan        |
| 2     | Sintaxe da Linguagem e Manipulação de Datas             |                |
| 3     | Visibilidade de variáveis, Classe String, Casting       |                |
| 4     | Operadores e métodos de conversão                       |                |
| 5     | Operador condicional if e formatação da saída de dados  |                |
| 6     | Operador condicional switch e operador de repetição for |                |
| 7     | Avaliação 1                                             | Cavaleiro Jedi |
| 8     | Operadores de repetição while e do while                |                |
| 9     | Vetores e ArrayList                                     | Mestre Jedi    |
| 10    | Interface Gráfica Java Swing                            |                |
| 11    | Avaliação A2                                            |                |
| 12    | O paradigma orientado a objetos                         | Anakin         |
| 13    | Construindo métodos e Métodos Construtores              |                |
| 14    | Pilares da POO                                          | Obi Wan        |
| 15    | Classes Abstratas e Interfaces                          |                |
| 16    | Avaliação de Projeto                                    |                |
| 17    | Avaliação 3                                             |                |
| 18    | Avaliação Final                                         |                |



# AVALIAÇÃO





# Avaliação

1ª Unidade (0,00 a 10,00):

- Prova – 10,00

2ª Unidade (0,00 a 5,00):

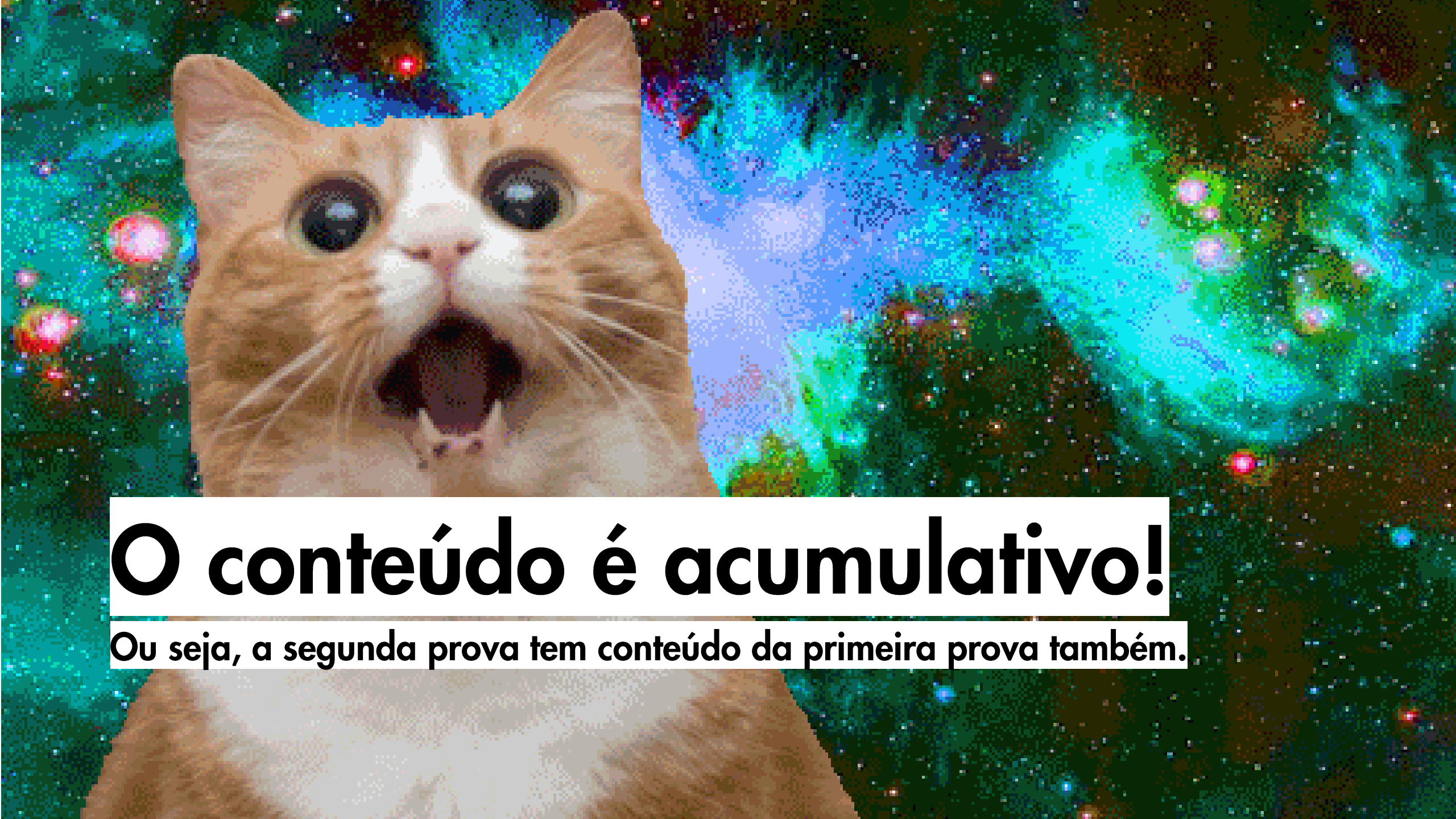
- Prova – 5,00 e Mini Projeto – 5,00

3ª Unidade (0,00 a 5,00):

- Prova – 5,00, Mini Projeto – 3,00 e Integrador 2,00

# Sugestões de Mini Projetos

- Algo com inteligência artificial
- Hospital/ clínica / anamnese
- Jogo – dama / ludo / jogo da forca / campo minado / batalha naval
- Estoque

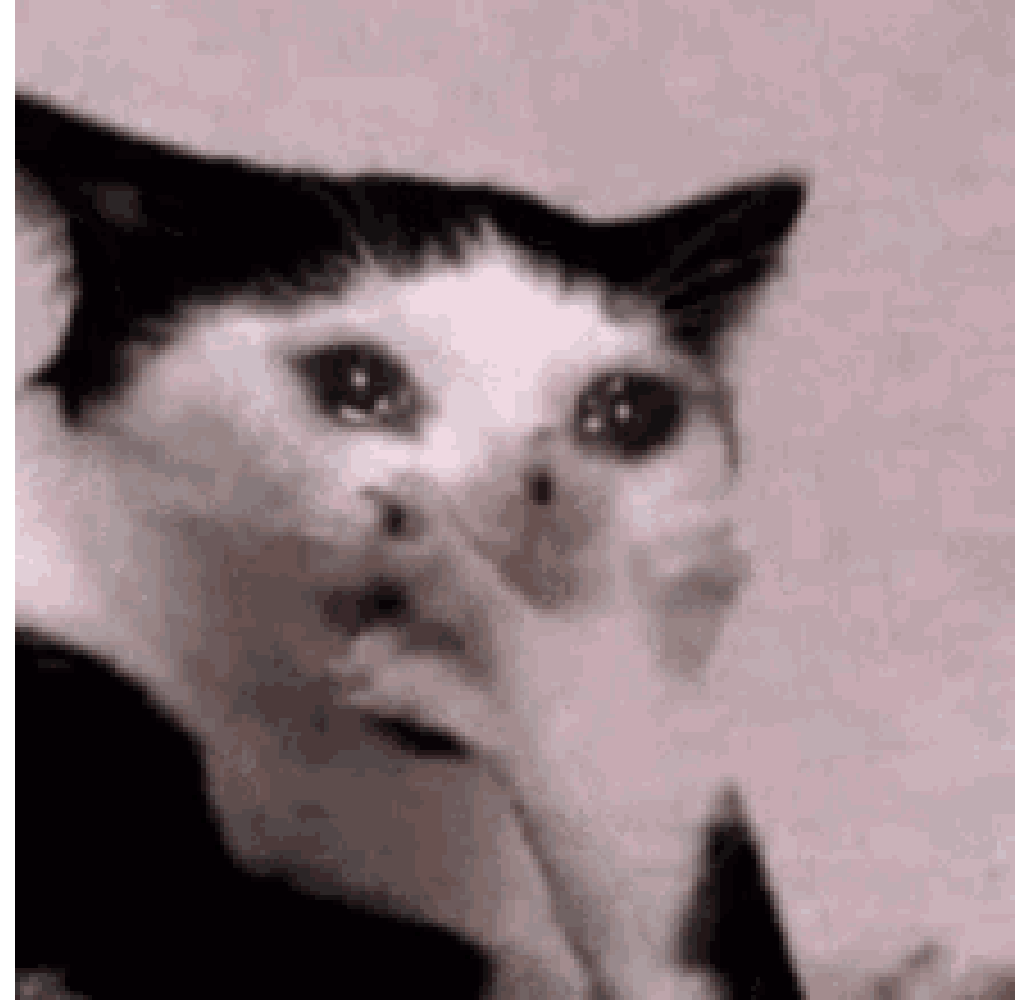


**O conteúdo é acumulativo!**

**Ou seja, a segunda prova tem conteúdo da primeira prova também.**



**Nota Final Mínima:**  
**7,00**



# Faltas Reprovam!

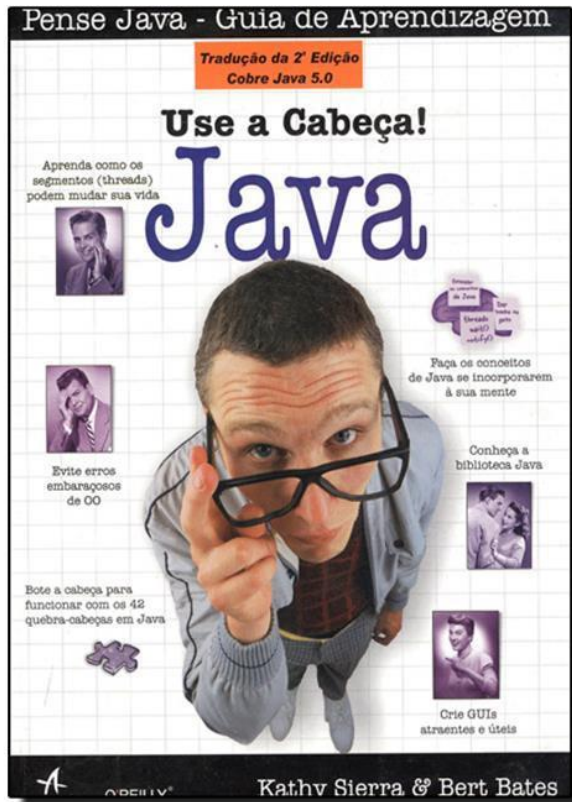
Todo estudante tem direito a faltar até 25% das aulas.  
Faremos chamada todos os dias.

# Recomendações para assistir as aulas remotas ou presenciais

- Apenas assista a aula (processo psicológico da atenção).
- Esteja presente e participe.
- Seja cordial na exposição de suas ideias.

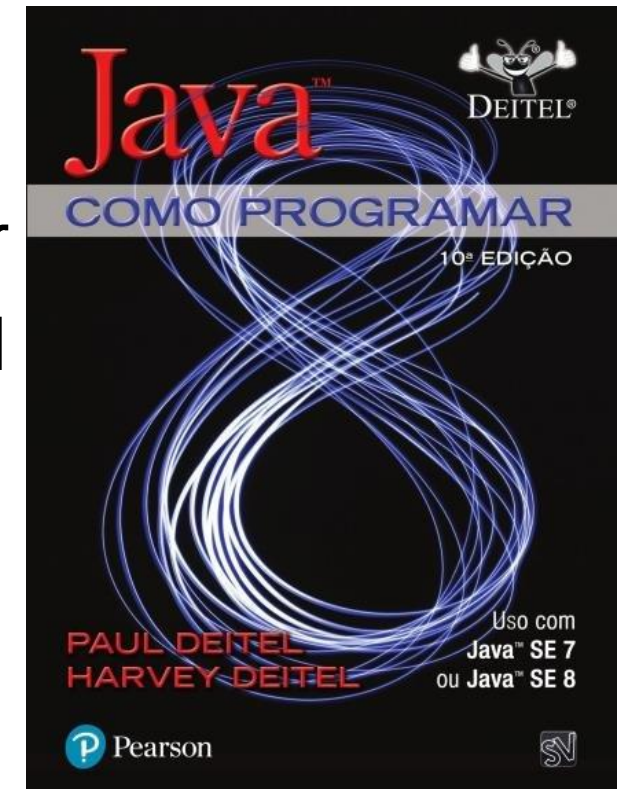


# Bibliografia



Use a Cabeça! Java  
Bert Bates e Kathy Sierra

Java - Como Programar  
Paul Deitel e Harvey Deitel





**Bora começar?**

# Ambientes de Desenvolvimento (IDEs)

IDE – Integrated Development Environment

# Ambientes de Desenvolvimento (IDEs)

- São programas de computador que reúnem características e ferramentas de apoio ao desenvolvimento de software.
- Objetivam agilizar o processo de desenvolvimento de software.

# IDEs que serão utilizados

Eclipse

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

Netbeans

<https://netbeans.org/>

Repl.it

<https://replit.com/>

**Não conseguiu instalar nenhum?**

**Use o Replit**



# As versões do Eclipse

Calisto (30 Junho 2006)  
Europa (29 Junho 2007)  
Ganymede (25 junho 2008)  
Galileo (24 Junho 2009)  
Helios (23 Junho 2010)  
Indigo (22 Junho 2011)  
Juno (27 Junho 2012)  
Kepler (26 Julho 2013)  
Luna (25 Junho 2014)  
Mars (24 Junho 2015)  
Neon (22 Junho 2016)  
Oxygen (28 Junho 2017)  
Photon (27 Junho 2018)

# **Instalando JDK e Eclipse IDE**

# Requisitos para o Desenvolvimento em Java

Serão necessários:

1. Um ambiente de desenvolvimento (IDE)
2. O Java Development Kit (JDK)
  - Fornece o compilador e outras ferramentas.
  - Fornece uma biblioteca de classes pré-construídas.
3. O Java Runtime Environment (JRE)
  - Inclui uma Máquina Virtual Java (JVM), bibliotecas de código e componentes necessários para executar programas Java.
  - Está incluído no JDK, ou seja, só precisa do JDK!

# Primeiro instale o JDK

<https://www.oracle.com/br/java/technologies/javase-jdk16-downloads.html>

| ORACLE                                                                                             |           |                                                                                                                         |
|----------------------------------------------------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------|
| Produtos Setores Recursos Suporte Eventos Desenvolvedor                                            |           |                                                                                                                         |
| Exibir Contas Fale com um consultor Oracle                                                         |           |                                                                                                                         |
| Java SE Development Kit 16.0.2                                                                     |           |                                                                                                                         |
| 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                                                                           | 144.87 MB |  jdk-16.0.2_linux-aarch64_bin.rpm    |
| Linux ARM 64 Compressed Archive                                                                    | 160.73 MB |  jdk-16.0.2_linux-aarch64_bin.tar.gz |
| Linux x64 Debian Package                                                                           | 146.17 MB |  jdk-16.0.2_linux-x64_bin.deb        |
| Linux x64 RPM Package                                                                              | 153.01 MB |  jdk-16.0.2_linux-x64_bin.rpm        |
| Linux x64 Compressed Archive                                                                       | 170.04 MB |  jdk-16.0.2_linux-x64_bin.tar.gz    |
| macOS Installer                                                                                    | 166.6 MB  |  jdk-16.0.2_osx-x64_bin.dmg        |
| macOS Compressed Archive                                                                           | 167.21 MB |  jdk-16.0.2_osx-x64_bin.tar.gz     |
| Windows x64 Installer                                                                              | 150.58 MB |  jdk-16.0.2_windows-x64_bin.exe    |
| Windows x64 Compressed Archive                                                                     | 168.8 MB  |  jdk-16.0.2_windows-x64_bin.zip    |

# Vídeo Aula

Download e teste do JDK:

<https://youtu.be/EGQ5bkpaRzg>



# Eclipse IDE

- É um ambiente de desenvolvimento (open source) para o criação de software em Java (e outras linguagens).
- Projeto iniciado pela IBM e doado para a comunidade software livre.

# Após instalar o JDK, baixe e instale o Eclipse

Download do Eclipse em <https://www.eclipse.org/downloads/>

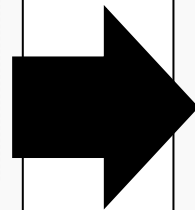
The Eclipse Installer 2020-12 R now includes a JRE for macOS, Windows and Linux.



Get **Eclipse IDE 2020-12**  
Install your favorite desktop IDE packages.

[Download x86\\_64](#)

[Download Packages](#) | [Need Help?](#)



All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

[Download](#)

Download from: Brazil - C3SL - Federal University of Parana ([https](https://www.c3sl.ufpr.br))

File: [eclipse-inst-jre-win64.exe](#) SHA-512

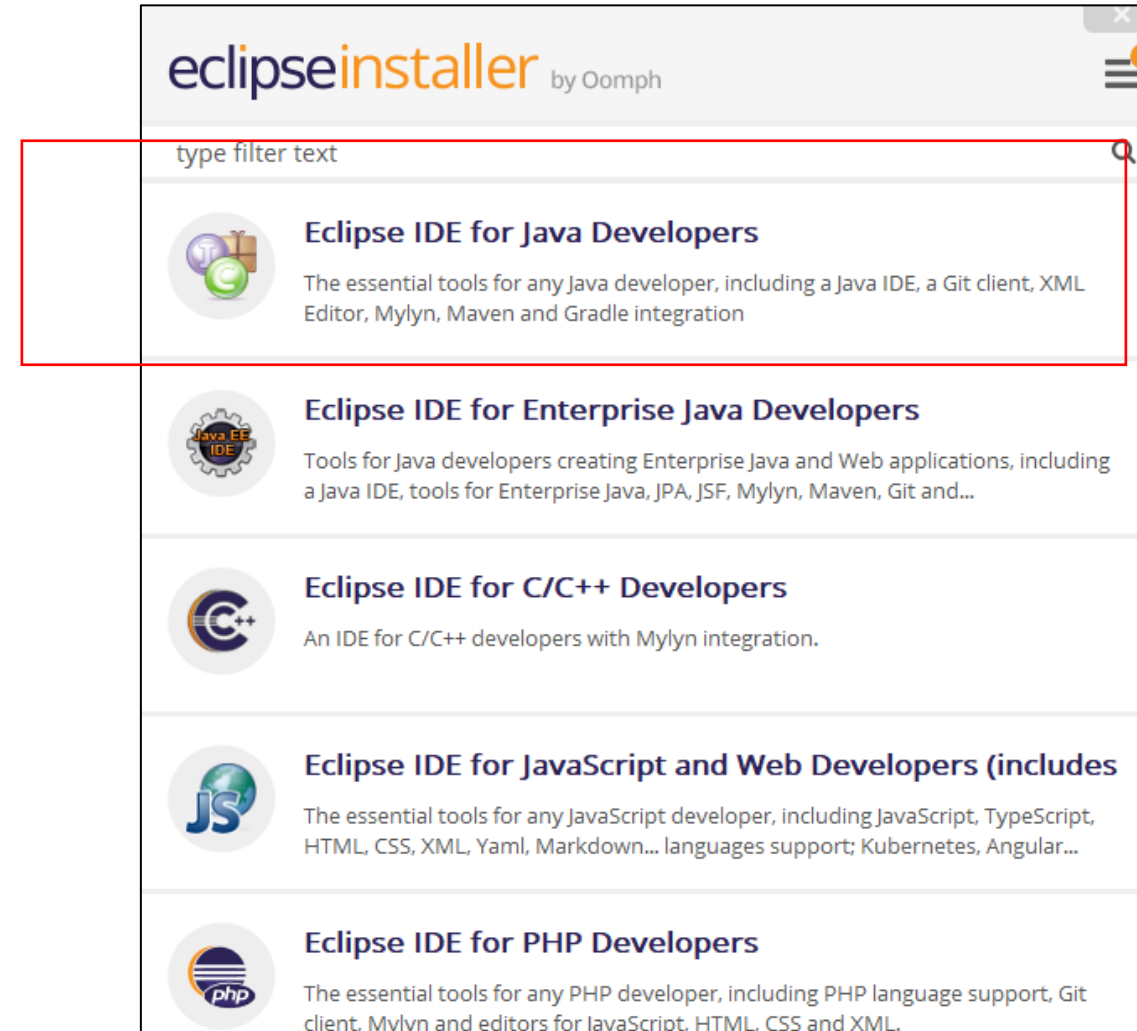
[>> Select Another Mirror](#)

# Versões do Eclipse

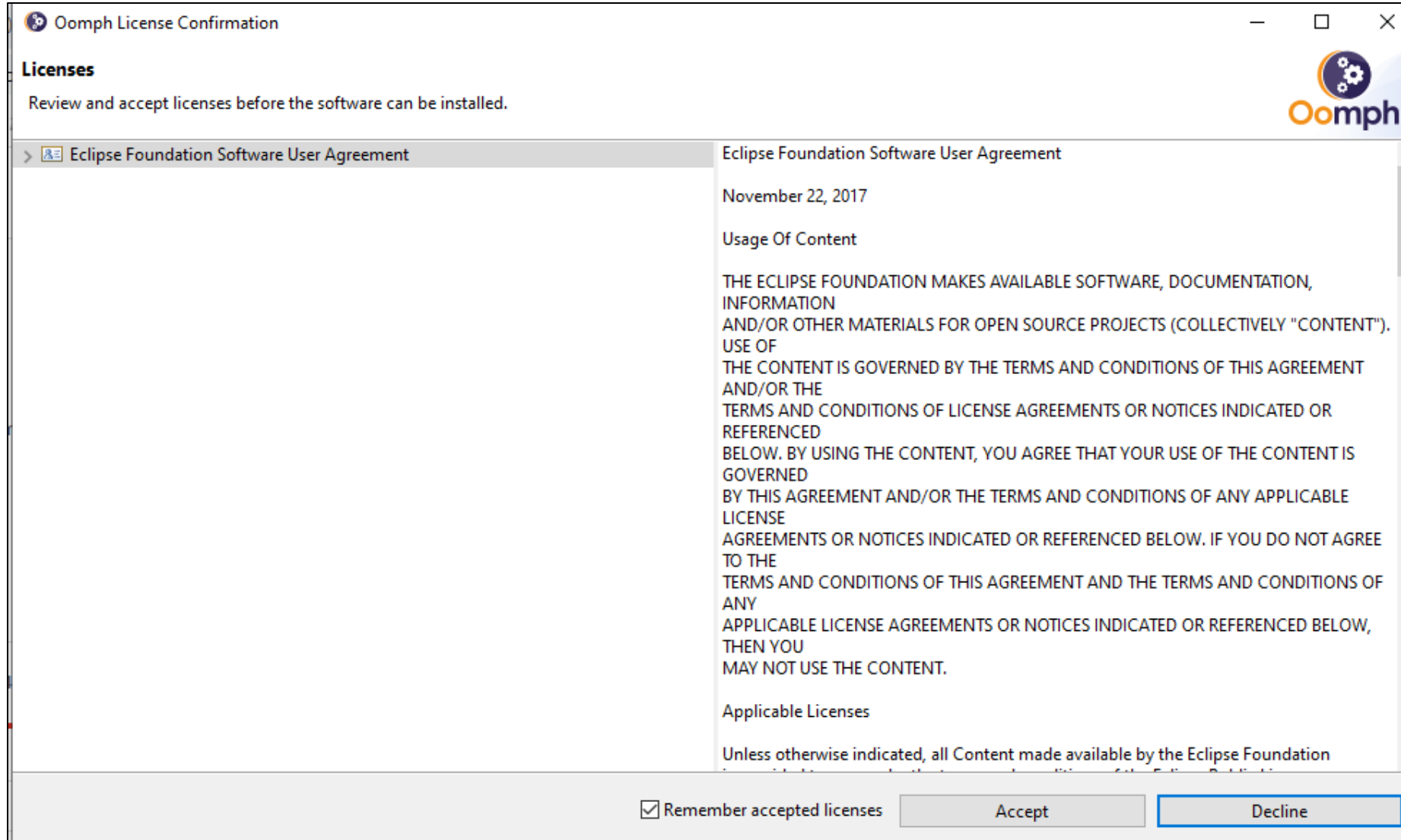
- Mais no início da história do Java, tínhamos 4 versões: SE, EE, ME e Java TV).
- Atualmente as versões que ainda permanecem com o mesmo nome são a EE (Enterprise Edition) e a SE (Standard Edition).
- A versão EE deve ser usada para desenvolver Java Web.
- Nesta disciplina, iremos utilizar apenas os recursos de uma versão SE.
- No entanto, na instalação IDE do Eclipse, essa nomenclatura é um pouco diferente.

# Qual opção escolher?

- Após abrir o instalador, escolha a opção “eclipse IDE for Java developers”.
- Clique em “accept” para todas as mensagens que aparecerem.

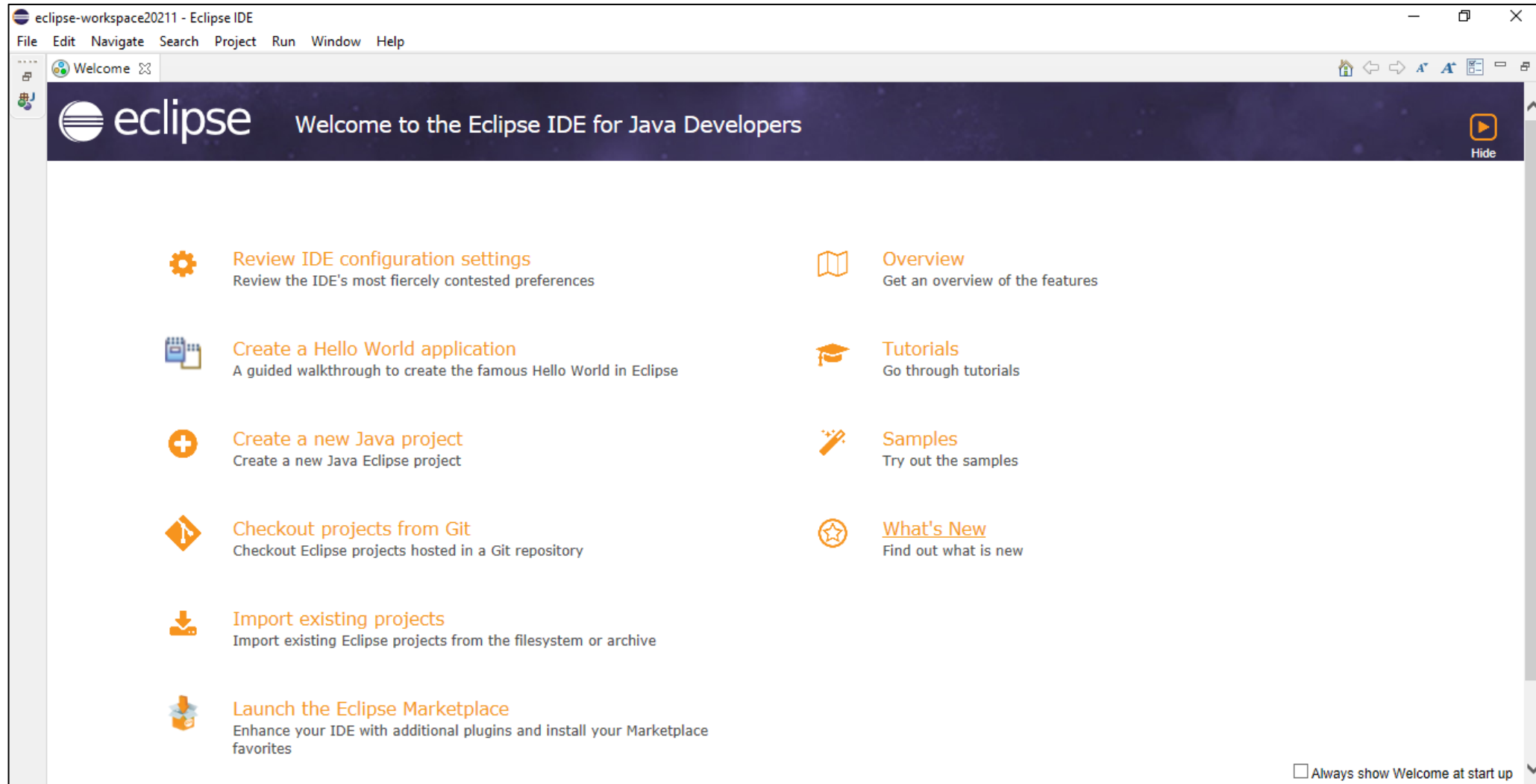


# Aceite as licenças...

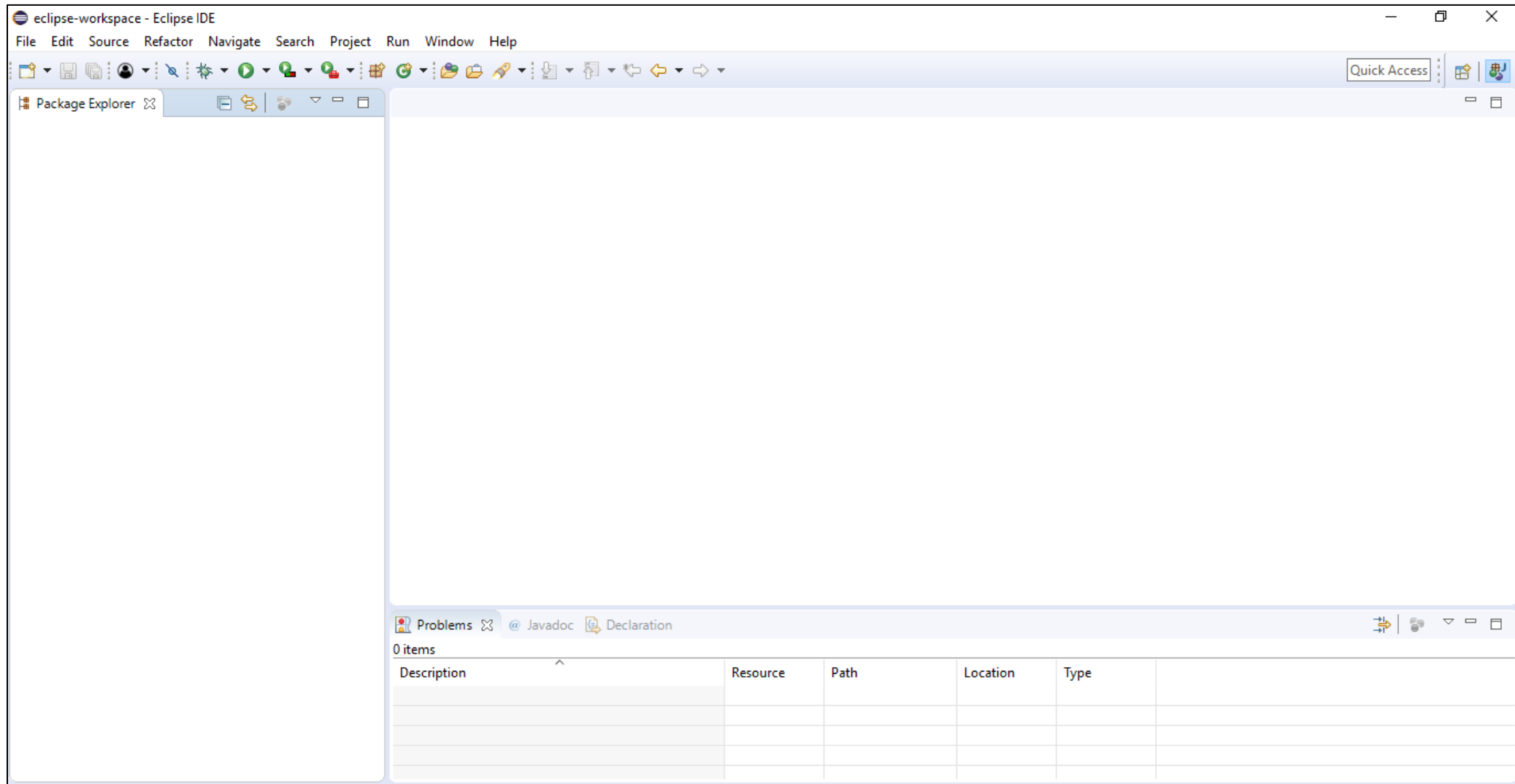




# Tela Inicial do Eclipse



# Tela Inicial do Eclipse



# **Intervalo: 15 minutos**

Baixem o eclipse!



## **As polêmicas que rondam o Java**

<https://www.youtube.com/watch?v=oOZZJ3HCMKs>

# Paradigmas de Programação

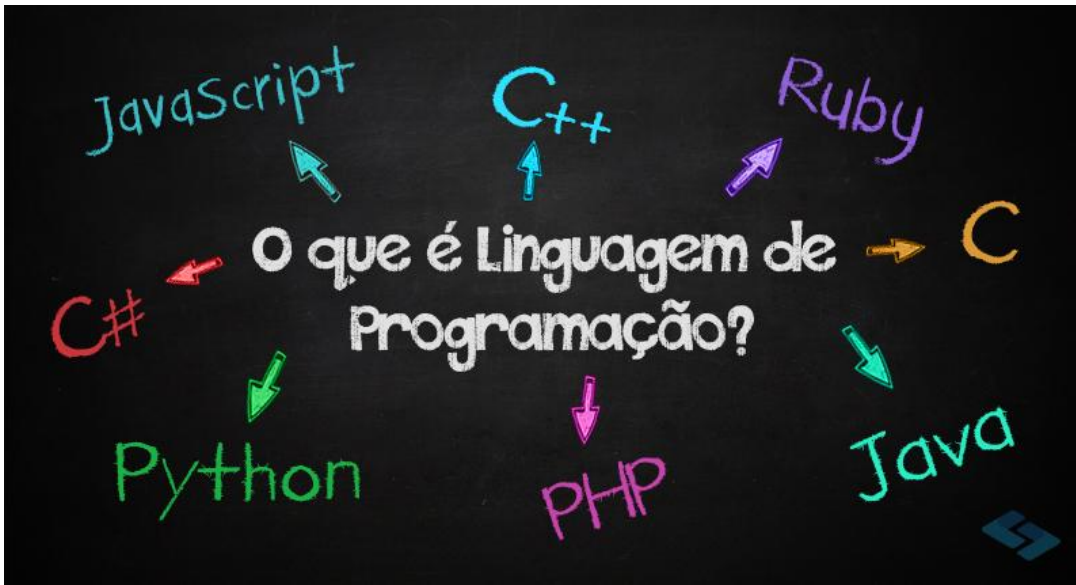
# O que é um Paradigma de Programação?

Um paradigma é um modelo, um padrão, um estilo.



# Paradigmas de Programação

- Uma Linguagem de Programação (LP) é a notação formal para descrição de algoritmos em um computador.
- É necessário entender as organizações das LP, com ênfase na compreensão dos seus conceitos abstratos e dos seus elementos de representação.



# Paradigmas de Programação

- A LP pode se relacionar com um método (estilo, padrão) de projeto específico.
- A linguagem de programação pode forçar um certo **estilo de programação**.
- Esses estilos são chamados de **paradigmas de programação**.

# Principais Paradigmas de Programação

- Programação Procedural
- Programação **Orientado a Objetos**
- Mas existem outros: programação orientada a restrições, programação orientada a aspectos, programação imperativa...

# Paradigmas de Programação

## Procedural (Estruturado)

- Estilo de programação convencional (descrita no aprendizado em algoritmos).
- Programas são decompostos em “passos” de processamento.
- Rotinas são usadas na modularização de passos específicos.
- Ex.: Pascal e C.
- Palavras-chave: Procedimentos e Funções.

# Paradigmas de Programação Orientado a Objetos

- Enfatiza a definição de classes de objetos.
- Objetos são estruturas de dados contendo lógica.
- Objetos interagem a partir da troca de mensagens.
- Ex.: Simula (década de 1960), Smalltalk (década de 1970), Eiffel (1985) e **Java** (1995). Embora Simula tenha introduzido os conceitos básicos de orientação a objetos, Smalltalk é frequentemente creditada por popularizar esses conceitos e por ter uma influência significativa no desenvolvimento de linguagens orientadas a objetos subsequentes, como Java e Python.
- Palavras-chave: Classes, Objetos e Métodos.

# Exercício

Um paradigma de programação fornece e determina a visão que o programador possui sobre a estruturação e execução do programa. Acerca do paradigma de programação orientado a objeto, analise as asserções a seguir:

I. É um estilo de programação **convencional** (descrita no aprendizado em algoritmos). Os programas são decompostos em “passos” de processamento e são utilizadas rotinas para modularização de passos específicos;

II. É um paradigma de programação que se refere ao **uso de restrições** na construção de relações entre variáveis. Consiste em especificar, para uma solução, que critérios (restrições) esta tem de cumprir.

III. Apresenta um modelo de análise, projeto e programação de sistemas de software baseado na composição e interação entre **diversas unidades de software chamadas de objetos**.

Assinale a opção que contém apenas asserções verdadeiras.

- a) I, apenas.      b) II, apenas.      c) III, apenas.      d) I e III.      e) II e III.      f) I, II e III

# Resposta do Exercício

C



# Linguagens de Programação

- Para armazenar um algoritmo na memória de um computador, e para que o mesmo seja executável, é necessário que ele seja programado (codificado).
- Precisamos transcrever nossos algoritmos para uma linguagem de programação.



**Nesta disciplina,  
iremos aprender  
POO utilizando a  
linguagem Java!**

# História do Java

Em 1991, na Sun Microsystems, foi iniciado o Green Project, o berço do Java, uma linguagem de programação orientada a objetos. Os mentores do projeto eram Patrick Naughton, Mike Sheridan, e James Gosling. Eles acreditavam que, eventualmente, haveria uma convergência dos computadores com os equipamentos e eletrodomésticos frequentemente usados pelas pessoas no seu dia-a-dia.

Para provar a viabilidade desta ideia, 13 pessoas trabalharam arduamente durante 18 meses. No verão de 1992 eles emergiram de um escritório de Sand Hill Road, no Menlo Park, com uma demonstração funcional da ideia inicial. O protótipo se chamava \*7 (lê-se “Star Seven”), um controle remoto com uma interface gráfica touchscreen. Para o \*7, foi criado um mascote, hoje amplamente conhecido no mundo Java, o Duke. O trabalho do Duke no \*7 era ser um guia virtual ajudando e ensinando o usuário a utilizar o equipamento. O \*7 tinha a habilidade de controlar diversos dispositivos e aplicações. James Gosling especificou uma nova linguagem de programação para o \*7. Gosling decidiu batizá-la de “Oak”, que quer dizer *carvalho*, uma árvore que ele podia observar quando olhava através da sua janela.

O próximo passo era encontrar um mercado para o starseven. A equipe achava que uma boa ideia seria controlar televisões e vídeo por demanda com o equipamento. Eles construíram uma demonstração chamada de *MovieWood*, mas infelizmente era muito cedo para que o vídeo por demanda bem como as empresas de TV a cabo pudessem viabilizar o negócio. A ideia que o \*7 tentava vender, hoje já é realidade em programas interativos e também na televisão digital. Permitir ao telespectador interagir com a emissora e com a programação em uma grande rede de cabos, era algo muito visionário e estava muito longe

do que as empresas de TV a cabo tinham capacidade de entender e comprar. A ideia certa, na época errada.

Entretanto, o estouro da internet aconteceu e rapidamente uma grande rede interativa estava se estabelecendo. Era este tipo de rede interativa que a equipe do \*7 estava tentando vender para as empresas de TV a cabo. E, da noite para o dia, não era mais necessário construir a infraestrutura para a rede, ela simplesmente estava lá. Gosling foi incumbido de adaptar o Oak para a internet e em janeiro 1995 foi lançada uma nova versão do Oak que foi rebatizada para *Java* — diz-se que inspirado no café que o time de desenvolvimento consumia, oriundo da ilha de Java, e que também está presente na logomarca *Java*.





# A linguagem Java

- Em 1991, A Sun Microsystems financiou uma pesquisa corporativa interna com codinome **Green** (berço do Java).
- O projeto resultou no desenvolvimento de uma linguagem baseada em C e C++, chamada inicialmente de **Oak** (carvalho).
- Só que infelizmente o nome Oak já pertencia a uma outra linguagem de programação.



# Algumas Curiosidades



- Os criadores consumiam muito café enquanto desenvolviam. Então a equipe da Sun foi visitar uma cafeteria local para pensar num novo nome.
- Lá eles descobriram que o café que consumiam era produzido na ilha de Java na Indonésia, fato que culminou com a escolha do nome.
- Foi nesse momento que o nome Java foi sugerido e pegou.

# James Gosling – O “pai” do Java



- James Gosling é um cientista da computação canadense, amplamente reconhecido como o criador da linguagem de programação Java.
- Em 1991, enquanto trabalhava na Sun Microsystems, liderou o projeto Green, que inicialmente buscava desenvolver software para dispositivos eletrônicos domésticos.
- A necessidade de uma linguagem portátil, segura e independente de plataforma levou Gosling e sua equipe a criar o Java, lançado oficialmente em 1995.
- Com sua filosofia “Write Once, Run Anywhere” (Escreva uma vez, execute em qualquer lugar), o Java revolucionou o desenvolvimento de software, tornando-se uma das linguagens mais utilizadas no mundo. Gosling recebeu diversos prêmios e continua sendo uma figura influente na área de tecnologia.



# A Oracle compra a Sun

20 de Abril de 2009

A Oracle compra Sun Microsystems (num lance de **7,4 bilhões** de dólares) e agora é dona do Java, OpenOffice e MySQL.

<http://g1.globo.com/Noticias/Tecnologia/0,,MUL1091457-6174,00-ORACLE+ANUNCIA+COMPRA+DA+SUN+POR+MAIS+DE+US+BILHÕES.html>

BUY 



\$7,4 bilhões



# A Linguagem Java

- Utiliza o paradigma de programação orientada a objetos (POO).
- Programas em Java consistem em partes chamadas **classes**.
- Classes definem os **objetos** presentes no sistema de software.
- Cada classe vai determinar:
  - O **comportamento** de seus objetos (instruções e métodos).
  - Os **estados** possíveis dos objetos (atributos).
  - O **relacionamento** entre objetos.

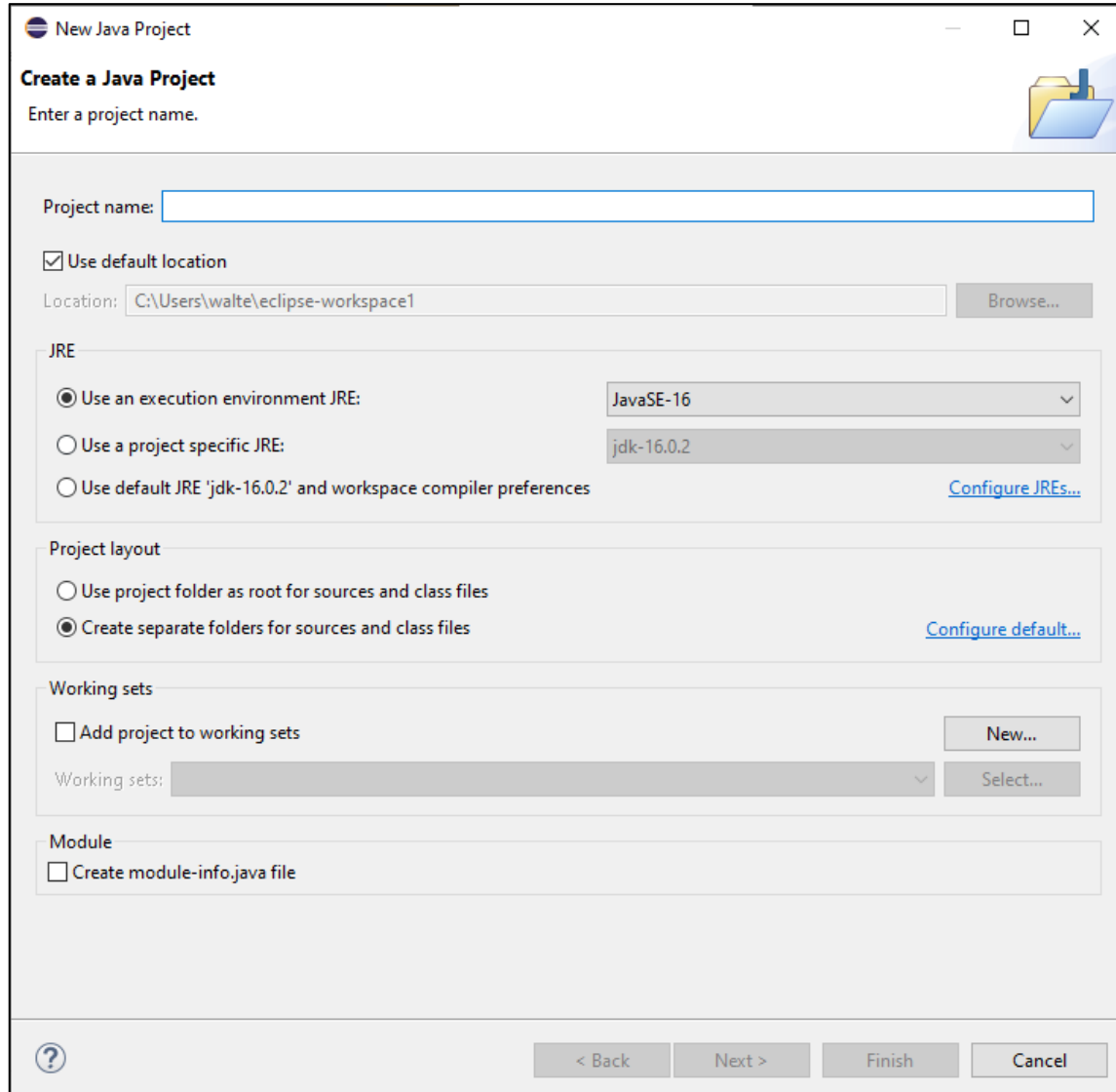
# Abram o





# **Criando um Projeto no Eclipse**

# Criando um Projeto no Eclipse



The screenshot shows the 'New Java Project' dialog box in Eclipse. The title bar says 'New Java Project'. Below the title bar, it says 'Create a Java Project' and 'Enter a project name.' There is a folder icon on the right. The 'Project name:' field is empty. The 'Use default location' checkbox is checked. The 'Location:' field shows 'C:\Users\walte\eclipse-workspace1' with a 'Browse...' button. The 'JRE' section has three radio buttons: 'Use an execution environment JRE:' (selected), 'Use a project specific JRE:', and 'Use default JRE 'jdk-16.0.2' and workspace compiler preferences'. The first option has a dropdown menu showing 'JavaSE-16'. The second option has a dropdown menu showing 'jdk-16.0.2'. There is a 'Configure JREs...' link. The 'Project layout' section has two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected). There is a 'Configure default...' link. The 'Working sets' section has a checkbox 'Add project to working sets' and a 'New...' button. The 'Working sets:' field is empty with a dropdown arrow. There is a 'Select...' button. The 'Module' section has a checkbox 'Create module-info.java file'. At the bottom, there are buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Com o software iniciado, clique na opção de menu:

**File > New > Java Project;**

Deixe **desmarcada** a opção Create module-info.java file

# Recurso de Modularidade

- A modularidade chegou como uma das novidades do Java 9, no entanto, não precisa se preocupar com isso agora. Apenas opte por **não** criar o `module-info.java`
- As ferramentas de linha de comando da plataforma Java são capazes de criar, manipular e consumir arquivos JAR. Diante disso, de forma a facilitar a adoção e migração para a nova versão do JDK, os desenvolvedores criaram também o arquivo JAR modular.
- Assim, para fazer uso de um arquivo JAR modular como se fosse um JAR comum, basta informar o mesmo no classpath da aplicação. Dessa forma o arquivo `module-info.class` será ignorado.
- Essa estratégia permite que aplicações legadas, que executam em versões anteriores do Java, possam se beneficiar de bibliotecas desenvolvidas de forma modular, **mantendo com isso a retrocompatibilidade**.

# Criando um Projeto no Eclipse

- Em **Project Name**, defina um nome para seu projeto.
- Sugestão de nome: **MeuProjetoJava**.
- Clique em **Finish**.



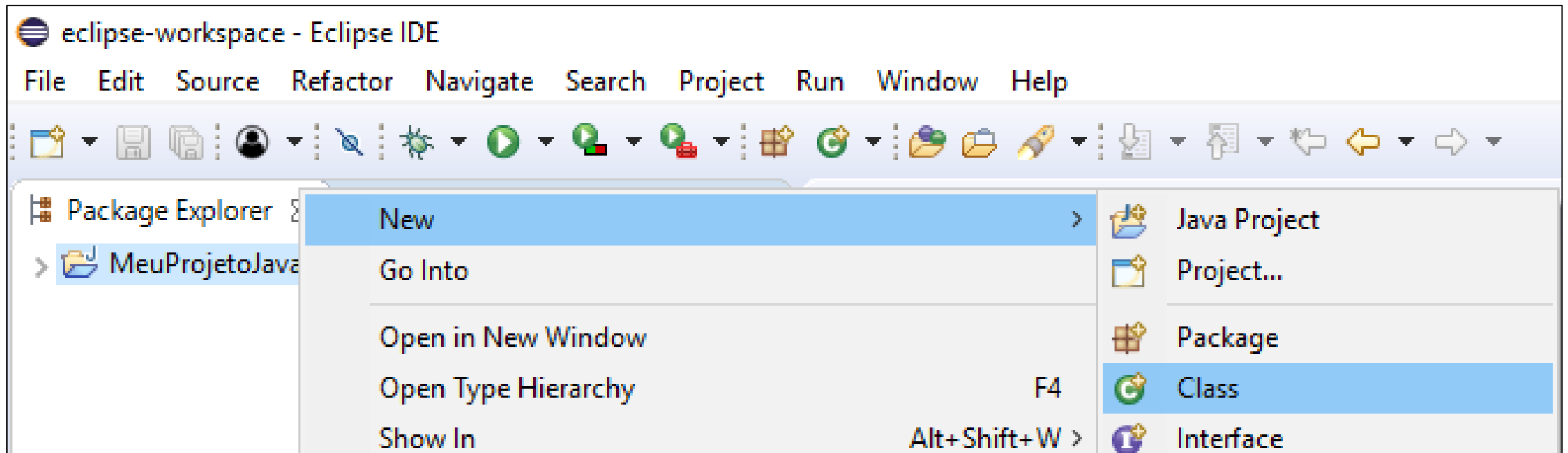
**Após criar um projeto, é preciso criar uma classe. Iremos escrever os programas dentro das classes.**

# **Criando uma classe Java**



# Criando uma classe Java

- 1 - No painel Package Explorer (lado esquerdo da janela principal da IDE), clique com o botão direito do mouse sobre o projeto que você acabou de criar.
- 2 - No menu apresentado, selecione a opção New > Class.



# Criando uma Classe

3. No campo **Name** defina o nome da sua classe (sem espaços e caracteres especiais).

4. A primeira letra do nome de sua classe deve ser maiúscula. Exemplos de nomes que podem ser utilizados: **Principal**, **Aplicação**, **App**.

5. Marque a opção *public static void main(String args[])* em *Which method stubs would you like to create?*

6. Clique em **Finish**;

**New Java Class**

**Java Class**

⚠ The use of the default package is discouraged.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

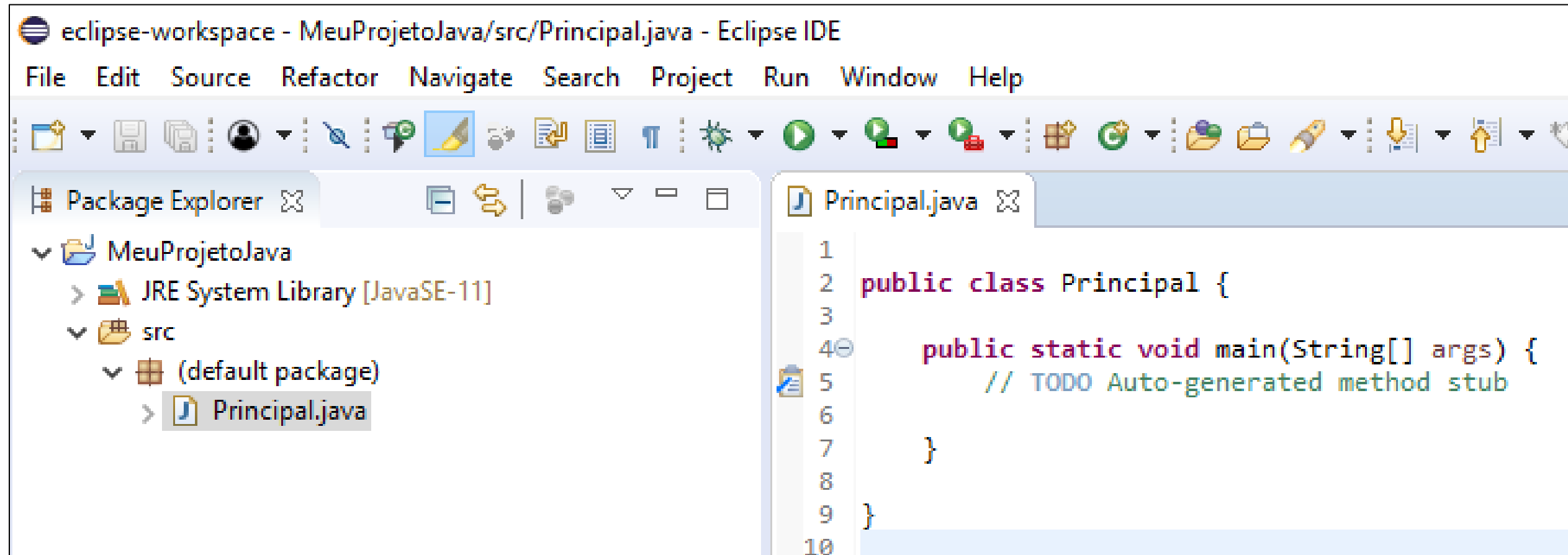
☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

# Resultado



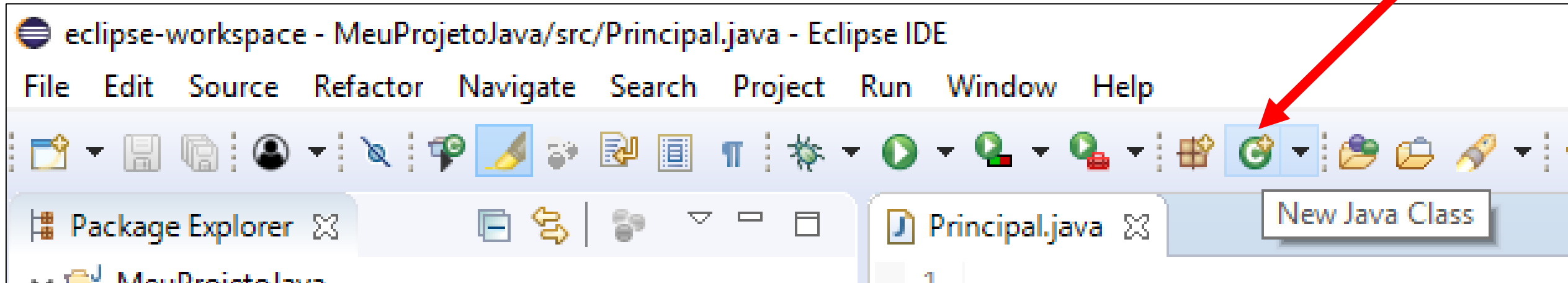
**As classes possuem métodos.  
O método principal de um  
programa java é o `main`.**

# O método main

- Ao selecionarmos a opção de criar o método `public static void main (String[] args)` indicamos a criação do método que servirá de ponto de partida para a execução do programa Java.
- É possível criar várias classes, no entanto, cada programa java **deve ter apenas uma única classe contendo o método main!**
- Caso o projeto tenha mais de um método `main`, o programa pode até compilar, mas em algum momento haverá problemas de execução.

# Botão de Atalho para criar uma classe

- Clique uma única vez no projeto (apenas para selecioná-lo) e clique no botão com a letra C no menu.
- Irá te levar para a janela de **new class**.

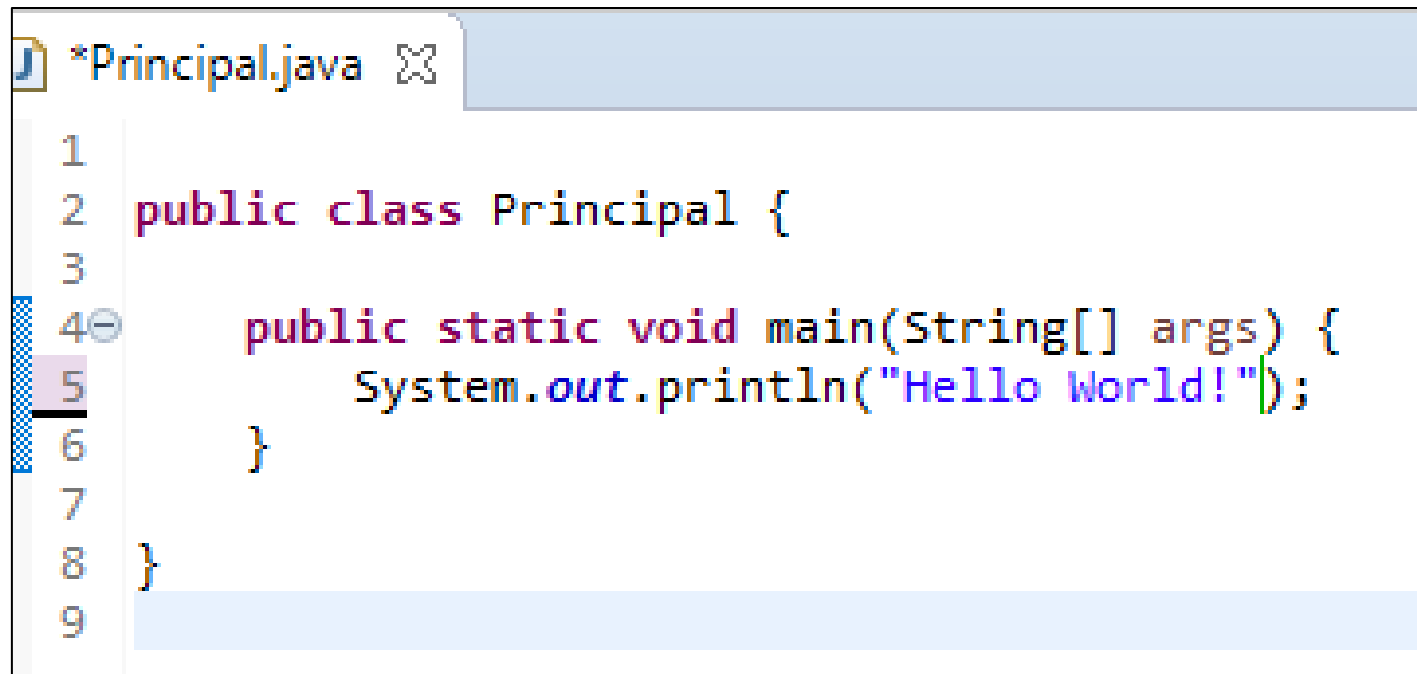




**Criando um Hello World!**

# Criando um Hello World!

- `System.out.println()` é o método de saída padrão para o console (equivalente a função escreva em algoritmos).
- Escreva o valor do método entre aspas.

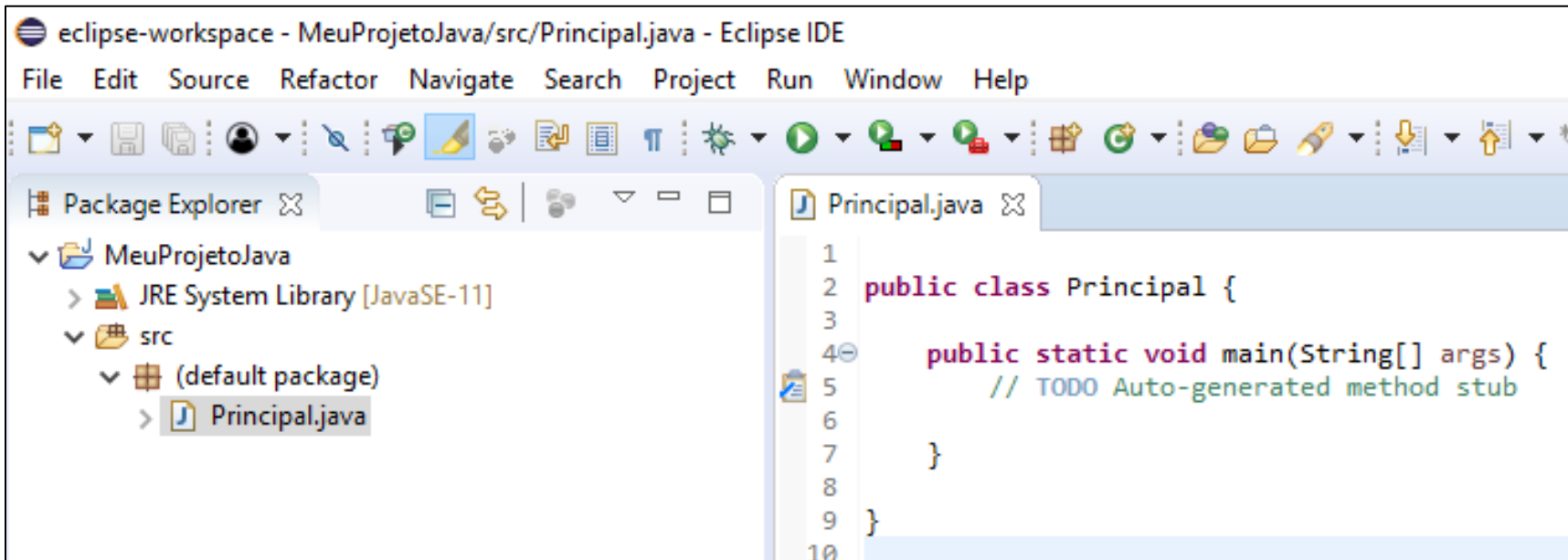


The screenshot shows a code editor window with a tab labeled `*Principal.java`. The code is as follows:

```
1
2 public class Principal {
3
4     public static void main(String[] args) {
5         System.out.println("Hello World!");
6     }
7
8 }
9
```

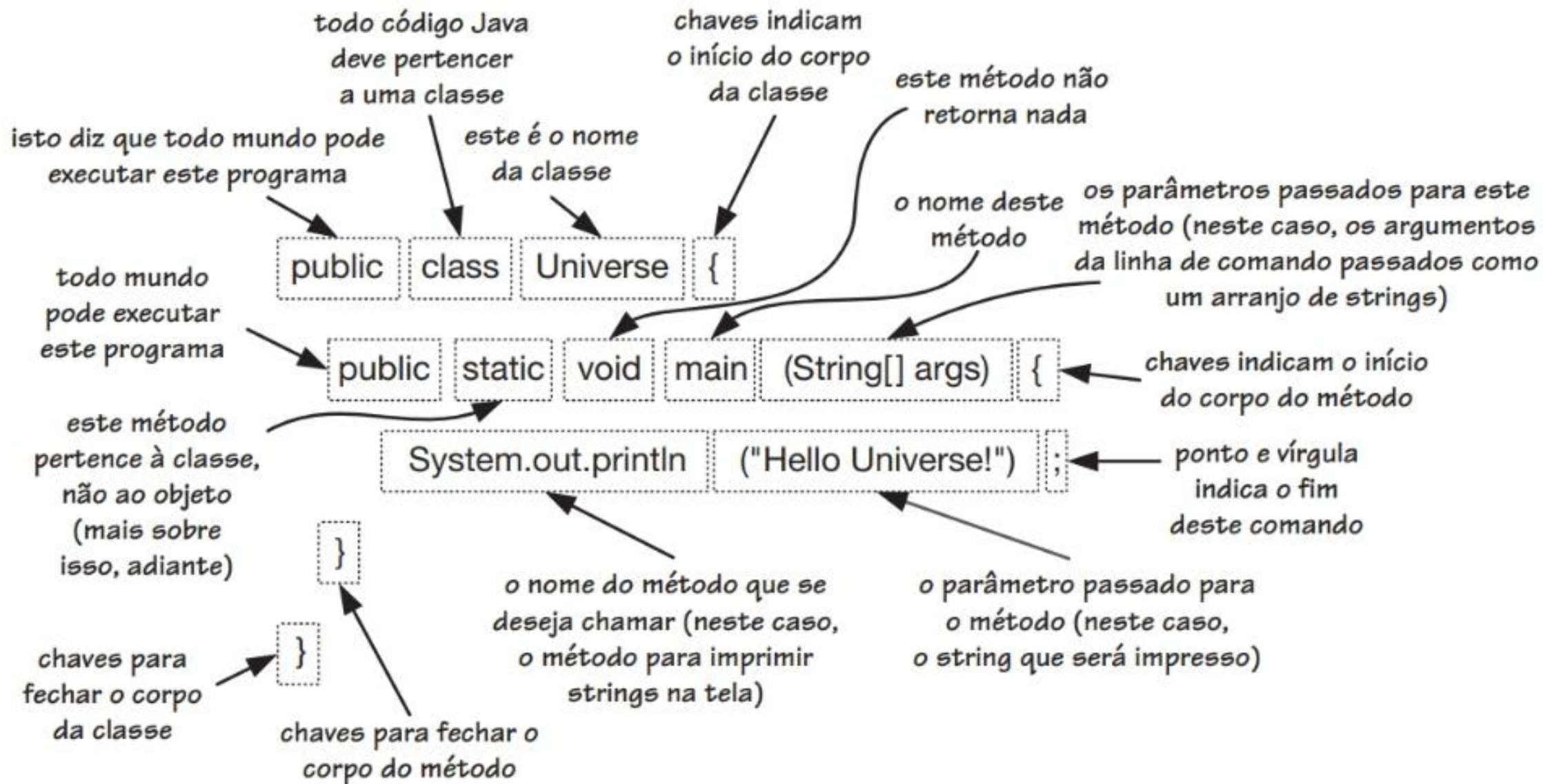
# Criando um Hello World!

- Apague os comentários desnecessários.
- `//` é o comentário de linha.
- `/*` e `*/` indicam o início e fim do comentário.



## ATALHOS

- Use o atalho **control + D** para apagar uma linha de código.
- Use o atalho **control + shift + C** para transformar o texto selecionado em comentário ou **control + /**

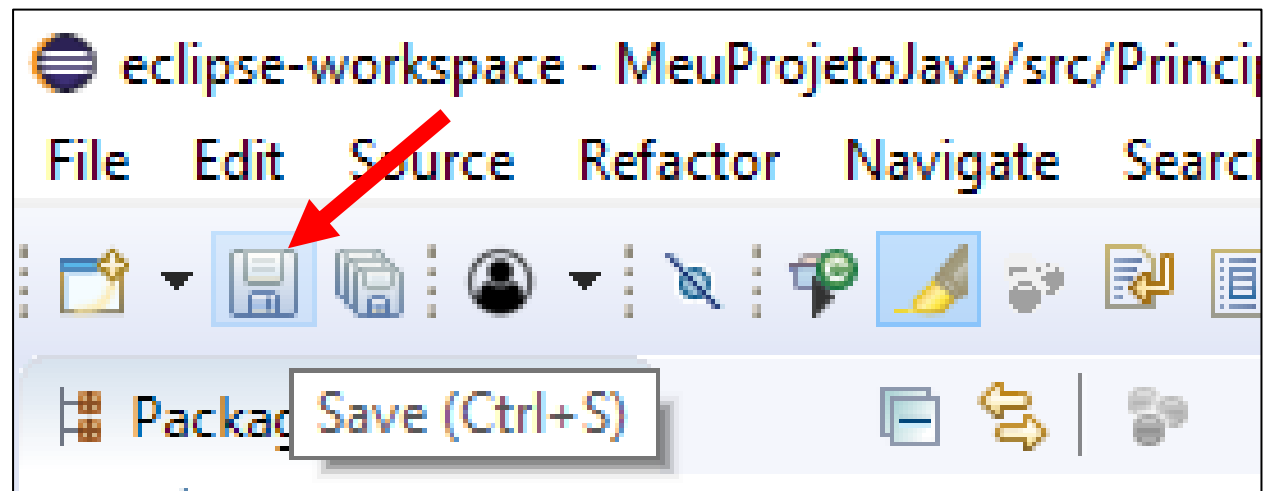


**Figura 1.1** O programa “Hello Universe!”.

# **Testando sua aplicação Java no Eclipse IDE**

# Testando sua aplicação Java no Eclipse IDE

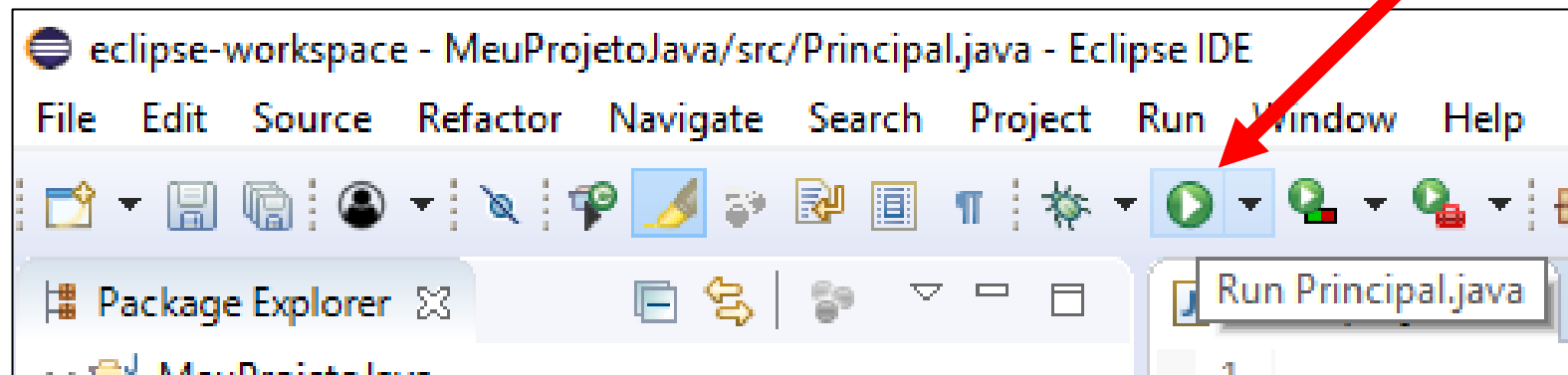
- Primeiro salve a aplicação apertando o botão de Salvar no menu ou com o atalho **control + s**.
- Atente que existe logo ao lado um botão para “salvar tudo”.
- Caso tenham sido realizadas modificações em várias classes do projeto, use esta opção.



# Testando sua aplicação Java no Eclipse IDE

Em seguida é preciso compilar. Existem 3 formas para realizar esta ação:

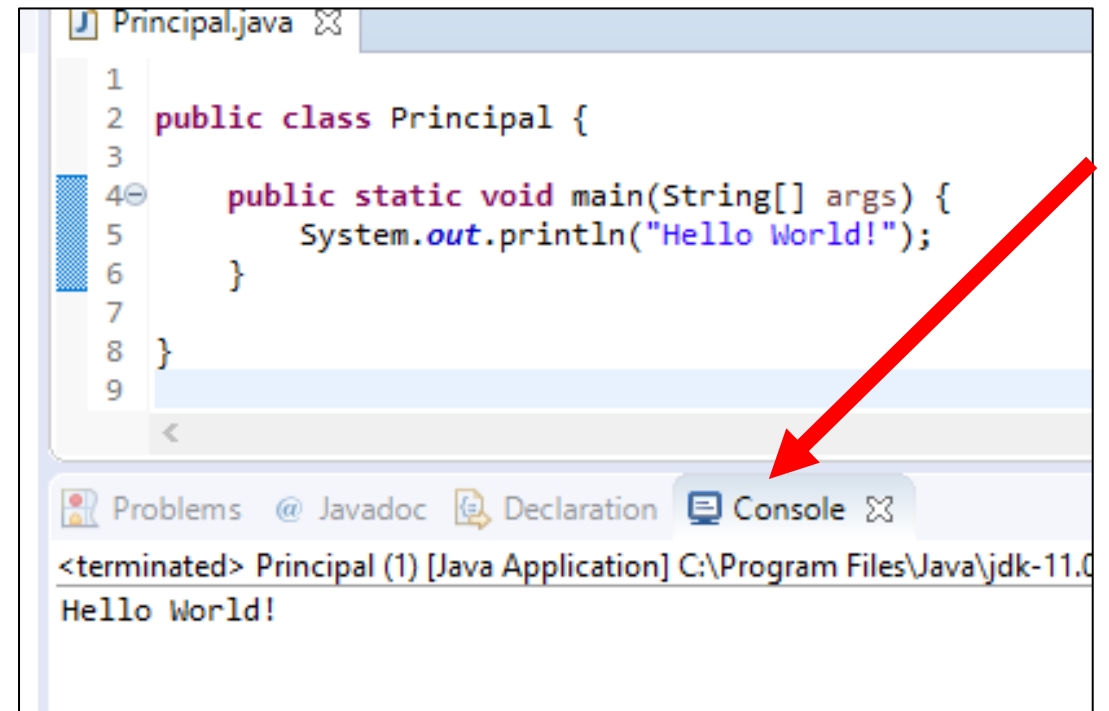
1. Atalho control + F11.
2. No Package Explorer, clique com o botão direito do mouse sobre o projeto java e selecione a opção **Run As > Java Application**.
3. Aperte o botão **Run** no menu.





# Testando sua aplicação Java no Eclipse IDE

- Sua aplicação será executada na parte inferior da janela principal do Eclipse (Aba Console).
- Neste mesmo painel, podemos verificar possíveis erros de compilação e execução das nossas classes (Aba Problems).



# Método de saída de dados `System.out.println();`

Você pode exibir o conteúdo de uma variável String (atente para o S maiúsculo de String):

```
System.out.println("Hello World");
```

```
String teste = "Hello World";
```

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

# Método de saída de dados `System.out.println();`

Você pode concatenar texto e/ou variáveis usando o `+`.  
Atente para as aspas:

```
String nome = "walter";  
System.out.println("Saudações " + nome);
```

# **Lista de Exercícios 1**

# PROGRAMAÇÃO ORIENTADA A OBJETOS

## AULA 1 - APRESENTAÇÃO DA DISCIPLINA E PARADIGMAS DE LINGUAGEM

**WALTER TRAVASSOS SARINHO**

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR