

Técnicas

DE PROGRAMAÇÃO

Prof. Rafael Fernandes



O que teremos hoje?

Por que estudar Técnicas de Programação?
Introdução às IDEs e aos Editores de Código
Linguagem de Programação JAVA
"Hello World" em JAVA



Por que estudar TP?

Além do conhecimento de Algoritmos é fundamental ao desenvolvedor de sistemas saber utilizar as mais diversas **técnicas de programação**, para ter condições de *escolher* entre as melhores alternativas de mercado e **aplicar as técnicas nas linguagens de programação** existentes.



Por que estudar TP?

Desta forma, a partir de certos requisitos definidos pelos *stakeholders* pode-se estabelecer um **direcionamento** e as opções **adequadas** de desenvolvimento.



IDEs (Ambiente de Desenvolvimento Integrado)

São ferramentas necessárias à elaboração, compilação e execução de aplicações.

Para desenvolvimento em JAVA, as principais são: Eclipse e Netbeans.

Nada mais são que ambientes que proporcionam a execução de tarefas mais produtivamente.

São como editores de texto com muito mais recursos.

IDES





Editores de código-fonte

São alternativas mais leves que as IDEs e também permitem a edição de códigos para nossas aplicações.

EDITORES DE TEXTO



Algumas alternativas são:
ATOM, Sublime, Notepad++,
Visual Studio Code, etc.



Linguagem de Programação JAVA

Java é uma **linguagem de programação orientada a objetos** desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa **Sun Microsystems**. Em 2008 o Java foi adquirido pela empresa Oracle Corporation. Diferente das linguagens de programação convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um *bytecode* que é interpretado por uma **máquina virtual** (Java Virtual Machine, mais conhecida pela sua abreviação JVM).



Principais características da linguagem

- Portabilidade
 - Uma mesma aplicação pode ser executada em diferentes plataformas (*hardware* e *software*) sem a necessidade de adaptação de código.
- *Multithreading*
 - Possibilidade de execução de diferentes processos simultaneamente.
- Suporte a comunicação
 - Oferece um conjunto de classes para desenvolvimentos de aplicações rodando em rede.



Principais características da linguagem

- Orientação a objetos:
 - Técnica de programação que modela componentes de *softwares* em termos de objetos do mundo real.
- Vantagens:
 - Modularidade
 - Reusabilidade
 - Produtividade
 - Facilidade de manutenção e expansão



Ambientes de desenvolvimento Java

- JSE (Java Standard Edition)
 - Seu uso é voltado a PCs e servidores.
 - Contém todo o ambiente necessário para a criação e execução de aplicações desktop e web de pequeno e médio porte.
 - Pode-se dizer que essa é a plataforma principal, já que, o JEE e o JME tem sua base aqui.



Ambientes de desenvolvimento Java

- JEE (Java Enterprise Edition)
 - Voltada para o desenvolvimento de softwares corporativos.
 - Baseados em componentes que são executados em um servidor de aplicação.
- JME (Java Micro Edition)
 - Ambiente de desenvolvimento para dispositivos móveis ou portáteis, como telefones celulares e palmtops.



Ambientes de desenvolvimento Java

Distribuições para baixar do NetBeans IDE

Tecnologias suportadas *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	Tudo
① SDK da plataforma NetBeans	•	•				•
① Java SE	•	•				•
① Java FX	•	•				•
① Java EE		•				•
① Java ME						•
① HTML5/JavaScript		•	•	•		•
① PHP			•	•		•
① C/C++					•	•
① Groovy						•
① Java Card(tm) 3 Connected						•
Servidores embutidos						
① GlassFish Server Open Source Edition 4.1.1		•				•
① Apache Tomcat 8.0.27		•				•
	Download	Download	Download x86 Download x64	Download x86 Download x64	Download x86 Download x64	Download
	95 MB livre(s)	197 MB livre(s)	108 - 112 MB livre(s)	108 - 112 MB livre(s)	107 - 110 MB livre(s)	221 MB livre(s)



JAVA - Componentes básicos

- JRE (Java Runtime Environment)
 - Significa Ambiente de Tempo de Execução
- É um pacote de softwares, que é executado como um aplicativo do sistema operacional e que interpreta a execução de programas Java
 - A JRE é composta pela JVM somada ao conjunto de API's.
 - JVM + API's = JRE



JAVA - Componentes básicos

- API (Application Programming Interface)
 - Significa Interface de Programação de Aplicativos
 - Biblioteca (ou uma série delas) com funções e procedimentos públicos que permitem aos programadores desenvolverem aplicações fazendo uso de recursos já definidos.



Leia o artigo disponível em <https://becode.com.br/framework-biblioteca-api-entenda-as-diferencas/> para entender a diferença entre 3 conceitos importantes no desenvolvimento de aplicações.



JAVA - Componentes básicos

- JVM (Java Virtual Machine)
 - Significa Máquina Virtual Java
 - *Software* que emula uma CPU e Memória para a execução de programas Java.

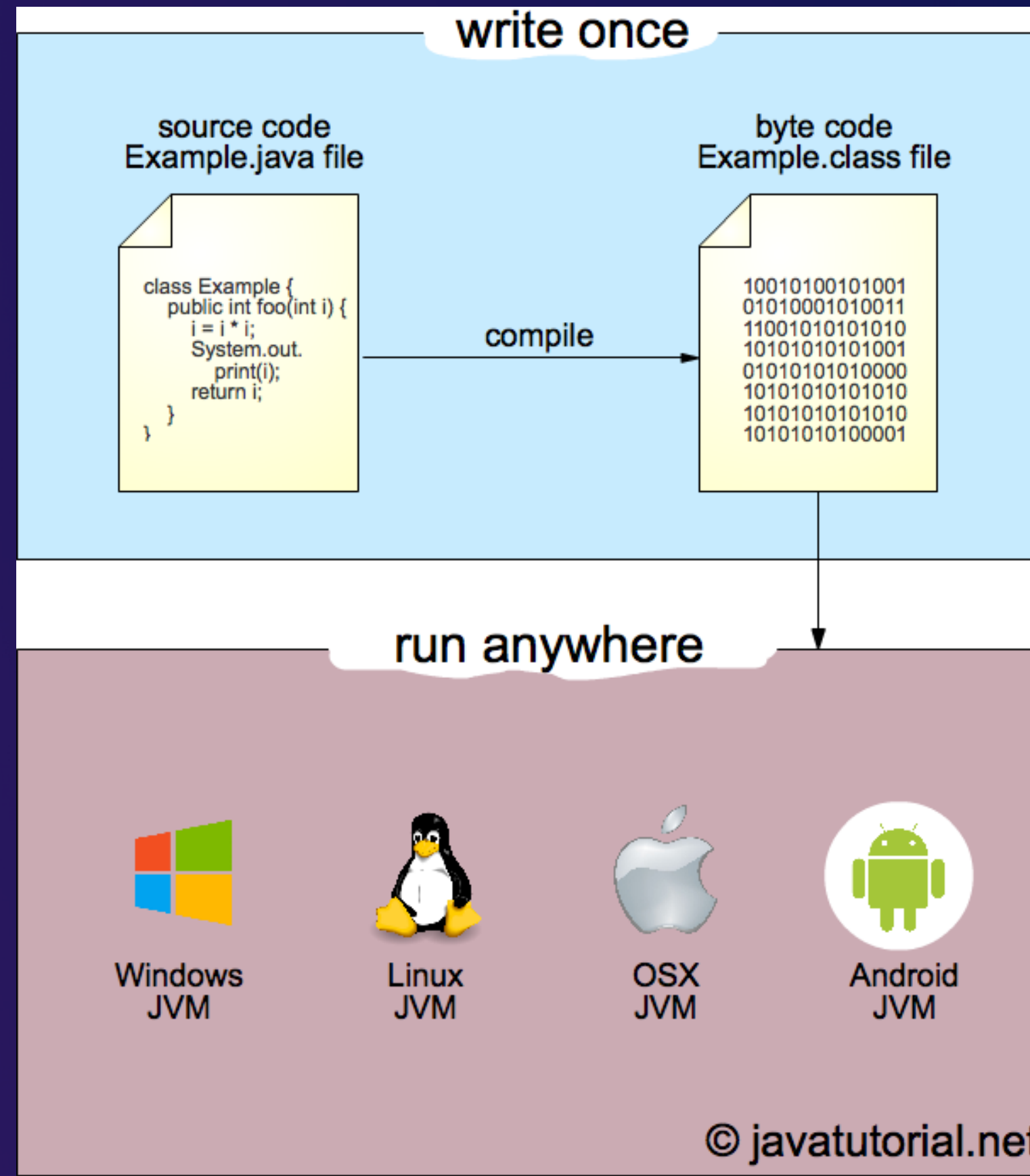


JAVA - Componentes básicos

- JDK (Java Development Kit) ou SDK (Software Development Kit)
 - Significa Kit de Desenvolvimento Java
 - Conjunto de ferramentas para a compilação, documentação e debug. de aplicativos Java.
 - Composto pela JRE somada às ferramentas de desenvolvimento.



“Write Once, Run Anywhere”



“Write Once, Run Anywhere”





Java - Tipos de programas

- **Stand-Alone:**

- Aplicação baseada na JSE, que tem total acesso aos recursos do sistema (memória, disco, rede, dispositivos, etc)
- Um servidor pode executar uma aplicação Stand-Alone, por exemplo, um WebServer.
- Uma estação de trabalho pode executar uma aplicação de Automação Comercial.



Java - Tipos de programas

- **Java Applets:**

- Pequenas aplicações, que não tem acesso aos recursos de hardware e depende de um navegador que suporte a JSE para serem executados, geralmente usados para jogos, animações, teclados virtuais, etc.

- **Java Servlets:**

- Programas escritos e preparados para serem executados dentro de servidores web baseados em JEE, geralmente usados para gerar conteúdo dinâmico de websites.



Java - Tipos de programas

- **Java Midlets:**

- Pequenas aplicações, extremamente seguras, e construídas para serem executadas dentro da JME, geralmente, celulares, Palm Tops, controladores eletrônicos, computadores de bordo, smart cards, tecnologia embarcada em veículos, etc.

- **Java Beans:**

- Pequenos programas, que seguem um padrão bem rígido de codificação, e que tem o propósito de serem aproveitados em qualquer tipo de programa Java, sendo reaproveitados, e podendo ser chamados a partir de: stand-alone, applets, servlets e midlets.



NetBeans IDE 8.2

- É um ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores Java e outras linguagem de programação como, por exemplo, C, C++, PHP, entre outras.
- O NetBeans foi iniciado em 1996 por dois estudantes tchecos na Universidade de Charles, em Praga, quando a linguagem de programação Java ainda não era tão popular como atualmente.
- Pelo fato do Netbeans ser gratuito você poderá fazer download livremente na Internet.
 - Um dos caminhos para baixar o Netbeans 8.2 é acessando o endereço **<http://netbeans.org/downloads/>**. É necessário baixar também o JDK, que é o kit de desenvolvimento de programas Java (ambiente de desenvolvimento).



NetBeans IDE 11

- Agora fazendo parte do rol de softwares livres da Fundação Apache é possível obter a versão atual a partir do endereço
<https://netbeans.apache.org/download/nb110/index.html>

Just released!

Apache NetBeans 11.2

[Find out more](#)

Apache NetBeans

Fits the Pieces Together

Development Environment, Tooling Platform and Application Framework.

Fast & Smart Editing

Apache NetBeans is much more than a text editor. It highlights source code **syntactically and semantically**, lets you easily **refactor code**, with a range of handy and powerful tools.

Java, JavaScript, PHP, HTML5, CSS, and More

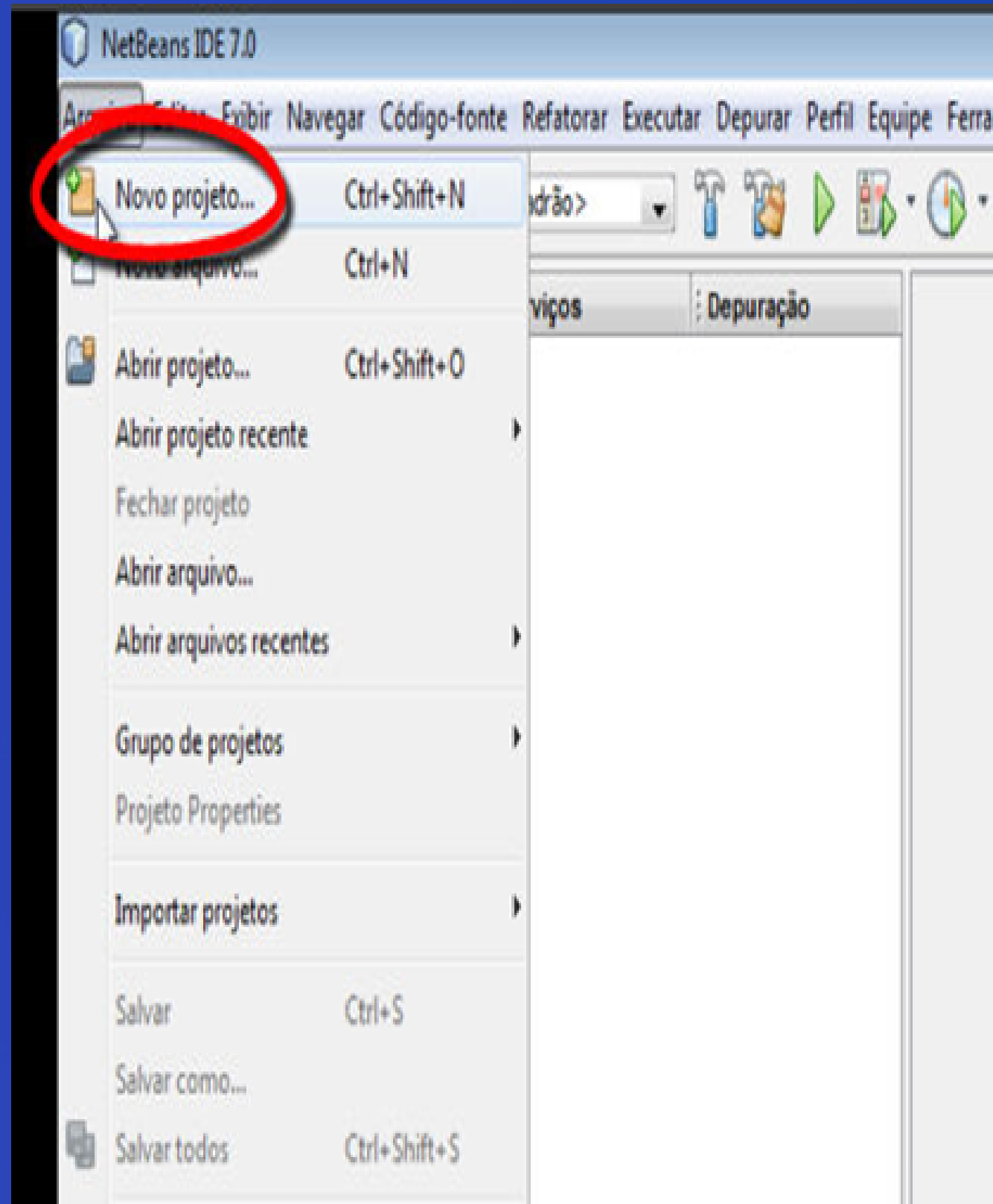
Apache NetBeans provides editors, wizards, and templates to help you create applications in **Java**, **PHP** and many other languages.

Cross Platform

Apache NetBeans can be **installed** on all operating systems that support Java, i.e., Windows, Linux, Mac OSX and BSD. Write Once, Run Anywhere, applies to NetBeans too.



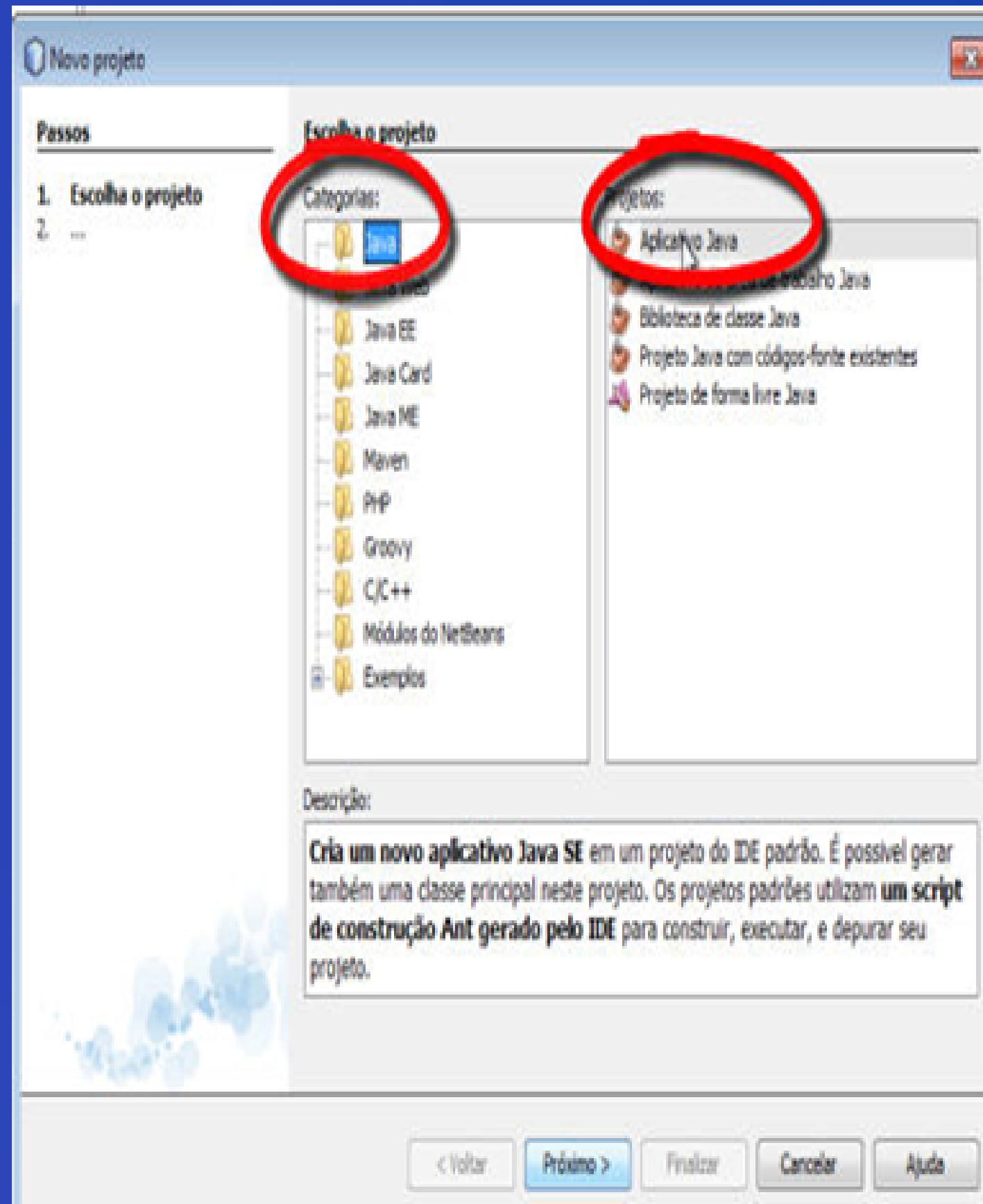
NetBeans - Escrevendo o primeiro código



- Agora vamos aprender a **trabalhar com o NetBeans**.
- É aqui que faremos a **codificação** e **execução** dos nossos programas.
- O primeiro passo é abrir um novo projeto, clicando em **Arquivo - Novo projeto...**



NetBeans - Escrevendo o primeiro código



- Em seguida, escolhemos a categoria **Java** e Projeto – **Aplicativo Java**.
- Clique no botão **Próximo** e dê um nome ao projeto, por exemplo, "ProgramaJava" ou "ProgramaTeste".
- É regra definir o nome do projeto com a primeira letra em maiúscula e as demais primeiras letras de cada palavra também.
- **Não é permitido** espaços nem caracteres especiais na definição do nome do projeto.



NetBeans - Escrevendo o primeiro código

Nome e local

Nome do projeto: JavaApplication26

Localização do Projeto: C:\E-Tec Brasil - Roteiros\Módulo II\netbeans\ProjetoNetbeans Procurar...

Pasta do projeto: steiros\Módulo II\netbeans\ProjetoNetbeans\JavaApplication26

☐ Usar pasta dedicada para armazenar bibliotecas

Pasta Bibliotecas: Procurar...

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

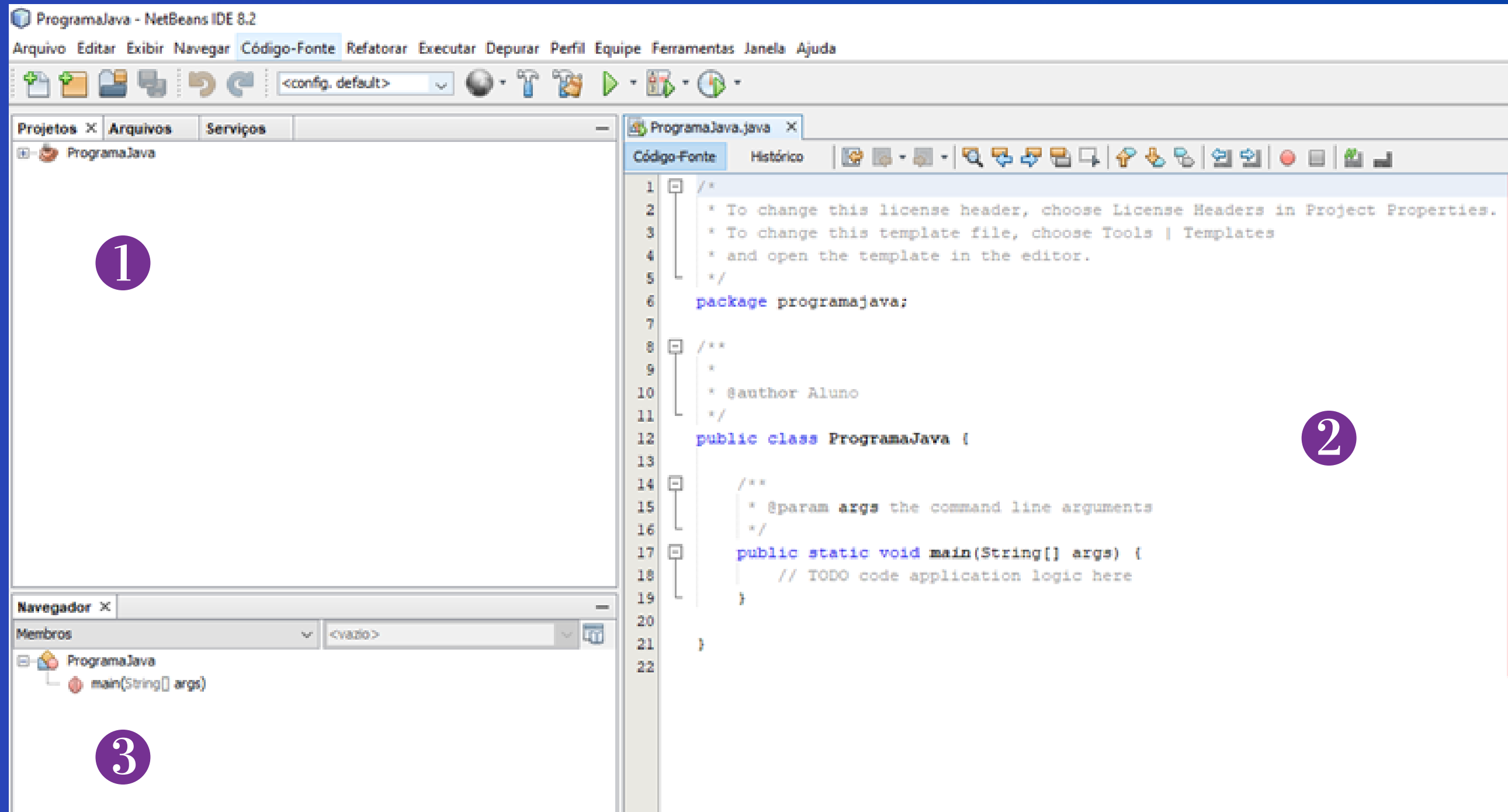
☒ Criar classe principal javaapplication26.JavaApplication26

☒ Definir como projeto principal

- Escolha o diretório onde gravará projeto e clique em **Finalizar**.
- Aguarde enquanto o Netbeans cria o novo projeto.
- Uma janela será aberta para digitarmos o nosso código. Esta janela será mostrada no próximo slide.



NetBeans - Escrevendo o primeiro código



Perceba que esta janela possui três divisões
(serão comentadas no próximo slide).



NetBeans - Escrevendo o primeiro código

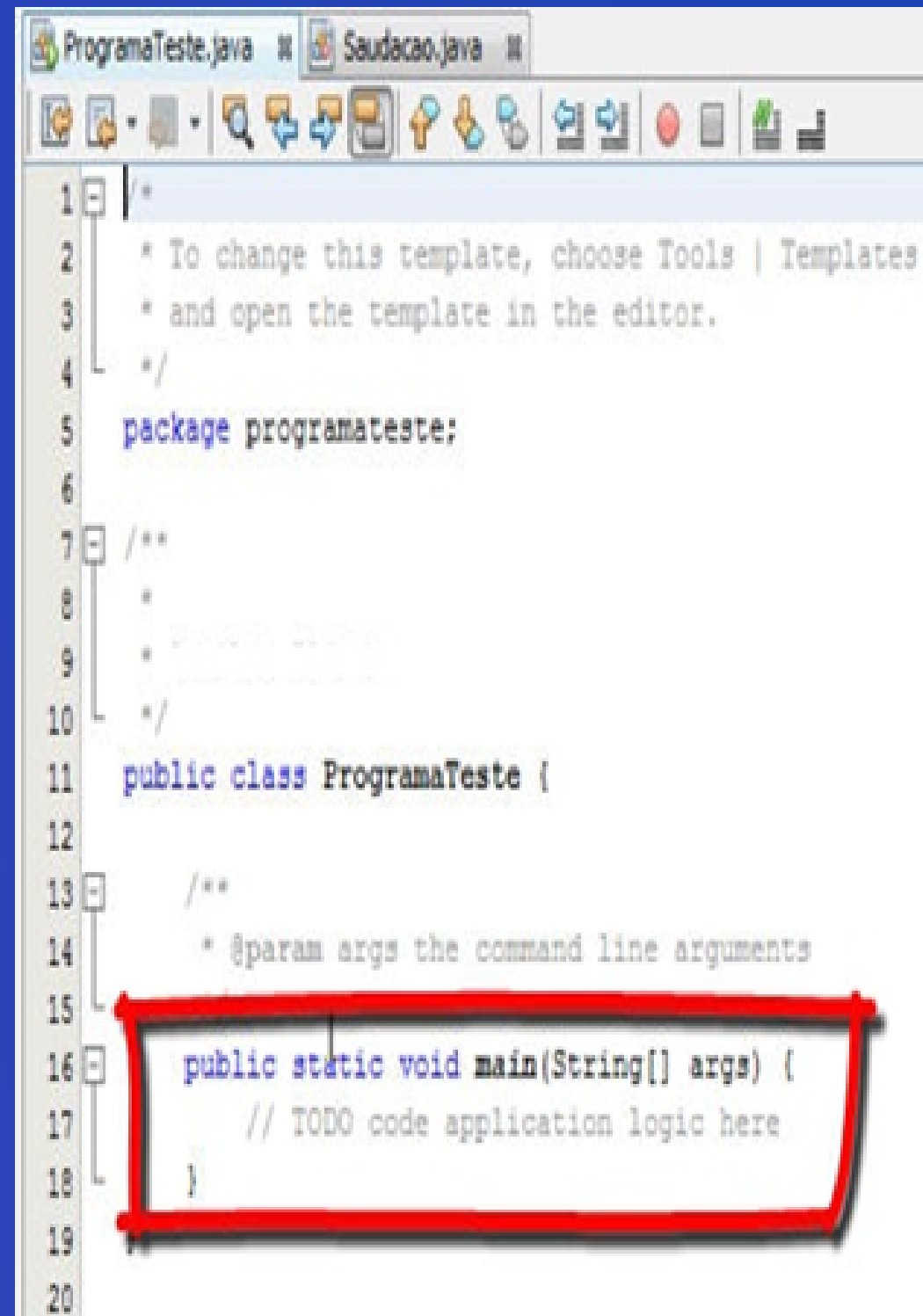
Perceba que esta janela possui três divisões:

- 1 - **Projetos:** onde “navegaremos” nos projetos existentes e executaremos o programa.
- 2 - **Código Fonte:** onde digitaremos o código fonte dentro das respectivas classes.
- 3 - **Membros:** onde encontraremos as “propriedades” (características) dos objetos gerados pelo projeto.

Além disso, esta tela tem a barra de ferramentas onde encontraremos um “atalho” para os comandos mais utilizados no desenvolvimento e execução de um projeto.



NetBeans - Escrevendo o primeiro código



```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package programateste;
6
7  /**
8   *
9   * @author
10  */
11 public class ProgramaTeste {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         // TODO code application logic here
18     }
19
20 }
```

Vejam:

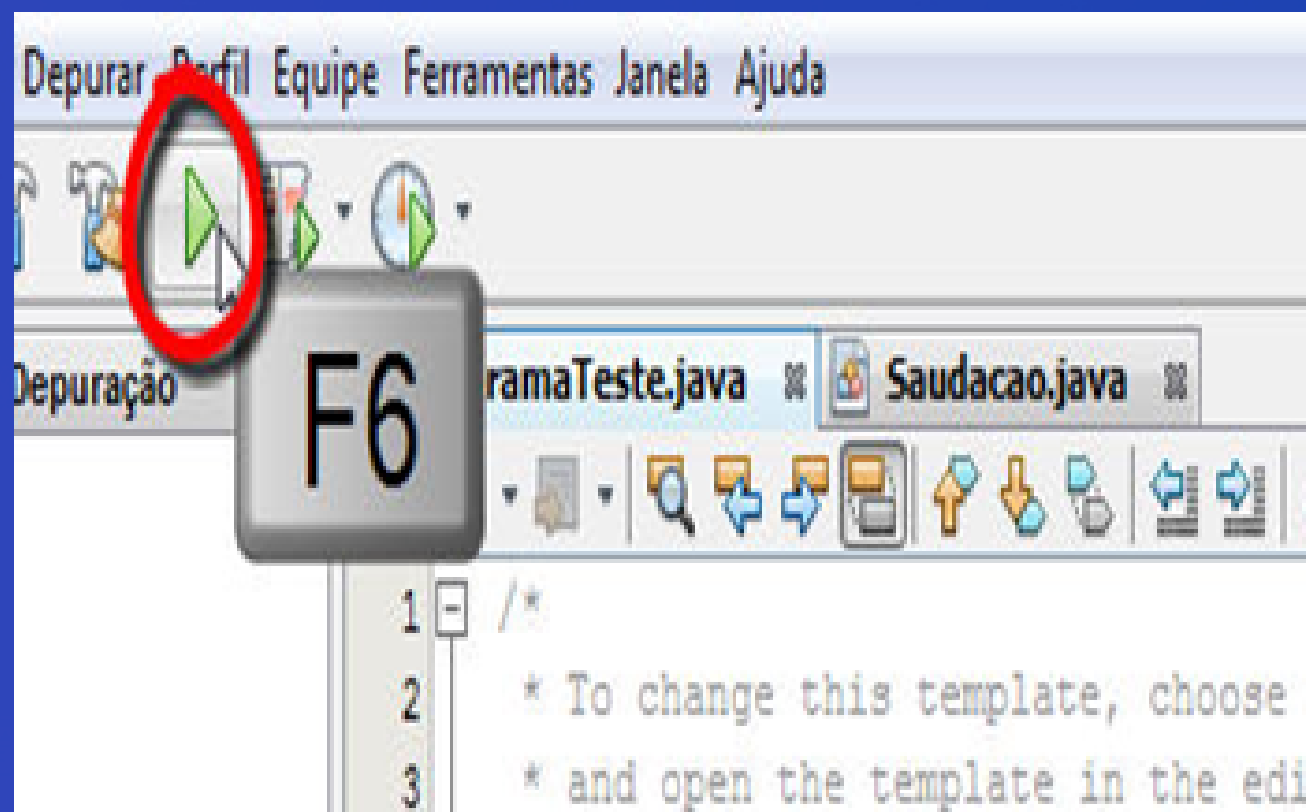
a classe "ProgramaTeste" é nossa classe Principal.
É a classe que contém o método "main" que a JRE procura para executar primeiro.

Vamos fazer um programa bastante simples, apenas para começar a entender o Netbeans!



NetBeans - Escrevendo o primeiro código

```
public class ProgramaTeste {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        System.out.println("Ola !");  
    }  
}
```



Criaremos uma classe bastante simples, apenas para apresentar uma mensagem na tela. Utilizaremos o comando **System.out.println()** para fazer a saída de dados do sistema. Observe:

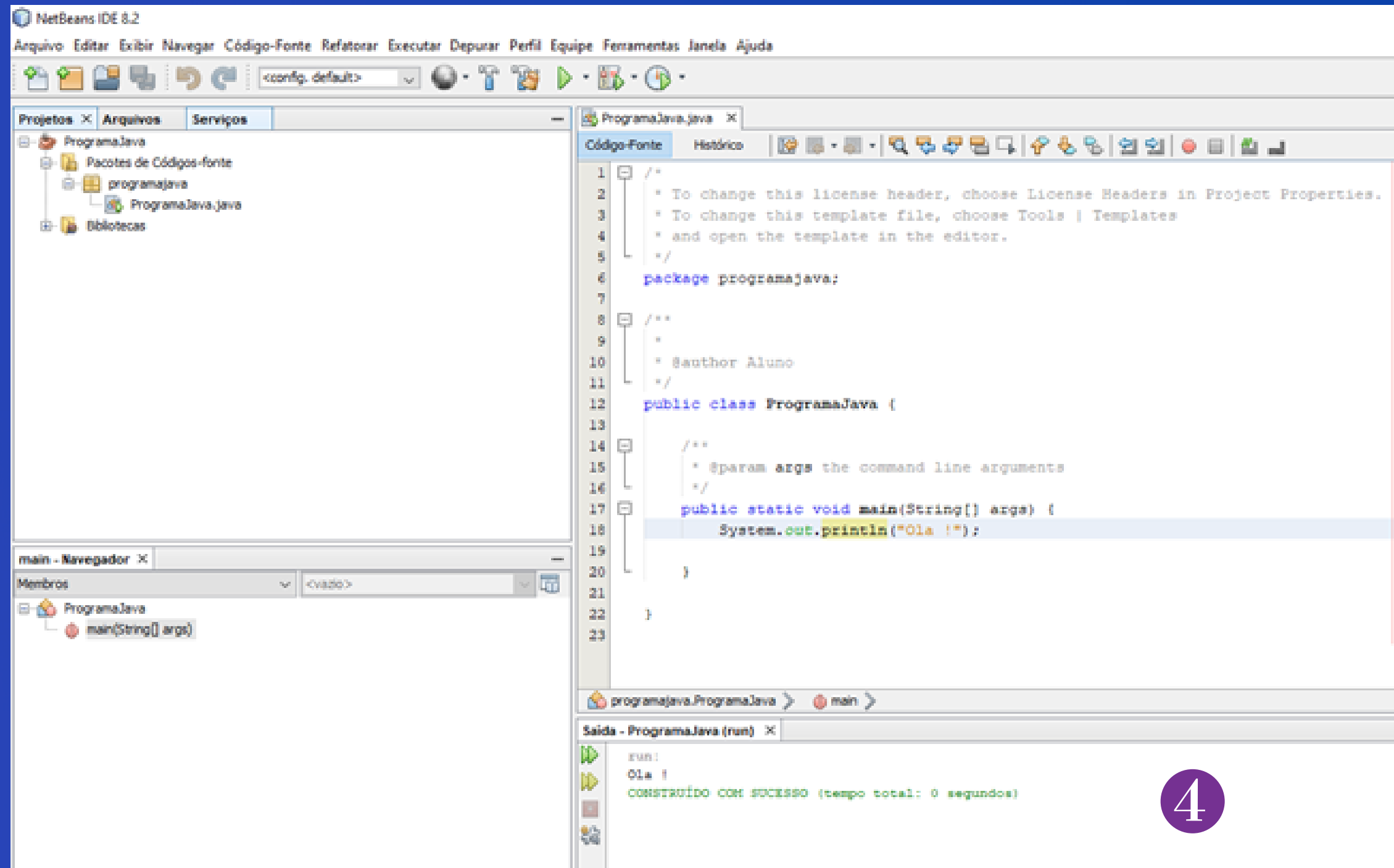
Entre os parênteses do comando escrevemos uma mensagem: "Olá, mundo!"

Pronto! Já podemos executar nosso pequeno programa! Mas antes é necessário salvá-lo, clicando no botão "Salvar Tudo".

Para executá-lo clicamos em "Executar Projeto Principal" ou pressionamos a tecla de atalho "F6".



NetBeans - Escrevendo o primeiro código



E já podemos ver a mensagem de saída do programa. Perceba que foi criada uma **4ª divisão na tela**, abaixo da classe.

Pronto! Você viu como o NetBeans é um ambiente fácil de desenvolvimento?



Escrevendo código em um Editor de Texto

Antes de começarmos a programar utilizando a IDE do Netbeans, veja, apenas para fins de informação/conhecimento uma interface de Editor de código-fonte.

Abra o aplicativo Visual Studio Code e acompanhe os códigos feitos pelo professor para este exemplo...