

# Linguagem Técnica I

**Prof. Jonatas Bastos**

**Email: [jonatasfbastos@gmail.com](mailto:jonatasfbastos@gmail.com)**

# Objetivo da Disciplina

- ❑ Entender os **conceitos** da programação **orientada a objetos**;
- ❑ Aplicar **conceitos básicos** relativos a **objetos** (aprendendo com o apoio da **teoria**, dos **exercícios** e das **atividades**, a “pensar em objetos”);
- ❑ Aprender a Linguagem **Java**, **projetar**, **modelar** e implementar uma aplicação JSE com uso do paradigma de orientação a objetos.



- ❑ Introdução a linguagem JAVA
- ❑ Introdução a IDE
- ❑ Sintaxe Java
- ❑ Introdução ao paradigma de orientação a objetos
- ❑ Orientação a objetos (conceitos avançados)
- ❑ Pacotes
- ❑ Coleções
- ❑ Tratamentos de exceções

- Teremos aulas **expositivas**
  - Com participação dos alunos;
- e aulas **práticas** no **laboratório**.
- Serão passados **exercícios** para **consolidação** da **aprendizagem**.
  - **É fundamental que vocês façam os exercícios!!**

# O que é realmente importante?

- ❑ Não mencionaremos **todos os detalhes** da linguagem Java **juntamente** com seus **princípios** básicos.
- ❑ Nossa ideia é **separar** o que é **primordial** aprender no início daquilo que pode ser **estudado mais a frente**;
- ❑ **Pratique a exaustão**;



# Linguagem JAVA

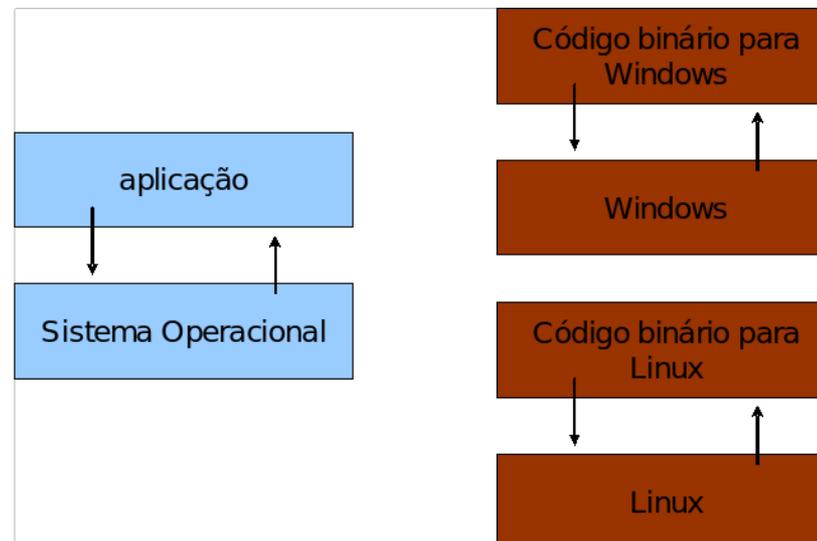
- Quais eram os **maiores problemas** DA **programação** na década de **1990**?
  - Ponteiros?
  - Gerenciamento de memória?
  - Organização?
  - Falta de bibliotecas?
  - Ter de reescrever parte do código ao mudar de sistema operacional?

# Linguagens Tradicionais

- Em uma linguagem de **programação como C e Pascal**, temos a seguinte situação quando vamos **compilar** um programa:

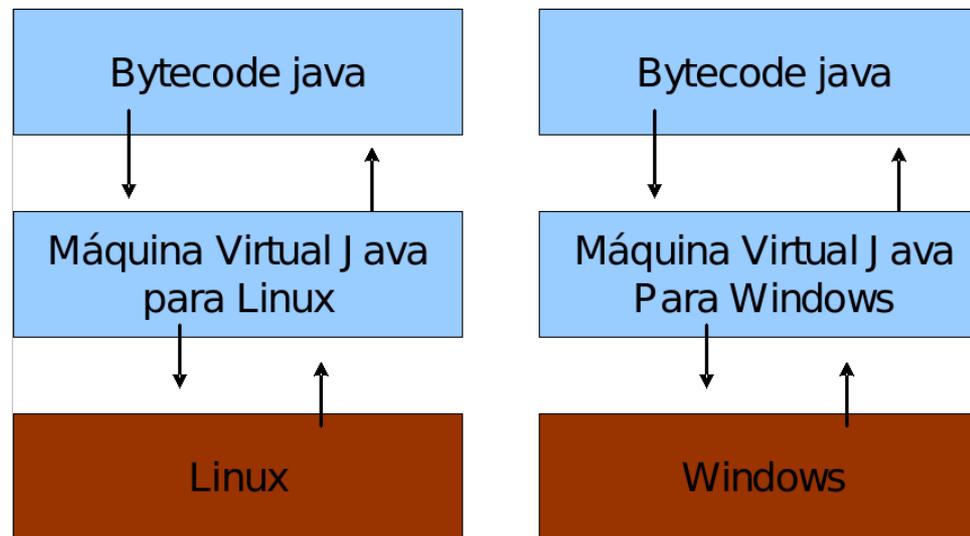


- O código fonte é **compilado** para código de **máquina específico** de uma plataforma e sistema operacional.



# Java – Máquina Virtual

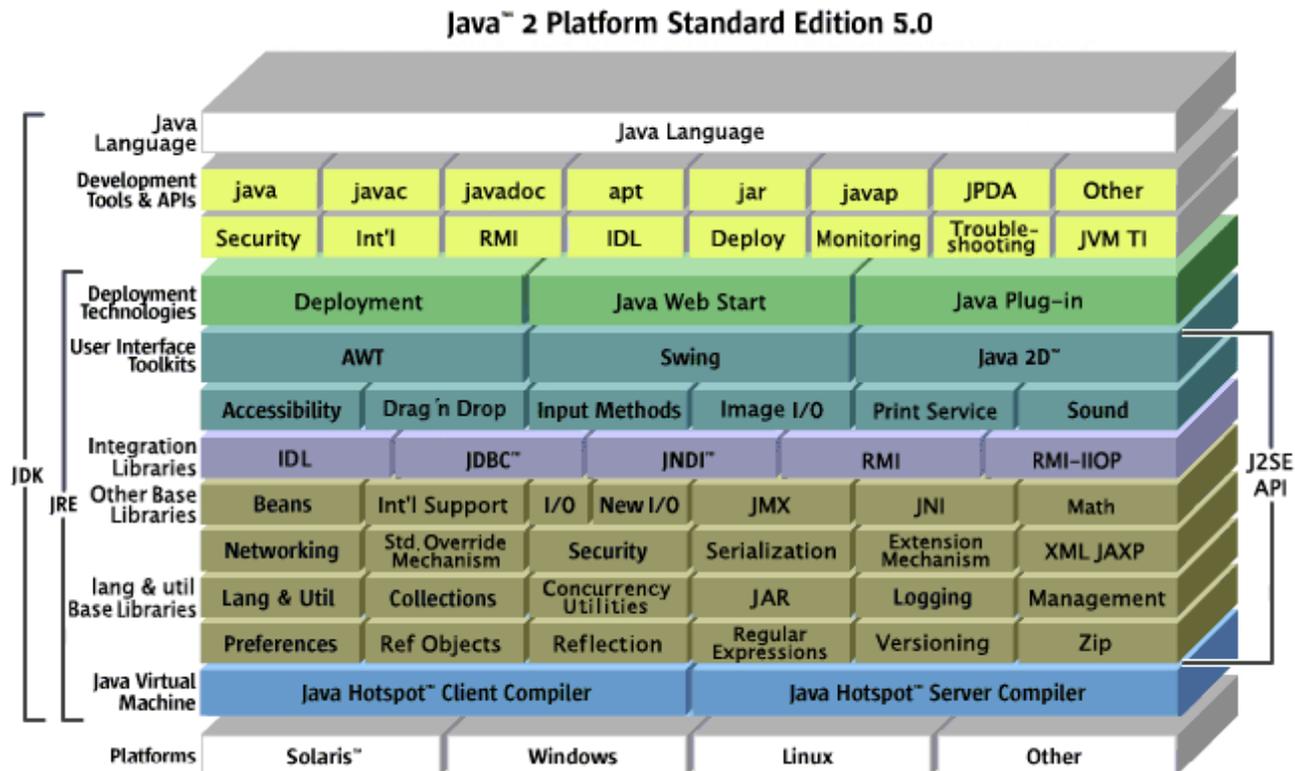
- Java utiliza do conceito de **máquina virtual**, onde existe, entre o **sistema operacional** e a **aplicação**, uma camada extra responsável por “**traduzir**”:



- Sua aplicação roda sem nenhum **envolvimento** com o **sistema operacional**! Sempre conversando apenas com a **Java Virtual Machine** (JVM).

# JRE e JDK

- ❑ **JRE = Java Runtime Environment**, ambiente de execução Java, formado pela JVM e bibliotecas, tudo que você precisa para executar uma aplicação Java.
- ❑ **JDK = Java Development Kit**. Ele é formado pela JRE somado a ferramentas, como o compilador.



# Alguns Indicadores

- ❑ No decorrer do curso, você pode achar que o Java tem **menor produtividade** quando comparada com a **linguagem** que você está **acostumado**;
- ❑ Mas lembrem-se **Java não** é para criar **sistemas pequenos**;
- ❑ O foco da plataforma é outro: aplicações de **médio a grande porte**;
- ❑ Java é uma **linguagem simples**: existem **poucas regras**, muito bem **definidas**.

# Alguns Indicadores

- ❑ **Quebrar** o paradigma **procedural** para mergulhar na **orientação a objetos** não é **simples**.
- ❑ **Criaremos classes** para **testar** esse pequeno aprendizado, sem saber **exatamente** o que é uma **classe**. Isso dificulta ainda mais a curva de aprendizado, porém cada **conceito** será **introduzido** no **momento considerado** mais **apropriado**;

# Preparando o Ambiente

# Preparando Ambiente

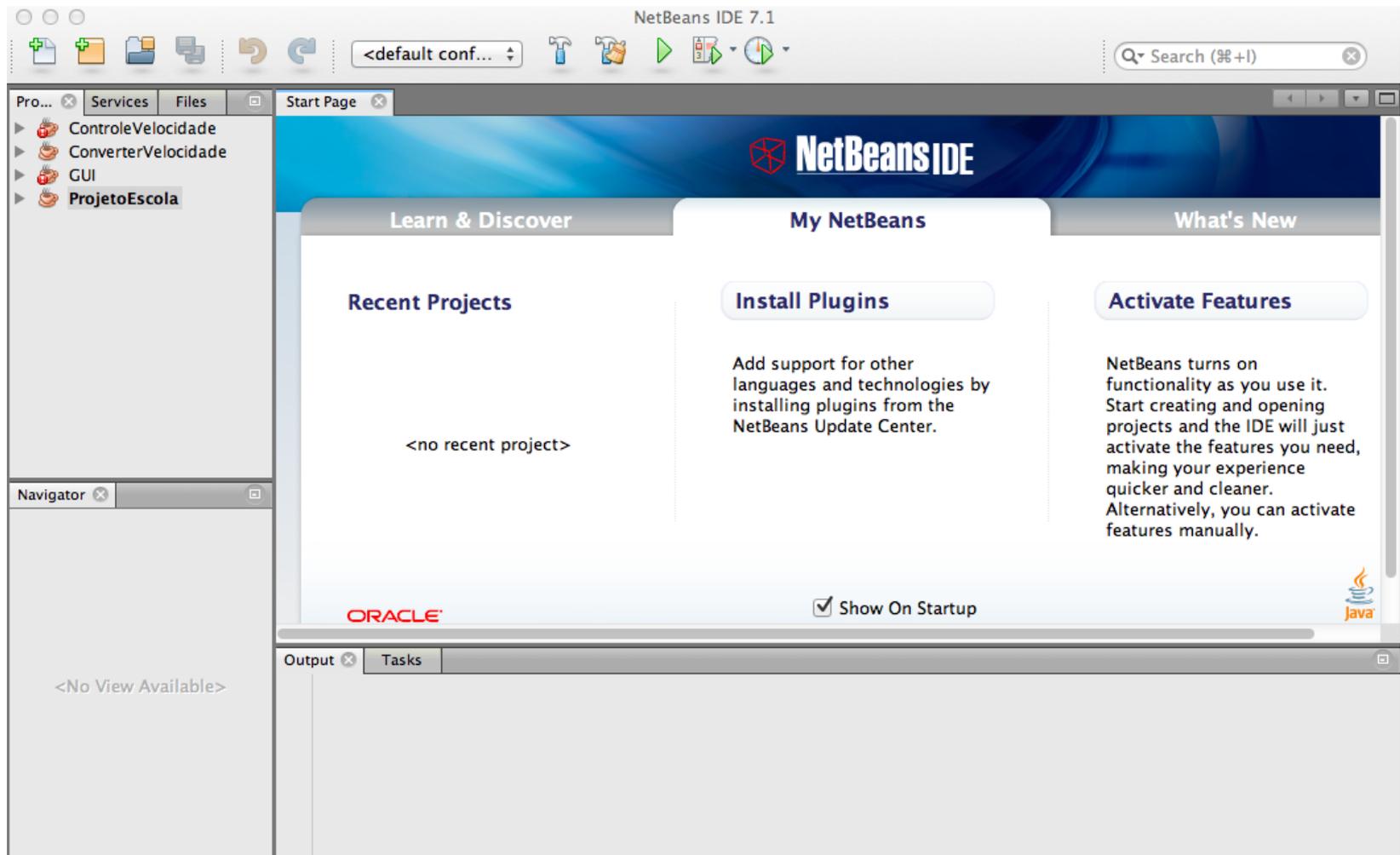
- Para iniciarmos o desenvolvimento de aplicações java é necessário a instalação dos seguintes itens:
  - **JDK - Java Development Kit.**
  - **IDE – Ambiente de Desenvolvimento Integrado**

# IDE - NetBeans

# NetBeans

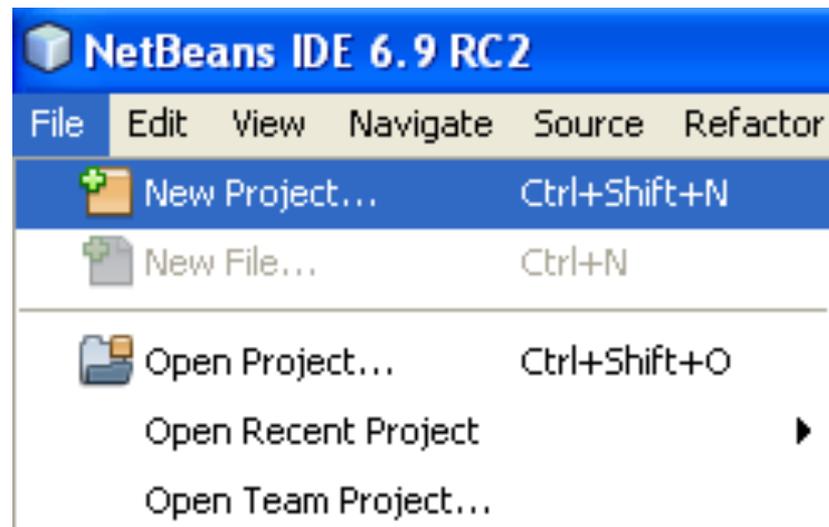
- ❑ O **Netbeans** é um **ambiente integrado de desenvolvimento (IDE)** que permite ao programador criar programas utilizando recursos gráficos.
- ❑ O NetBeans fornece uma **base sólida** para a criação de **projetos** e **módulos**.
- ❑ Como o NetBeans é **escrito** em **Java**, é **independente** de **plataforma**, funciona em qualquer **sistema operacional** que suporte a **máquina virtual Java (JVM)**.

# Conhecendo o NetBeans



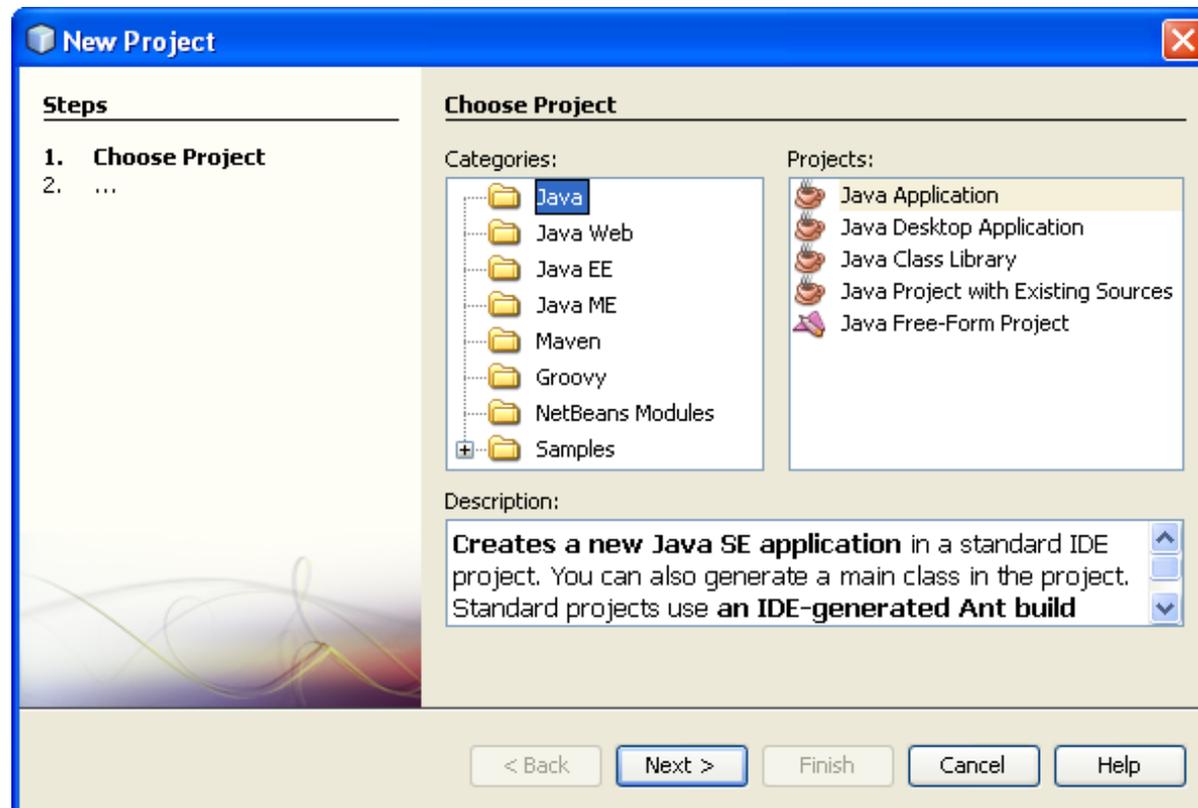
# Configurando o projeto

- No IDE, escolha Arquivo > Novo projeto (Ctrl-Shift-N), como mostrado na figura abaixo



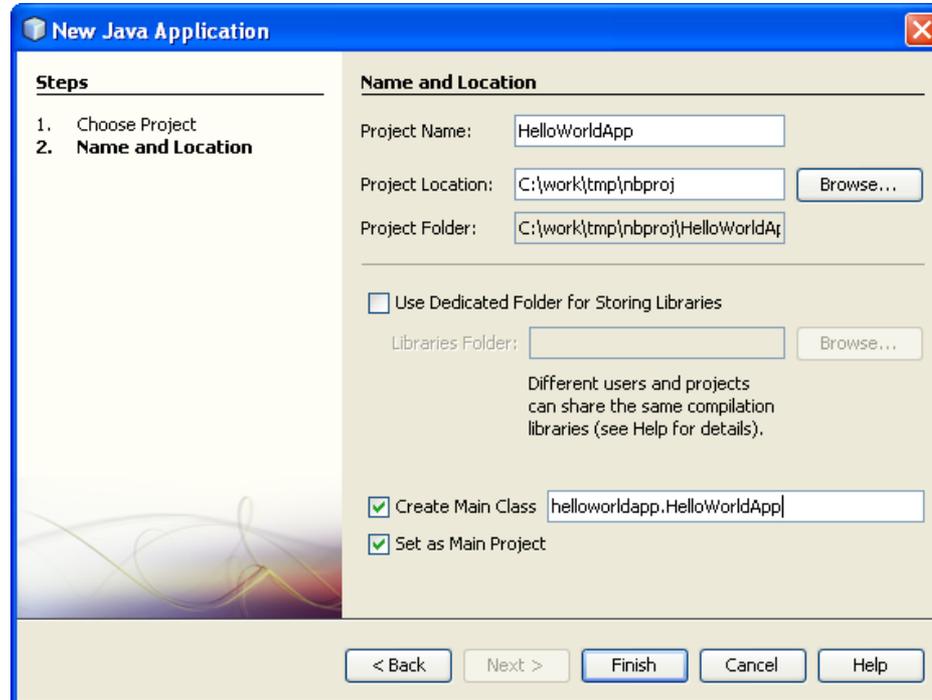
# Configurando o projeto

- ❑ No assistente para Novo projeto, expanda a categoria Java e selecione Aplicação Java, como mostrado na figura abaixo. Em seguida clique em Próximo.



# Configurando o projeto

- ❑ Na página Nome e localização do assistente, adote o procedimento a seguir :
  - no campo Nome do projeto, digite **LinguagemTecnicaIII**.
  - Deixe desmarcada a caixa de verificação Utilizar pasta dedicada para armazenar bibliotecas.
  - No campo Criar classe principal, digite **MeuPrograma**.
  - Deixe marcada a caixa de verificação Definir como projeto principal.



**New Java Application**

**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name: HelloWorldApp

Project Location: C:\work\tmp\nbproj

Project Folder: C:\work\tmp\nbproj\HelloWorldApp

Use Dedicated Folder for Storing Libraries

Libraries Folder:

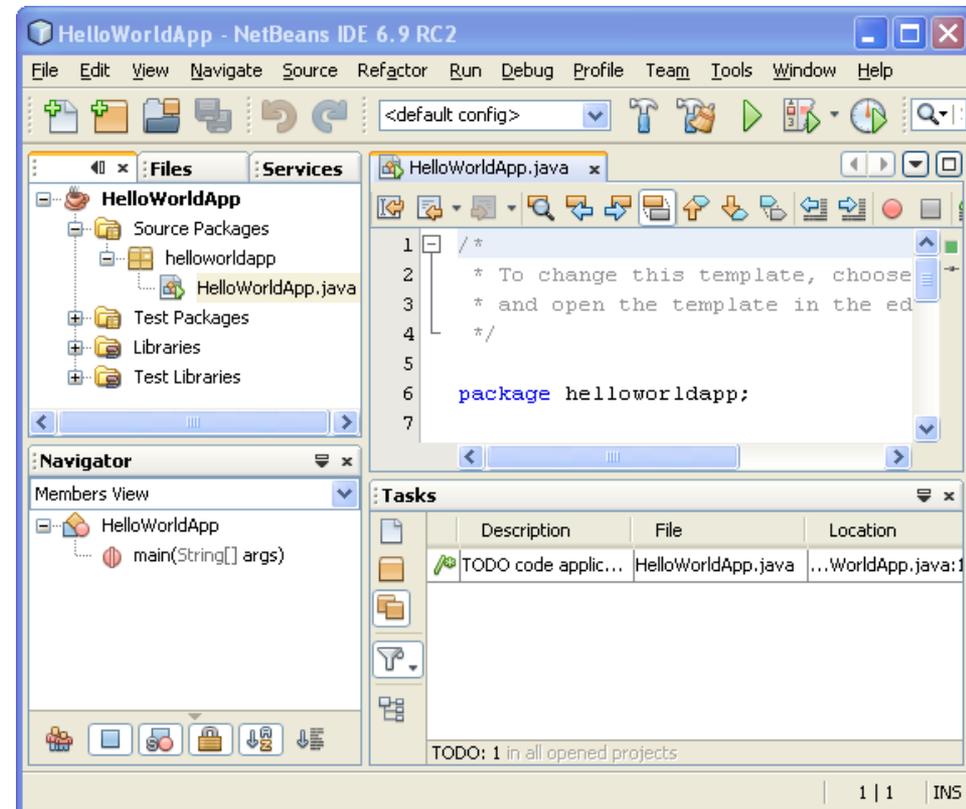
Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class helloworldapp.HelloWorldApp

Set as Main Project

# Configurando o projeto

- ❑ O projeto é criado e aberto no IDE. Agora você deve ver os seguintes componentes:
  - A janela Projetos, que contém uma visualização em árvore dos componentes do projeto.
  - A janela Editor de código-fonte com um arquivo chamado.
  - A janela Navegador, que você pode utilizar para navegar rapidamente entre elementos dentro da classe selecionada.
  - A janela Tarefas, que lista os erros de compilação bem como outras tarefas marcadas com palavras-chave como XXX e TODO.



# Primeiro Programa

```
1
2 public class MeuPrograma {
3
4     //método principal inicia a execução do aplicativo Java
5     public static void main( String args[] )
6     {
7         System.out.println( "Bem vindo a programação em java");
8
9     }//fim do método principal
10
11 }// fim da classe
12
```

- ❑ // na linha quatro é um comentário de fim de linha ( ou de linha única), porque termina no fim da linha que aparece.

- ❑ Java também tem os comentários tradicionais

*/\* Isso é um comentário tradicional*

*Ele pode ser dividido*

*em múltiplas linhas \*/*

# Primeiro Programa

- ***public class MeuPrograma {***
- A palavra chave **class** introduz uma **declaração de classe** em java imediatamente seguida pelo **nome** da classe;
- Por convenção todos os nomes de classe em java iniciam com letra maiúscula;
- O **nome da classe** java é um **identificador** – uma série de caracteres que consistem em **letras, dígitos, sublinhados (\_), e sinais de cifrão (\$)** que não iniciam com **dígito** e não contem espaços;
- Identificadores válidos **Welcome1, \$value, \_value, m\_inputField e button7.**

# Primeiro Programa

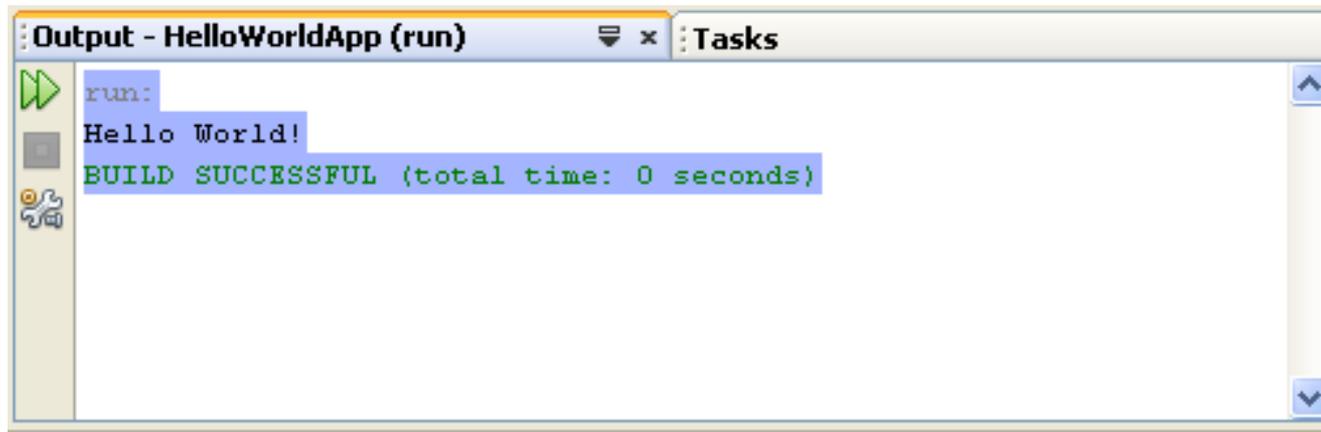
- O java faz distinção entre letras maiúsculas e minúscula, assim a1 e A1 são identificadores diferentes (mas ambos válidos);
  
- ***public static void main( String args[] )***
  - É o ponto de partida de cada aplicativo java;
  - Os parênteses depois do identificador main indicam que é um bloco de construção do programa chamado método.
  - Os métodos realizam tarefas e retornam informações quando completam suas tarefas;
  - A palavra chave void indica que esse método realiza uma tarefa, mas não retornará nenhuma informação;

# Primeiro Programa

- ❑ A chave a esquerda {, logo após o identificador do método, inicia o corpo do método e a chave a direita } finaliza o corpo do método;
- ❑ *System.out.println instrui o computador a realizar uma ação – a saber um string de caracteres contida entre aspas duplas.*
- ❑ *Lembre-se que cada instrução termina em um ; (ponto-e-vírgula);*

# Compilando e executando o programa

- **Para executar o programa:**
  - Escolha Executar > Executar projeto principal (F6).



The screenshot shows a window titled "Output - HelloWorldApp (run)" with a "Tasks" tab. The output text is as follows:

```
run:  
Hello World!  
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Primeiro Programa

- 1) Altere seu programa para imprimir uma mensagem diferente.
- 2) Altere seu programa para imprimir duas linhas de texto usando duas linhas de código `System.out`.
- 3) Sabendo que os caracteres `\n` representam uma quebra de linhas, imprima duas linhas de texto usando uma única linha de código `System.out`.