



# Java Foundations

1-3

Configurando o Java

**ORACLE**  
Academy



Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

# Objetivos

- Esta lição abrange os seguintes objetivos:
  - Entender a diferença entre o JDK e o JRE
  - Entender a diferença entre arquivos .java e .class
  - Descrever a finalidade de um Ambiente de Desenvolvimento Integrado (IDE, Integrated Development Environment)
  - Fazer download e instalar o JDK, o JRE e o NetBeans IDE
  - Importar um projeto para o NetBeans



# Tópicos

- **Noções Básicas sobre o JRE, o JDK e o NetBeans IDE**
- **É Hora de Configurar!**



**ORACLE**  
Academy

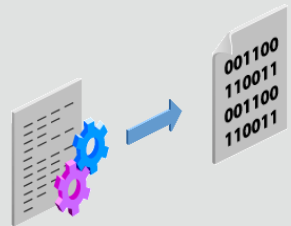
JFo 1-3  
Configurando o Java

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

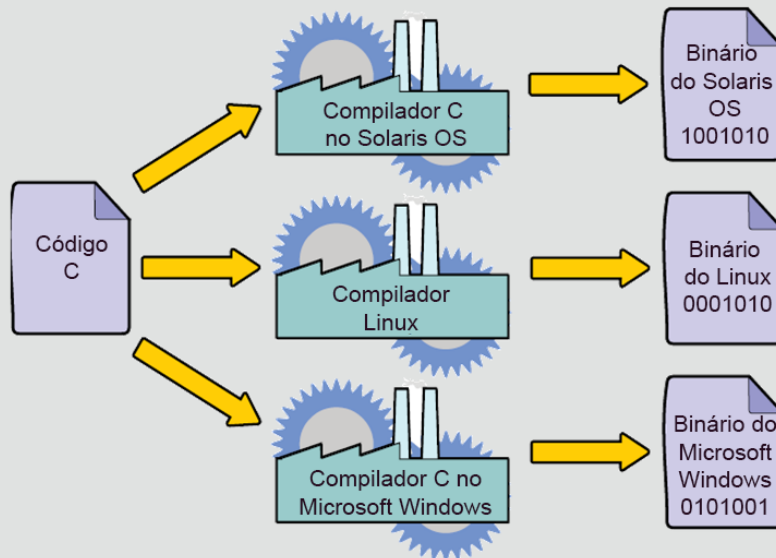
4

# Finalidade de um Programa de Computador

- Um programa de computador é um conjunto de informações que são executadas em um computador ou em outro dispositivo digital
- No nível da máquina, o programa consiste em instruções binárias (1s e 0s)
  - Código de máquina
- A maioria dos programas é criada com código de alto nível (legível)
  - Eles devem ser convertidos em código de máquina



# Convertendo Código de Alto Nível em Código de Máquina



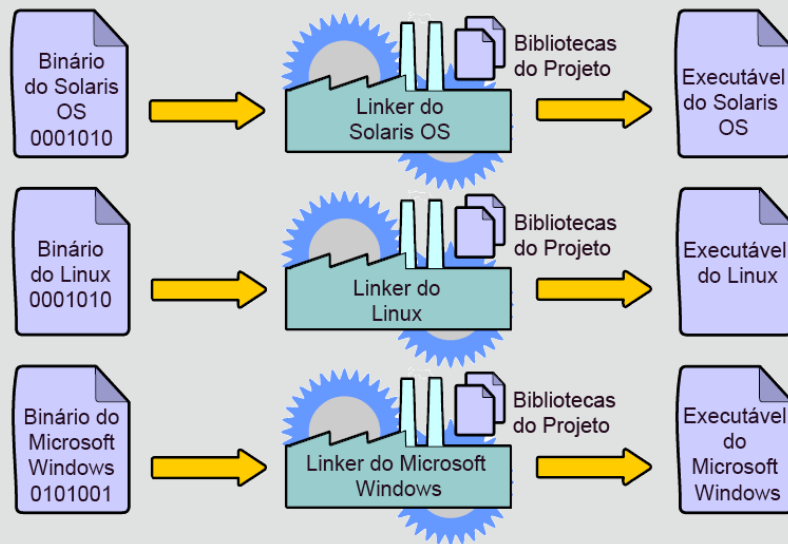
Os programas criados na maioria das linguagens geralmente requerem diversas modificações para serem executados em mais de um tipo de plataforma de computação (uma combinação de CPU e um sistema operacional). Isso acontece porque a maioria das linguagens requer que você crie um código específico para a plataforma subjacente.

As linguagens de programação conhecidas como C e C++ exigem que os programadores compilem e vinculem seus programas, o que resulta em um programa executável que é exclusivo para uma plataforma. Um compilador é um aplicativo que converte um programa em um código específico de CPU denominado *código de máquina*. Esses arquivos específicos da plataforma, denominados arquivos binários, geralmente são combinados com outros arquivos, como bibliotecas de código pré-criado. E um vinculador cria um programa dependente da plataforma, denominado *executável*, que o usuário final pode executar.

Ao contrário de C e C++, a linguagem de programação Java é independente de plataforma.

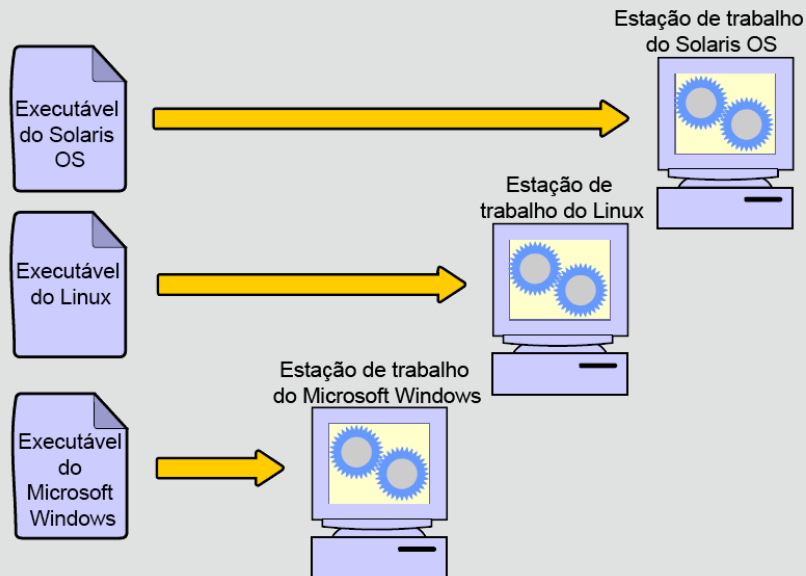
A imagem ilustra como um compilador cria um arquivo binário.

# Vinculado a Bibliotecas Específicas de Plataforma



A imagem ilustra como um arquivo binário é vinculado a bibliotecas para criar um arquivo executável dependente de plataforma.

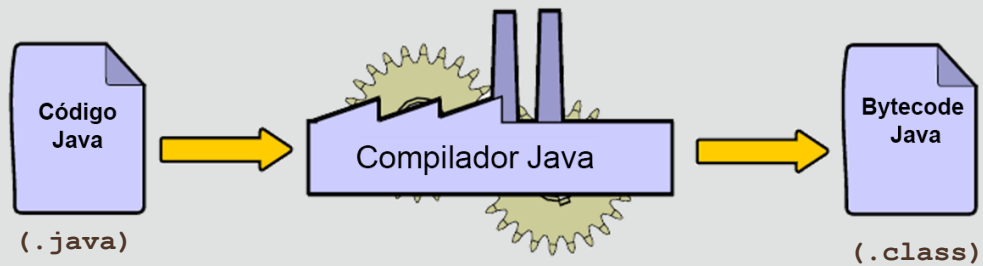
# Programas Dependentes de Plataforma



A imagem ilustra como os arquivos executáveis dependentes de plataforma só podem ser executados em uma única plataforma.



# O Java é Independente de Plataforma

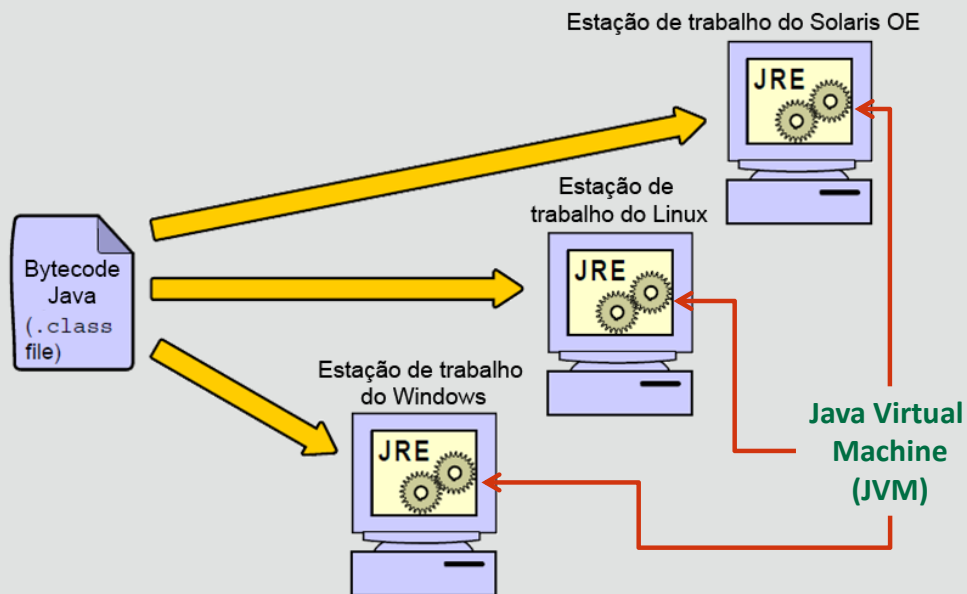


Um programa Java pode ser executado em várias combinações diferentes de CPUs e sistemas operacionais, como o SO Solaris em um chip SPARC e o MacOS X e o Microsoft Windows em um chip Intel, geralmente com poucas ou nenhuma modificação.

Como é ilustrado na imagem, os programas Java são compilados com um compilador Java. O formato resultante de um programa Java compilado é um código de bytes Java independente de plataforma, em vez de um código de máquina específico da CPU.

O código de bytes criado é interpretado por um interpretador de código de bytes denominado JVM (Java Virtual Machine). Uma máquina virtual é um programa específico de plataforma que compreende o código de bytes independente de plataforma e pode executá-lo em uma plataforma específica. Por esse motivo, a linguagem de programação Java costuma ser referida como uma linguagem de interpretação, e os programas de tecnologia Java são referidos como sendo portáteis ou executáveis em qualquer plataforma. Outra linguagem interpretada é Perl.

# Programas Java São Executados em uma JVM



ORACLE  
Academy

JFo 1-3  
Configurando o Java

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

10

A imagem ilustra um arquivo de código de bytes Java que é executado em várias plataformas em que existe um ambiente de run-time Java.

Uma máquina virtual recebe seu nome porque é um trecho de software que executa o código, uma tarefa que geralmente é feita pela máquina de hardware ou pela CPU. Para os programas Java serem independentes da plataforma, é necessário que a JVM esteja em toda plataforma em que seu programa será executado. A JVM é responsável por interpretar o código Java, carregando classes Java e executando programas Java.

No entanto, um programa Java precisa de mais de uma JVM para ser executado. Ele também precisa de um conjunto de bibliotecas de classe Java para a plataforma. As bibliotecas de classe Java são um código pré-criado que pode ser combinado com o código que você escreve para criar aplicativos robustos.

Combinados, o software da JVM e as bibliotecas de classe Java denominam-se **JRE (Java Runtime Environment)**. Os JREs estão disponíveis na Oracle para muitas plataformas comuns.

# JRE (Java Runtime Environment)

- Inclui o seguinte:

- A Máquina Virtual Java (JVM, Java Virtual Machine)
- Bibliotecas de classes Java



JRE

- Finalidade:

- Ler o código de bytes (.class)
- Executar o mesmo código de bytes em qualquer parte de uma JVM

# JDK (Java Development Kit)

- Inclui o seguinte:

- JRE
- Compilador Java
- Ferramentas adicionais



JRE



JDK

- Finalidade:

- Compilar o código de bytes  
(.java → .class)

# Ambiente de Desenvolvimento Integrado (IDE, Integrated Development Environment)

- Finalidade:

- Fornecer um editor de texto sofisticado
- Oferecer um código de depuração de assistência
- Gerenciar projetos
- Escrever o código-fonte (.java)

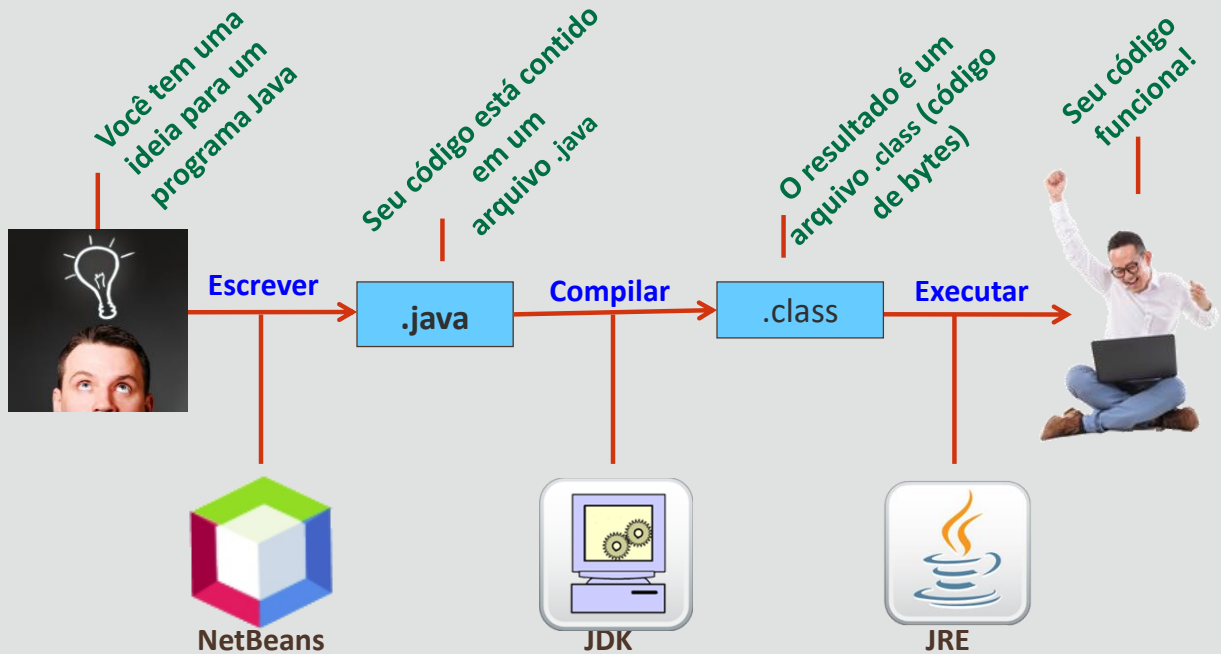
- Exemplos:

- NetBeans
- Greenfoot e BlueJ
- Alice



NetBeans IDE

# Compilando e Executando um Programa Java



ORACLE  
Academy

JFo 1-3  
Configurando o Java

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

14

O diagrama mostra o que acontece quando você compila e executa um programa Java:

- Você tem uma ideia para um programa Java.
- O código Java é escrito no NetBeans, e o arquivo tem uma extensão `.java`. Isso denomina-se “código-fonte Java”.
- O componente compilador do JDK compila o código-fonte em um código de bytes com uma extensão `.class`. Isso denomina-se classe Java.
- O componente JVM do JRE executa a classe Java. Esse é seu programa Java.
- Comemore muito quando seu código funcionar porque, na maioria das vezes, ele não funcionará. Este diagrama simplifica demais o aspecto de depuração do desenvolvimento.

# Tópicos

- Noções Básicas sobre o JRE, JDK e o NetBeans IDE
- **É Hora de Configurar!**



**ORACLE**  
Academy

JFo 1-3  
Configurando o Java

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

15

# É Hora de Configurar!



**JRE**



**JDK**



**NetBeans IDE**



# Download do JDK 8 e do NetBeans

1. Acesse a página de Downloads do Java SE em <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
2. Localize o Java SE Development Kit mais recente

Java SE Development Kit 8u191		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.97 MB	<a href="#">jdk-8u191-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	69.92 MB	<a href="#">jdk-8u191-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	170.89 MB	<a href="#">jdk-8u191-linux-i586.rpm</a>
Linux x86	185.69 MB	<a href="#">jdk-8u191-linux-i586.tar.gz</a>
Linux x64	167.99 MB	<a href="#">jdk-8u191-linux-x64.rpm</a>
Linux x64	182.87 MB	<a href="#">jdk-8u191-linux-x64.tar.gz</a>
Mac OS X x64	245.92 MB	<a href="#">jdk-8u191-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	133.04 MB	<a href="#">jdk-8u191-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	94.28 MB	<a href="#">jdk-8u191-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	134.04 MB	<a href="#">jdk-8u191-solaris-x64.tar.Z</a>
Solaris x64	92.13 MB	<a href="#">jdk-8u191-solaris-x64.tar.gz</a>
Windows x86	197.34 MB	<a href="#">jdk-8u191-windows-i586.exe</a>
Windows x64	207.22 MB	<a href="#">jdk-8u191-windows-x64.exe</a>

**ORACLE**  
Academy

JFo 1-3  
Configurando o Java

Copyright © 2020, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

17

Você precisará do Java SE Development Kit 8 (JDK 8) para este curso. Os números de versão do download podem diferir, pois passam por atualizações frequentes

# Download do JDK 8 e do NetBeans

3. Aceite o contrato de licença
4. Escolha o download correto para seu sistema

**Java SE Development Kit 8u191**

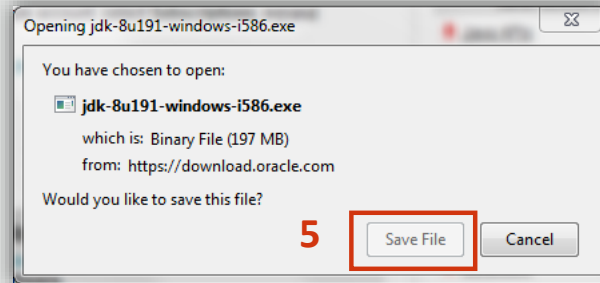
You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software

3 ☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.97 MB	<a href="#">jdk-8u191-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	69.92 MB	<a href="#">jdk-8u191-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	170.89 MB	<a href="#">jdk-8u191-linux-i586.rpm</a>
Linux x86	185.69 MB	<a href="#">jdk-8u191-linux-i586.tar.gz</a>
Linux x64	167.99 MB	<a href="#">jdk-8u191-linux-x64.rpm</a>
Linux x64	182.87 MB	<a href="#">jdk-8u191-linux-x64.tar.gz</a>
Mac OS X x64	245.92 MB	<a href="#">jdk-8u191-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	133.04 MB	<a href="#">jdk-8u191-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	94.28 MB	<a href="#">jdk-8u191-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	134.04 MB	<a href="#">jdk-8u191-solaris-x64.tar.Z</a>
Solaris x64	92.13 MB	<a href="#">jdk-8u191-solaris-x64.tar.gz</a>
Windows x86	197.34 MB	<a href="#">jdk-8u191-windows-i586.exe</a>
Windows x64	207.22 MB	<a href="#">jdk-8u191-windows-x64.exe</a>

## Download do JDK 8 e do NetBeans

5. Salve o arquivo .exe no seu computador, com atenção ao local



# Download do JDK 8 e do NetBeans

1. Acesse a página de Downloads do Netbeans em <https://netbeans.org/downloads/8.2/>
2. Faça download do Java SE

NetBeans IDE 8.2 Download

8.1 | 8.2 | Development | Archive

Email address (optional):

IDE Language:  Platform:

Subscribe to newsletters: ☒ Monthly ☐ Weekly

☒ NetBeans can contact me at this address

Note: Greyed out technologies are not supported for this platform

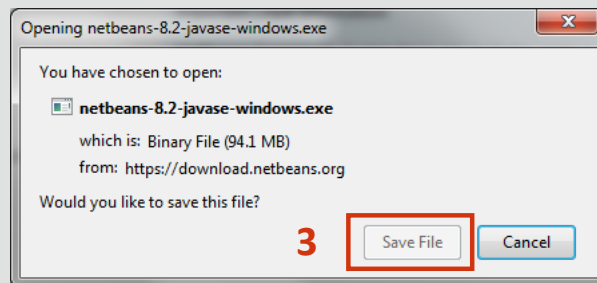
Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
NetBeans Platform SDK	•	•				•
Java SE	•	•				•
Java FX	•	•				•
Java EE		•				•
Java ME						•
HTML5/JavaScript		•	•	•		•
PHP			•	•		•
C/C++					•	•
Groovy						•
Java Card™ 3						•
Connected						•
Bundled servers						
GlassFish Server Open Source Edition 4.1.1		•				•
Apache Tomcat 8.0.27		•				•

Download bundles:

Download	Download	Download x86	Download x86	Download x86	Download
Free, 95 MB	Free, 197 MB	Free, 108 - 112 MB	Free, 108 - 112 MB	Free, 107 - 110 MB	Free, 221 MB

## Download do JDK 8 e do NetBeans

3. Salve o arquivo .exe no seu computador, com atenção ao local

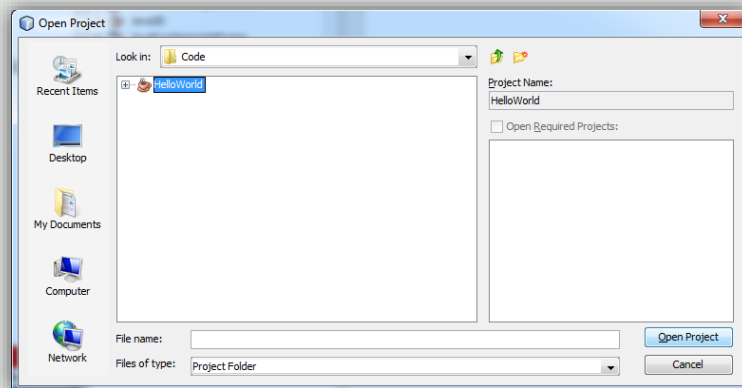


## Instalando o JDK 8 e o NetBeans

1. Localize e execute o arquivo de instalador do JDK no seu computador para iniciar a instalação
2. Siga as instruções na tela para concluir a instalação
3. Localize e execute o arquivo de instalador do NetBeans no seu computador para iniciar a instalação
4. Siga as instruções na tela para concluir a instalação

# Testando o NetBeans

1. Faça download do Projeto HelloWorld e descompacte-o:
2. Inicie o NetBeans:
3. Selecione File > Open Project e escolha HelloWorld

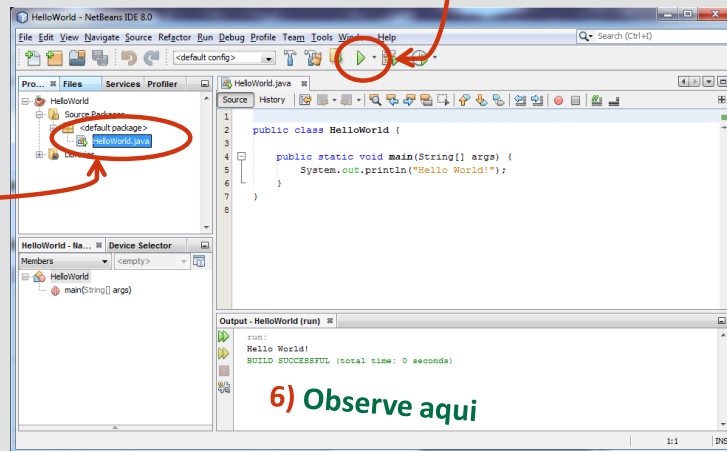


## Testando o NetBeans

4. Expanda o projeto e abra o HelloWorld.java
5. Compile e execute o projeto
6. Observe a saída

4) Clique duas vezes aqui

5) Clique aqui



6) Observe aqui





## Exercício

- Edite o código de modo que ele imprima uma mensagem diferente de "**Hello World!**"
  - Compile e execute seu código para verificar o efeito das alterações que foram feitas
- Altere o 'S' em "System" para letra minúscula e tente compilar o código
  - A sintaxe Java faz distinção entre letras maiúsculas e minúsculas?
- Remova o ponto e vírgula (;) e tente compilar
  - O ponto e vírgula é importante para a sintaxe Java?

Observação para os Instrutores: as instruções de impressão são apresentadas formalmente na Seção 2. A meta deste exercício é verificar se tudo está instalado corretamente. O exercício também oferece aos alunos uma chance de aprender por meio da experiência, o que irá torná-los mais receptivos quando as instruções de impressão forem discutidas mais adiante.

## Resumo

- Um programa de computador é criado em uma linguagem de alto nível, mas deve ser compilada no código da máquina
  - A maioria das linguagens de programação é compilada como um arquivo executável separado para cada plataforma
  - O Java é independente de plataforma



Um IDE como o NetBeans é usado para **criar** o código-fonte (.java) .



O JDK **compila** um código de bytes (.java → .class).



O código de bytes **é executado** em uma JVM, que faz parte do JRE.

# Resumo

- Nesta lição, você deverá ter aprendido a:
  - Entender a diferença entre o JDK e o JRE
  - Entender a diferença entre arquivos .java e .class
  - Descrever a finalidade de um IDE.
  - Fazer download e instalar o JDK, o JRE e o NetBeans IDE
  - Importar um projeto para o NetBeans



The Oracle Academy logo is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by a thin black border, with dark gray horizontal bars at the top and bottom.

# ORACLE

## Academy