

The logo for Oracle Academy 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

# Java Foundations

1-3

Configurando o Java

**ORACLE**  
Academy



Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

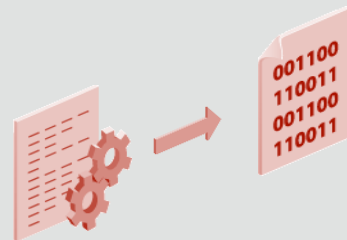
# Objetivos

- 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)
  - Adicione o arquivo .java existente a um projeto Java

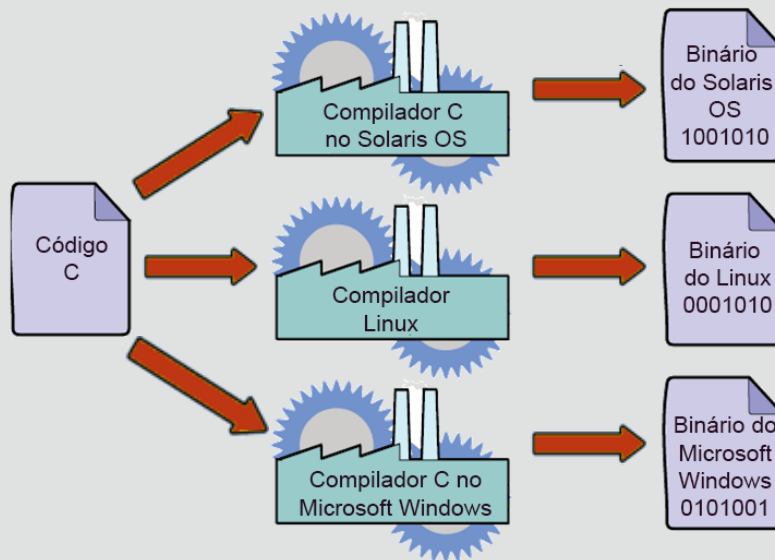


# Finalidade de um Programa de Computador

- 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



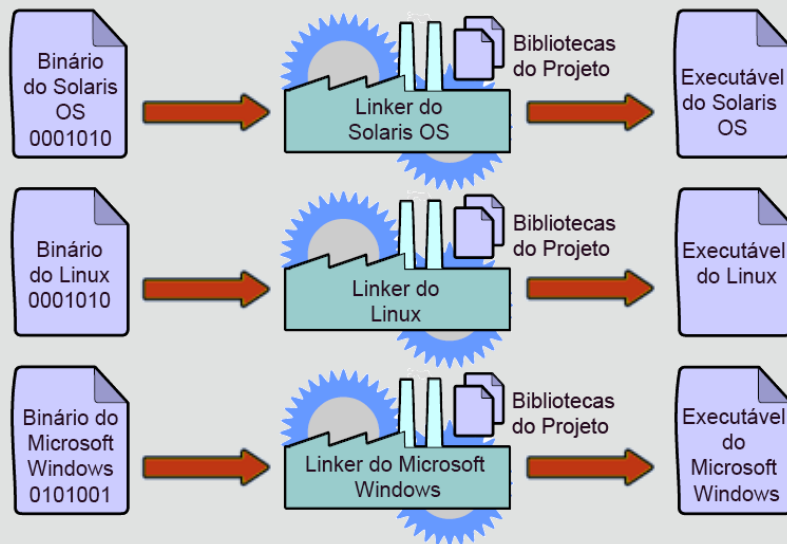
Os programas criados na maioria das linguagens geralmente requerem diversas modificações para serem executados em mais de um tipo de plataforma de computação (uma combinação de CPU e um sistema operacional). Isso acontece porque a maioria das linguagens requer que você crie um código específico para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

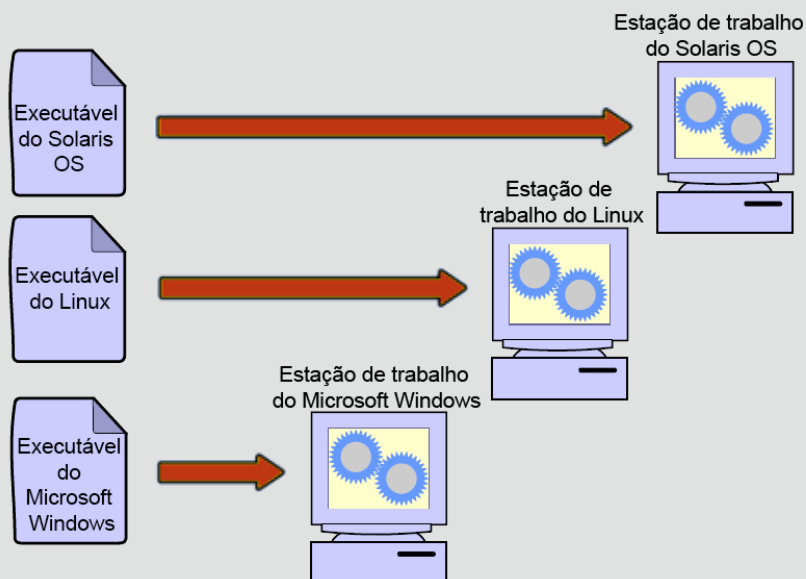
A imagem ilustra como um compilador cria um arquivo binário.

# Vinculado a Bibliotecas Específicas de Plataforma



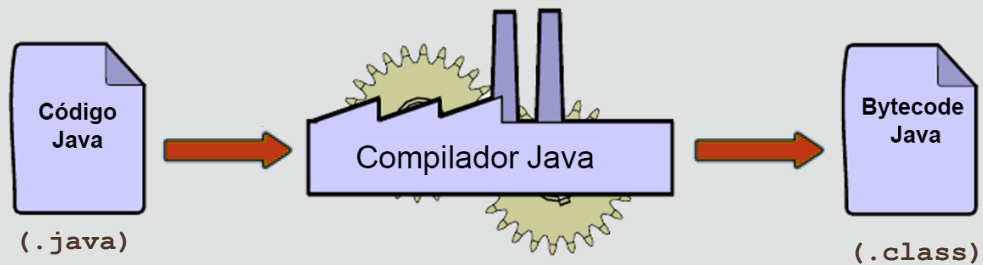
A imagem ilustra como um arquivo binário é vinculado a bibliotecas para criar um arquivo executável dependente de plataforma.

# 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



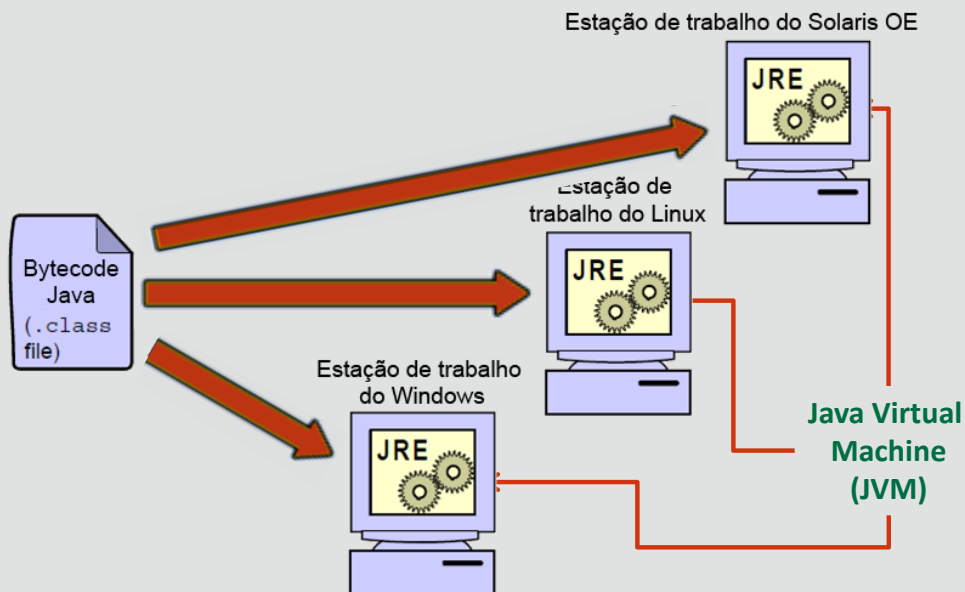
Um programa Java pode ser executado em várias combinações diferentes de CPUs e sistemas operacionais, como o SO Solaris em um chip SPARC e o MacOS X e o Microsoft Windows em um chip Intel, geralmente com poucas ou nenhuma modificação.

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



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)

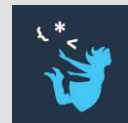
# Ambiente de Desenvolvimento Integrado (IDE, Integrated Development Environment)

- Finalidade:

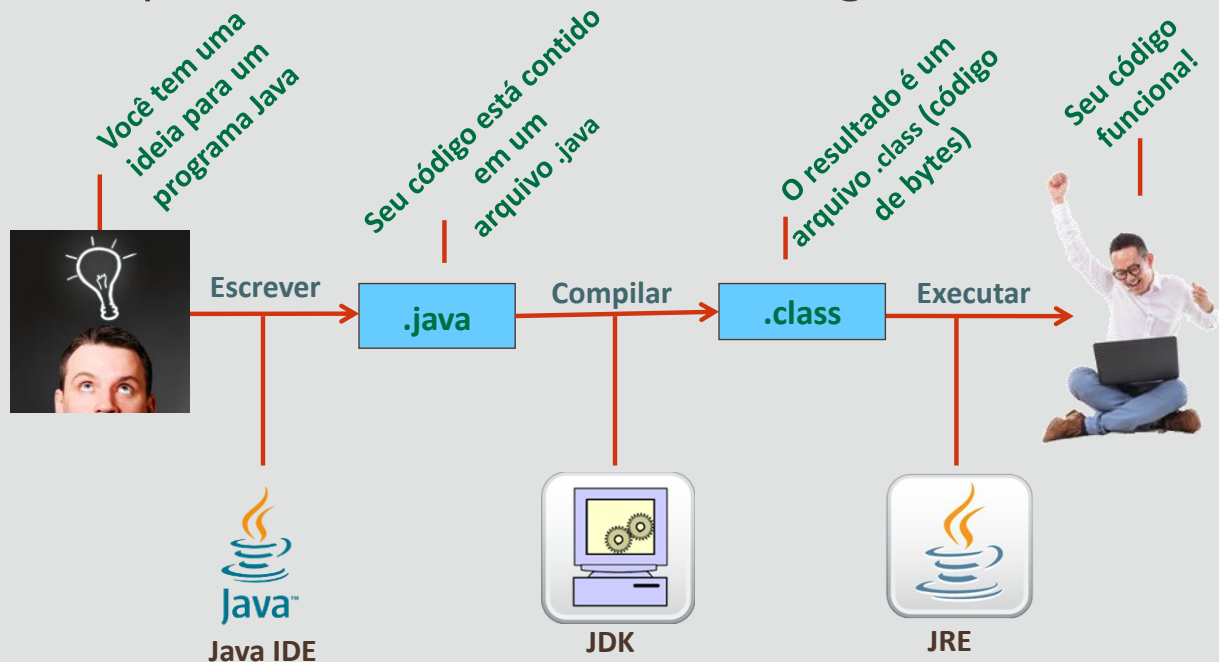
- Fornecer um editor de texto sofisticado
- Oferecer um código de depuração de assistência
- Gerenciar projetos
- Escrever o código-fonte (.java)

- Exemplos:

- NetBeans
- Greenfoot e BlueJ
- Alice
- Eclipse



# Compilando e Executando um Programa Java



**ORACLE**  
Academy

JFo 1-1  
Sobre este Curso

Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

13

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 Java IDE, 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.

## Como trabalhar com arquivos de código existentes

- Ao longo deste curso, o código de amostra e o código inicial do projeto são fornecidos como arquivos .java
- Faça esta lição até o fim, pois ela demonstra como adicionar arquivos .java existentes a um projeto em IDEs Java comumente usados. (Se você estiver usando outro IDE Java, consulte a documentação do IDE para obter instruções sobre como fazer isso)



## 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 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.
  - Adicione o arquivo .java existente a um projeto Java





