

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

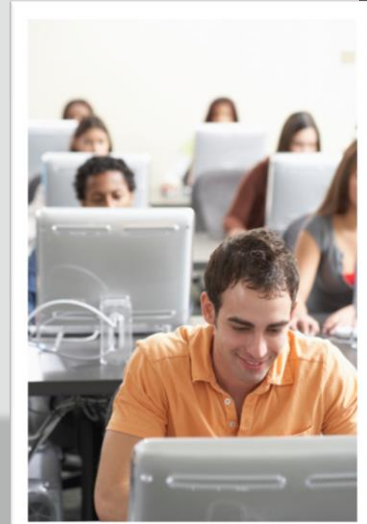
## Academy

# Java Foundations

4-2

Os Pacotes e a Declaração `import`

**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:
  - Acessar uma classe usando o respectivo nome totalmente qualificado
  - Descrever a função da instrução `import`
  - Usar a instrução `import` para acessar uma classe em um pacote
  - Entender a finalidade de um asterisco em uma instrução `import`
  - Identificar pacotes que são importados automaticamente



## Por que Você Precisa Reinventar a Roda?

- Pode ser que você precise escrever de novo o mesmo código frequentemente para programas diferentes
- Como alternativa para escrever novamente o mesmo código, você pode usar a biblioteca fornecida pelo Java, que organiza o código usado com frequência
- Essa biblioteca denomina-se biblioteca de classes Java
- A documentação da biblioteca de classes Java está disponível aqui:

– <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/module-summary.html>

# Pacotes na Biblioteca de Classes Java

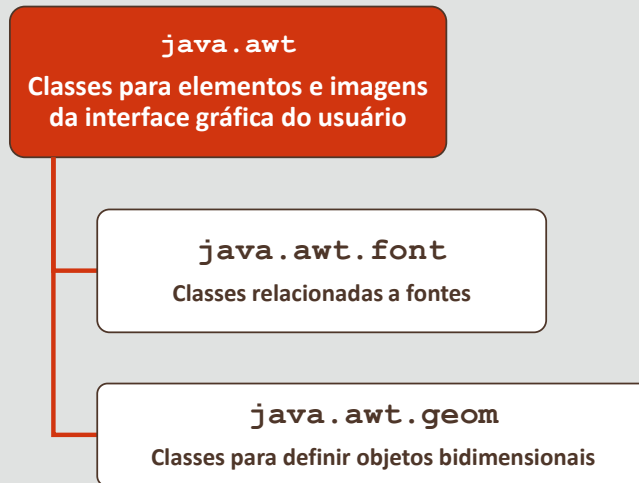
- As classes da biblioteca de classes Java são organizadas em pacotes
- Um pacote contém um grupo de classes relacionadas
- Com um pacote, fica mais fácil localizar as classes relacionadas

# Pacotes na Biblioteca de Classes Java

Pacote	Finalidade
<code>java.lang</code>	Fornecer classes que são fundamentais para o design da linguagem Java
<code>javax.swing</code>	Fornecer classes para construir componentes da interface gráfica do usuário
<code>java.net</code>	Fornecer classes para aplicativos de rede
<code>java.time</code>	Fornecer classes para datas, horas, instantes e durações

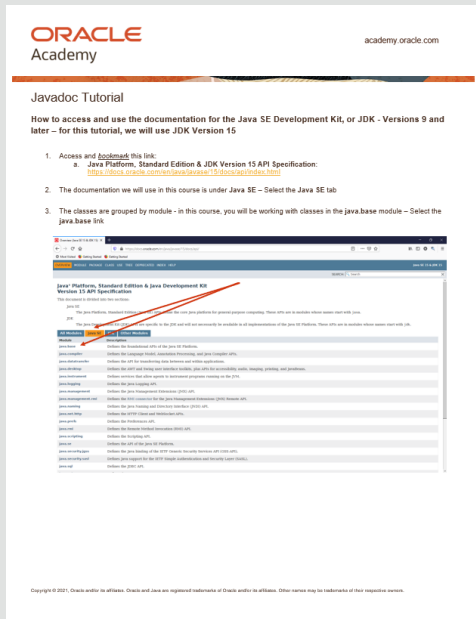
# Como os Pacotes São Organizados?

- A vasta coleção de classes é organizada em uma hierarquia semelhante a uma árvore, que permite aos pacotes serem divididos em subpacotes, como este:



# Tutorial do Javadoc

- Na Seção 4 da Programação de Estudo do Java Foundations:
  - Acesse e complete o Tutorial do Javadoc





## Usando uma Classe de um Pacote

- Para usar uma classe de um pacote no seu programa, você precisa especificar seu nome totalmente qualificado
- Por exemplo, para usar a classe `Scanner` a fim de ler uma entrada do teclado, o nome totalmente qualificado da classe `Scanner`, que é definido no pacote `java.util`, é

`java.util.Scanner`

**Pacote**                      **Nome da Classe**

## Usando o Nome de Classe Totalmente Qualificado

- Como você pode ver, o uso de um nome totalmente qualificado cria nomes muito longos para as classes
- Os nomes longos prejudicam a legibilidade do código e dificultam a codificação

```
public static void main(String[] args) { Nome de Classe Totalmente Qualificado
    int num;
    java.util.Scanner keyboard = new java.util.Scanner(System.in);
    System.out.print("Insira um número");
    num = keyboard.nextInt();
    System.out.print("O número que você inseriu é" + num);
} //fim do método main
```

## Existe uma Alternativa para o Nome Totalmente Qualificado?

- Suponha que você tenha um amigo cujo nome seja Santi Inez Luis Vidal
  - Seria complicado chamá-lo sempre pelo nome completo?
  - Se você pudesse simplesmente chamá-lo de “Santi,” seria bem melhor
- Da mesma forma, acessar classes Java usando nomes totalmente qualificados é igualmente tedioso
- Vamos ver se existe uma maneira de especificar apenas o nome da classe, em vez de usar seu nome totalmente qualificado

## Usando a instrução `import`

- Você pode evitar o nome de classe totalmente qualificado usando a instrução `import`
- Insira a instrução `import` acima da sua definição de classe, este será o formato:

- `import package.className`

- Exemplo:

```
import java.util.Scanner;

public class AddNums {
    //o código da classe entra aqui
} //fim do método AddNums
```

## Como Você Importa uma Única Classe?

- Você pode importar uma única classe ou um pacote inteiro
- Para importar uma única classe para seu programa, escreva uma instrução `import` como esta:

```
import javax.swing.JOptionPane;
```

Nome do Pacote

Nome da Classe

A palavra-chave `import` seguida do nome do pacote, ponto e o nome da classe

## Pacote `javax.swing`

- O Java tem uma ampla biblioteca para construir interfaces gráficas do usuário
- Esta biblioteca, denominada `swing`, pode ser importada para seu programa, a fim de conceder acesso à funcionalidade da interface gráfica do usuário do Java
- A biblioteca `swing` está no pacote `javax.swing`

## Acessando uma Classe por meio do Pacote swing

- O pacote `swing` tem uma classe `JOptionPane`
- Ela cria uma janela pop-up que pode ser usada para exibir strings de texto para o usuário
- Para usar a classe `JOptionPane`, primeiro você deve importá-la para sua classe:

```
import javax.swing.JOptionPane;

public class Welcome {
    //o código da classe entra aqui
} //end class Welcome
```

Instrução `import` que importa uma única classe, `JOptionPane`, do pacote `swing`

# Importando a Classe JOptionPane

- Você pode usar `JOptionPane` para exibir o texto chamando o método `showMessageDialog` dentro da classe `JOptionPane`

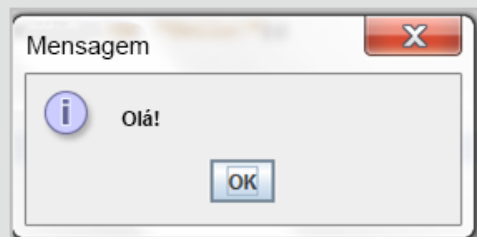
```
import javax.swing.JOptionPane;

public class Welcome {

    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "Olá!");
    } //fim do método main
} //fim da classe Welcome
```



# Esta é a Aparência da Saída!



## Como Você Importa Todas as Classes em um Pacote?

- Você pode importar todas as classes de determinado pacote usando o caractere curinga `*` na instrução `import`

## Como Você Importa Todas as Classes em um Pacote?

- Suponha que você queira ampliar o exemplo anterior criando uma instância da classe `JFrame` e adicionar sua referência a `JOptionPane`, desta maneira:

```
import javax.swing.JOptionPane;
import javax.swing.JFrame;

public class Welcome {

    public static void main(String[] args) {
        JFrame frame = new JFrame();
        JOptionPane.showMessageDialog(frame, "Olá!");
    } // fim do método main
} // fim da classe Welcome
```

Importando duas classes do pacote `swing`

## Acessando Todas as Classes por meio do Pacote `swing`

- À medida que você acessa mais classes do pacote `swing` no seu programa, o número de instruções `import` também aumenta

## Acessando Todas as Classes por meio do Pacote swing

- Para evitar isso, você pode importar todas as classes do pacote `swing` usando o caractere curinga `*` na instrução `import`, como é mostrado a seguir:

```
import javax.swing.*;
```

Substitua todas as instruções `import` do exemplo anterior por uma instrução `import`

```
public class Welcome {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame();  
        JOptionPane.showMessageDialog(frame, "Olá!");  
    } // fim do método main  
} // fim da classe Welcome
```

## Incluindo Várias Instruções `import`

- Você pode incluir várias instruções `import` em um programa Java para acessar classes no mesmo pacote ou em pacotes diferentes
- Por exemplo:

```
import java.util.Date;  
import java.util.Calendar; }  
import javax.swing.*;
```

Importando classes do mesmo pacote

```
public class DisplayDate {  
    //a definição da classe entra aqui  
} //fim da classe DisplayDate
```

Importando classes de pacotes diferentes

## Identificar Pacotes que São Importados Automaticamente

- Até o momento, você usou `System.out.println()` para imprimir texto no console

```
public class DisplayOutput {  
    public static void main(String[] args) {  
        System.out.println("Olá, como você está?");  
    } //fim do método main  
} //fim da classe DisplayOutput
```

- No entanto, você não importou um pacote para usar esse método no seu programa
- Então como o Java saberá o que fazer quando você chamá-lo?

## Pacote `java.lang`

- Se você observar a biblioteca Java, verá que a classe `System` está organizada no pacote `java.lang`
- Por padrão, o pacote `java.lang` é importado automaticamente para todos os programas Java



## Exercício 1

- Crie um novo projeto e adicione o arquivo `AddImport.java` a ele
- Examine `AddImport.java`
  - Faça o seguinte:
  - Substitua o nome totalmente qualificado para acessar o componente `JLabel` por uma instrução `import`
  - Para importar classes do pacote `util`, substitua várias instruções `import` por uma única instrução `import`

# Resumo

- Nesta lição, você deverá ter aprendido a:
  - Acessar uma classe usando o respectivo nome totalmente qualificado
  - Descrever a função da instrução `import`
  - Usar a instrução `import` para acessar uma classe em um pacote
  - Entender a finalidade de um asterisco em uma instrução `import`
  - Identificar pacotes que são importados automaticamente



**ORACLE**  
Academy

JFo 4-2  
Os Pacotes e a Declaração Import

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.

26

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