




# Módulo Lógica de Programação com aplicações em Java

Projeto khouse Profissionalizante  
Profª Larissa Brandão



**Capturando dados  
através do teclado**

## Objetivos:



Criar códigos para a captura de dados pelo teclado

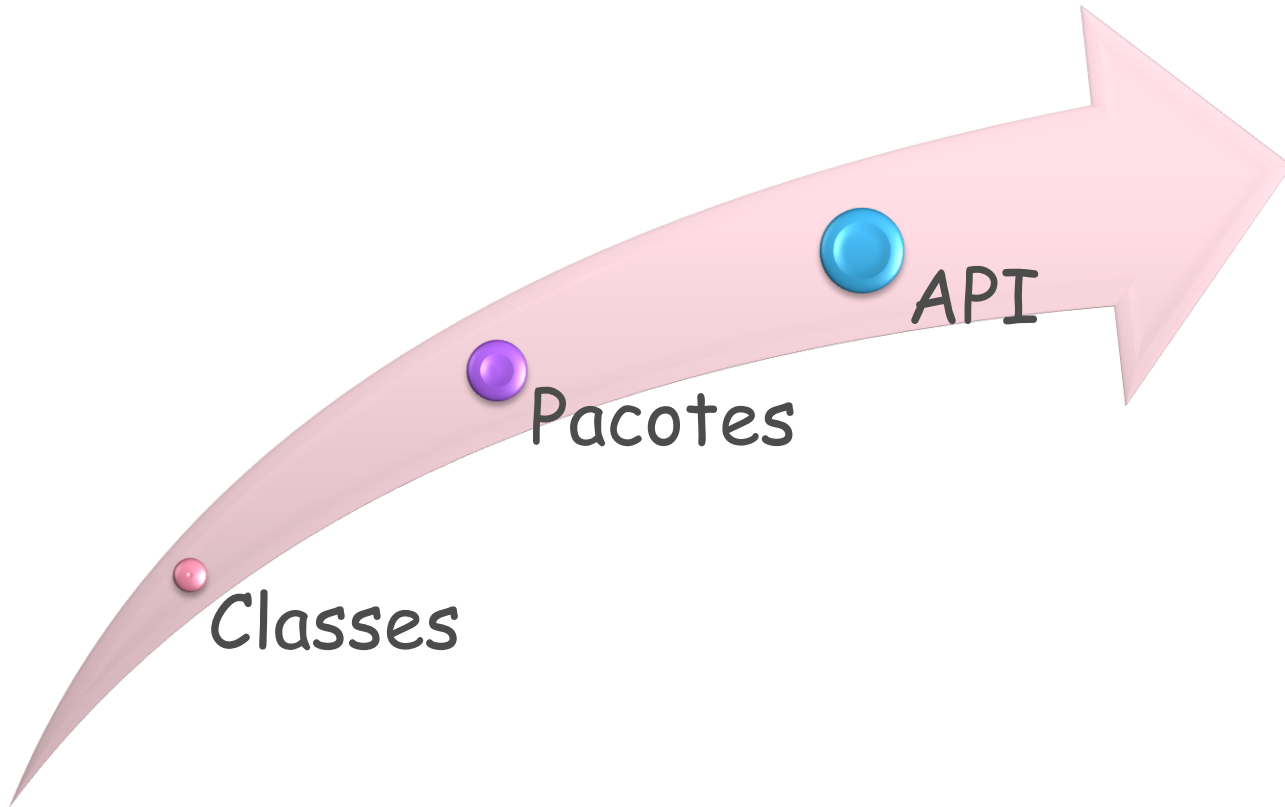
Utilizar a classe Scanner para captura, através de uma janela de console, de dados digitados no teclado

Utilizar a classe JOptionPane para captura, através de uma interface gráfica, de dados digitados no teclado



API

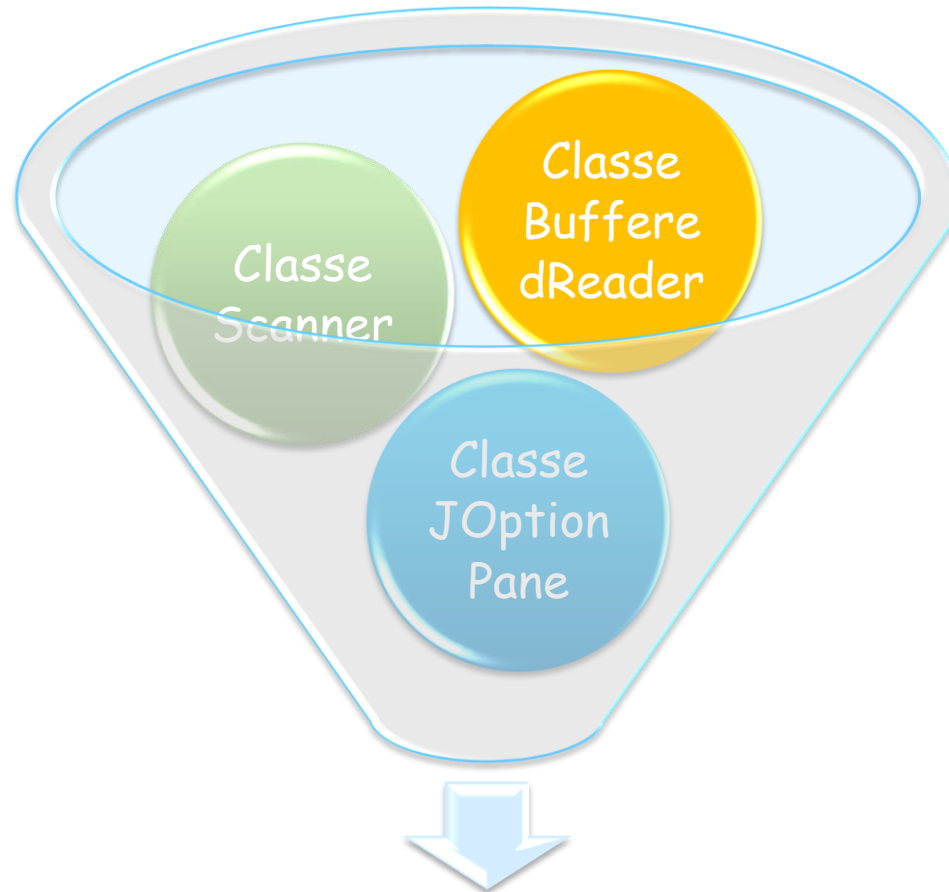
Interfaces de  
Programação de  
Aplicações, ou Application  
Programming Interface  
(API) contêm centenas de  
classes  
pré-definidas que se pode  
utilizar no programas



Classes

Pacotes

API



Capturando entrada através  
do teclado

Classe  
Buffere  
dReader

java.io

1. Digite as seguintes instruções no início do programa:

```
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.io.IOException;
```

2. Adicione as seguintes instruções no método main:

```
BufferedReader dataIn = new BufferedReader(new  
InputStreamReader(System.in));
```

3. Declare uma variável temporária do tipo `String` para receber a entrada de dados e chame o método `readLine()` para capturar, linha a linha, o que for digitado. Deve ser colocado em um bloco `try-catch`:

```
try {  
String temp = dataIn.readLine();  
} catch(IOException e) {  
System.out.println("Error in getting input");  
}
```



Classe  
Buffered  
Reader

java.io

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
```

```
public class ExemploBufferedReader {
```

```
    public static void main(String[] args) {
        BufferedReader dataIn = new
        BufferedReader(new
        InputStreamReader(System.in));
```

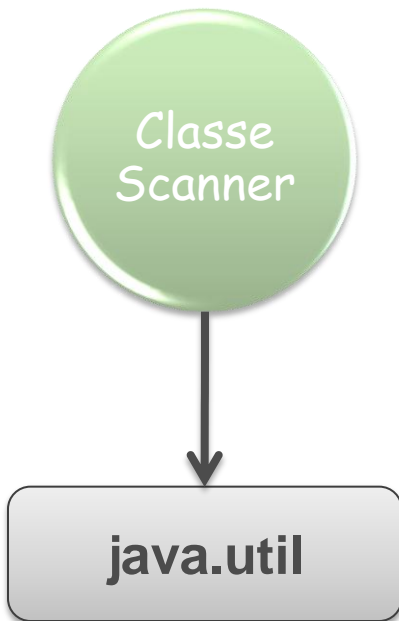
```
        String nome = "";
        System.out.print("Digite seu nome:");
```

```
        try {
            nome = dataIn.readLine();
        } catch (IOException e) {
            System.out.println("Error!");
        }
```

```
        System.out.println("Olá " + nome + "!");
```

```
    }
}
```





1. Digite as seguintes instruções no início do programa:  

```
import java.util.Scanner;
```
2. Adicione as seguintes instruções no método main:  

```
Scanner sc = new Scanner(System.in);
```
3. Utilize um dos métodos abaixo para capturar os dados:

Método	Finalidade
<code>next()</code>	Aguarda uma entrada em formato String
<code>nextInt()</code>	Aguarda uma entrada em formato Inteiro
<code>nextByte()</code>	Aguarda uma entrada em formato Inteiro
<code>nextLong()</code>	Aguarda uma entrada em formato Inteiro Longo
<code>nextFloat()</code>	Aguarda uma entrada em formato Número Fracionário
<code>nextDouble()</code>	Aguarda uma entrada em formato Número Fracionário

```
graph TD; A((Classe Scanner)) --> B(java.util);
```

Classe  
Scanner

java.util

```
import java.util.Scanner;
```

```
public class ExemploScanner{  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Digite seu nome:");  
        String nome = sc.next();  
        System.out.println("Olá " + nome + "!");  
    }  
}
```

Classe  
JOptionPane

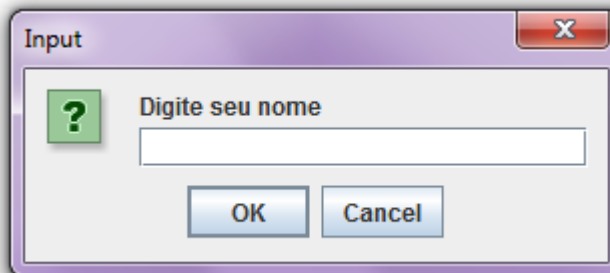
javax.swing

A `JOptionPane` possui métodos que permitem mostrar uma caixa de diálogo para que o usuário informe os dados.

A instrução:

```
nome=JOptionPane.showInputDialog("Digite seu nome:");
```

cria uma caixa de entrada, que exibirá um diálogo com uma mensagem, um campo de texto, para receber os dados do usuário, e um botão OK, como mostrado na figura



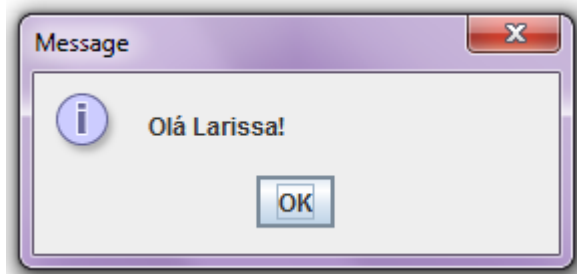
Classe  
JOptionPane

javax.swing

A instrução:

```
JOptionPane.showMessageDialog(null, msg);
```

exibirá um diálogo contendo a mensagem e o botão de OK.



Classe  
JOptionPane

javax.swing

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

```
public class ExemploJOptionPane {  
    public static void main( String[] args ) {  
        String nome = "";  
        nome = JOptionPane.showInputDialog("Digite seu  
        nome");  
        String msg = "Olá " + nome + "!";  
        JOptionPane.showMessageDialog(null, msg);  
    }  
}
```