



Introdução a Linguagem Java

Prof^a. Rachel Reis
rachel.@inf.ufpr.br



Linguagem Java

- Origem

- Linguagem desenvolvida pela *Sun Microsystems*
- Sintaxe similar ao C++

- Principais características

- Orientada a objetos
- Gerência automática de memória
- Portabilidade





Plataforma Java

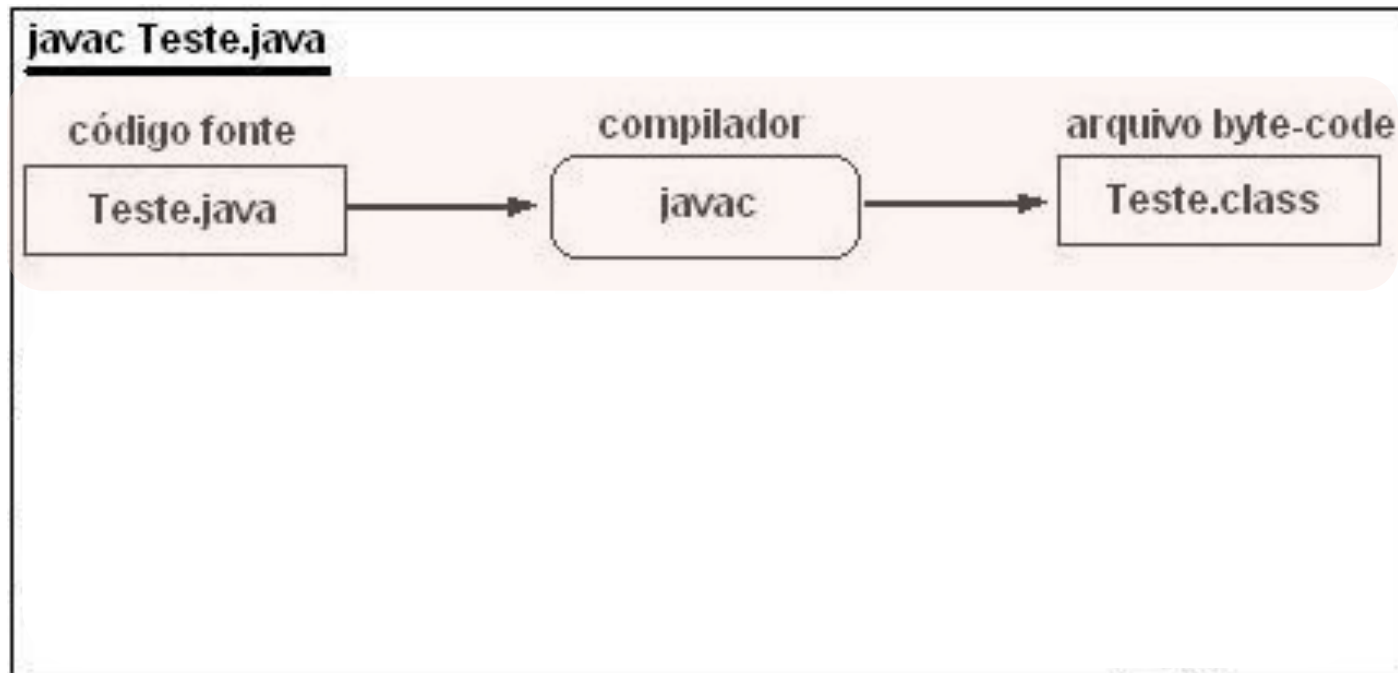
- Plataforma
 - Ambiente de *hardware* e/ou *software* no qual um programa é executado
- A plataforma Java é somente de *software*
 - Esta plataforma de execução funciona sobre outras plataformas de *hardware* e *software*.



Plataforma Java

- Composta por dois ambientes:
 - Ambiente de desenvolvimento – **JDK** (Java Development Kit)
 - Ambiente de execução – **JRE** (*Java Runtime Environment*)

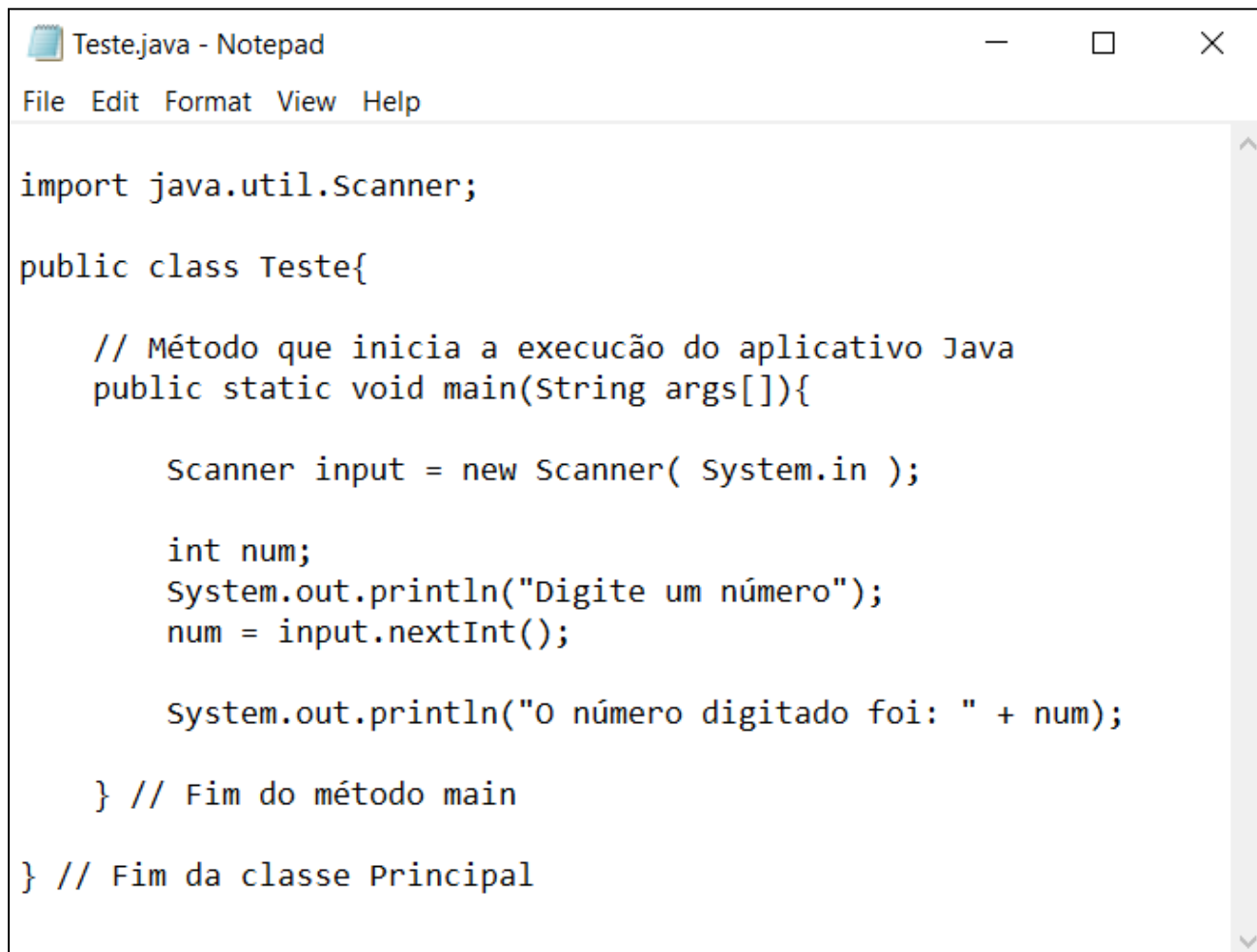
Ambiente de Desenvolvimento - JDK





Exemplo de código fonte

Teste.java



```
Teste.java - Notepad
File Edit Format View Help

import java.util.Scanner;

public class Teste{

    // Método que inicia a execução do aplicativo Java
    public static void main(String args[]){

        Scanner input = new Scanner( System.in );

        int num;
        System.out.println("Digite um número");
        num = input.nextInt();

        System.out.println("O número digitado foi: " + num);

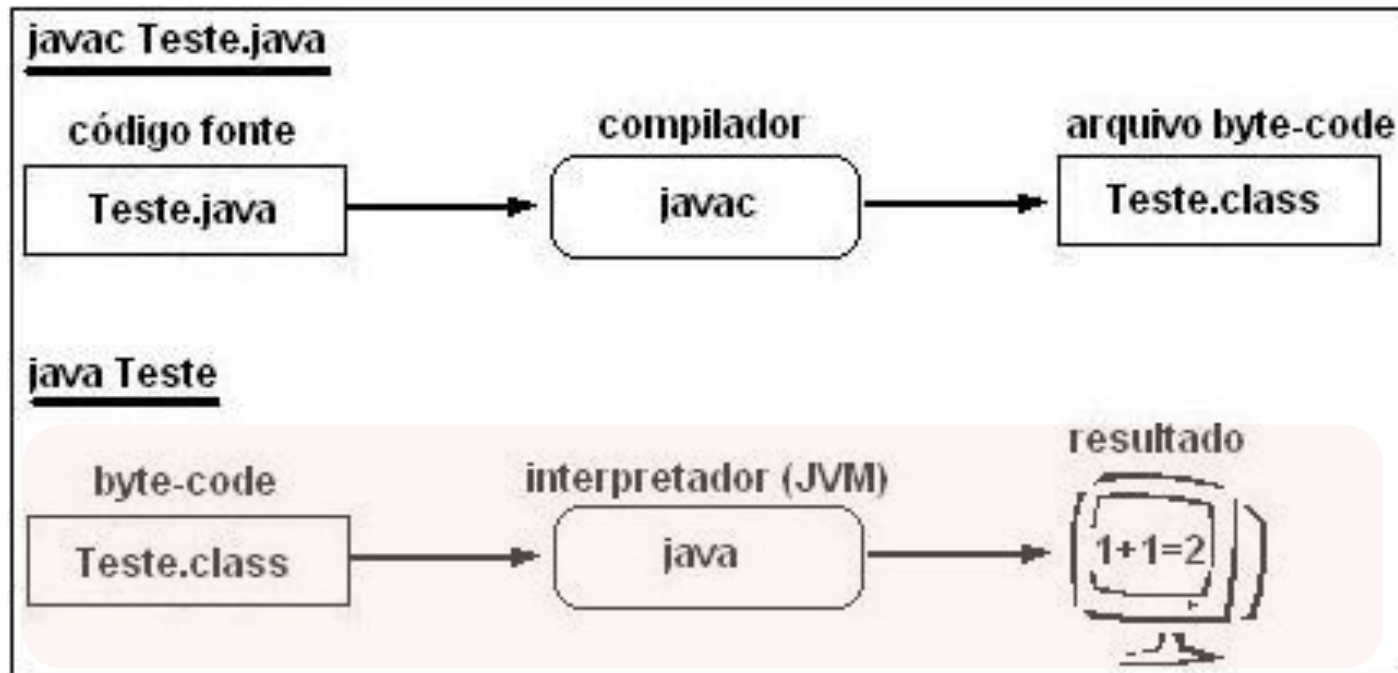
    } // Fim do método main
} // Fim da classe Principal
```

```

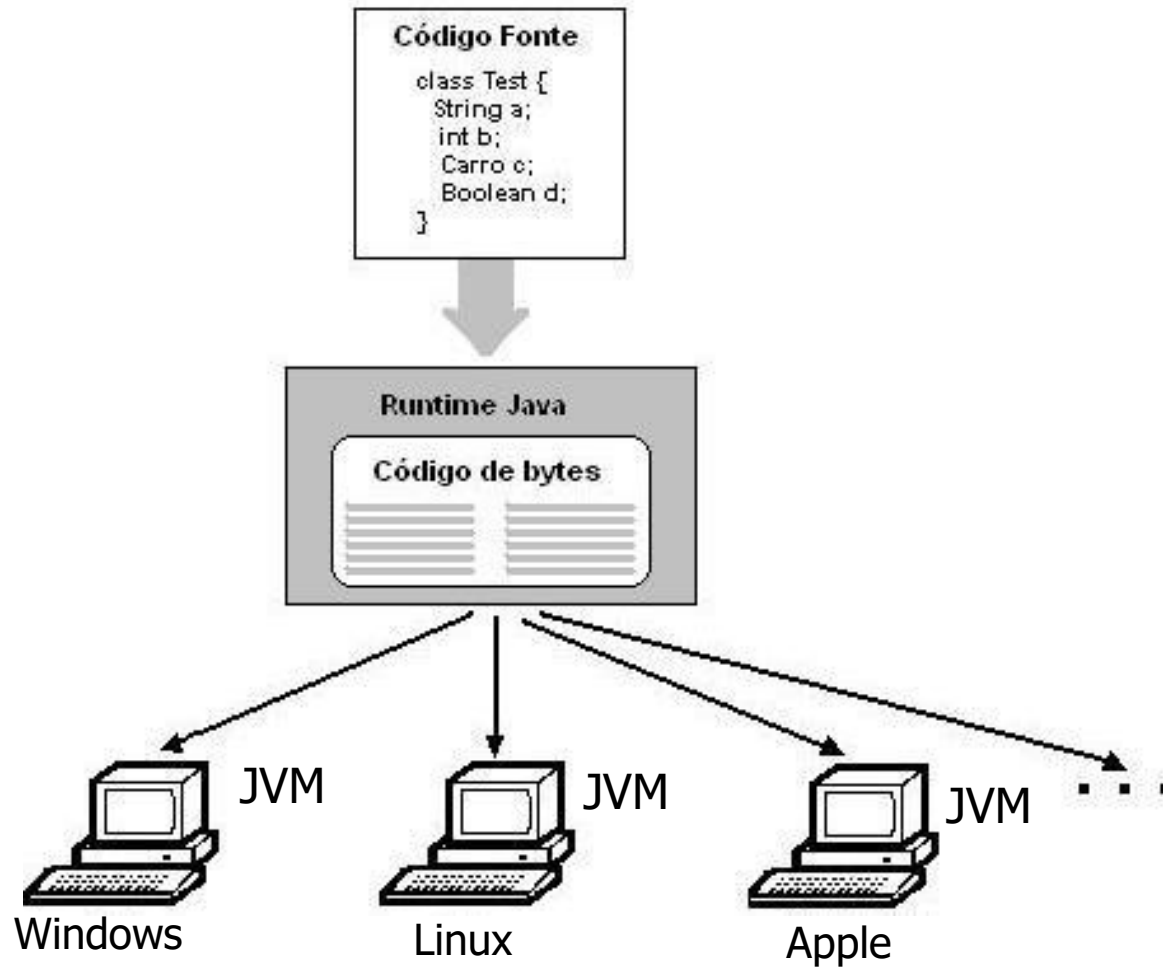
Teste.class - Notepad
File Edit Format View Help
Ëpº% 7 F
[] []
! "" #
$ %
& * + , <init> ()V []Code []LineNumberTable []LocalVariableTable []this []LTeste;[]
[]main []([Ljava/lang/String;)V []args [][Ljava/lang/String;[] []input []Ljava/util/Scanner;[] []num[]
[]I[]
SourceFile []Teste.java^ ^
[] []java/util/Scanner -^ . / ^ ^ 0^ 1 2 []Digite um nÃºmero[] 3^ 4 5^ 6 7 []BootstrapMethods[]
8[] 9^ : ;[] Teste[] []java/lang/Object []java/lang/System in[] []Ljava/io/InputStream;[] []
(Ljava/io/InputStream;)V []out []Ljava/io/PrintStream;[] []java/io/PrintStream []println[] []
(Ljava/lang/String;)V []nextInt[] ()I
< =[] []O nÃºmero digitado foi: [] []makeConcatWithConstants [](I)Ljava/lang/String;[] >^ : B[]
$java/lang/invoke/StringConcatFactory[] D[] []Lookup[] ^InnerClasses[] ~
(Ljava/lang/invoke/MethodHandles
$Lookup;Ljava/lang/String;Ljava/lang/invoke/MethodType;Ljava/lang/String;
[Ljava/lang/Object;)]Ljava/lang/invoke/CallSite;[] E[] %java/lang/invoke/MethodHandles
$Lookup[] java/lang/invoke/MethodHandles !
[] [] ^
[] [] / [] [] []*. []± [] [] [] [] ^ [] [] [] [] s [] [] %» Y² []· []L² [][]
[]+[] []=² []º []± [] [] [] [] ^ []
[] [] $ [] [] [] % [] [] [] [] [] []
[] [] [] A
[] ? C @ [] ' [] [] ( [] )

```

Ambiente de Execução - JRE



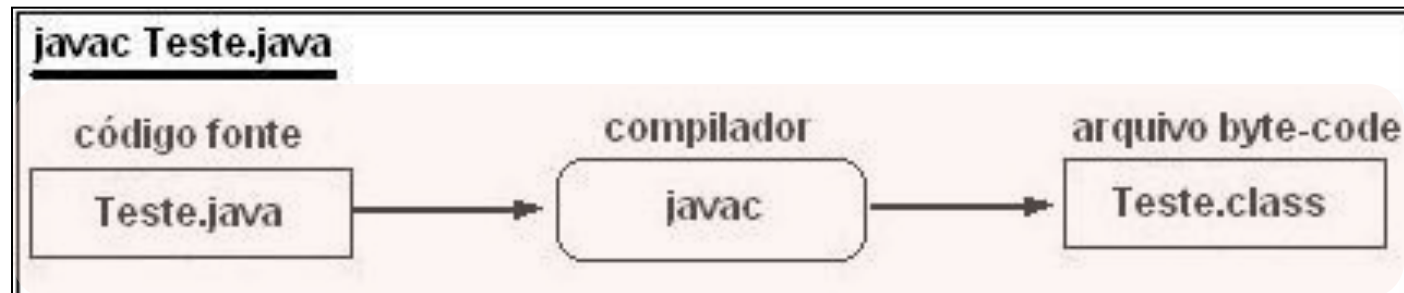
Ambientes Desenvolvimento e Execução



Resumo JDK, JRE, JVM

- **JDK** (ambiente de desenvolvimento java):
 - é necessário para desenvolver softwares Java.
 - Inclui a **JRE** e ferramentas de programação, como:
 - **javac** - compilador
 - **jar** - empacotador
 - **javadoc** - ferramenta para geração de documentação

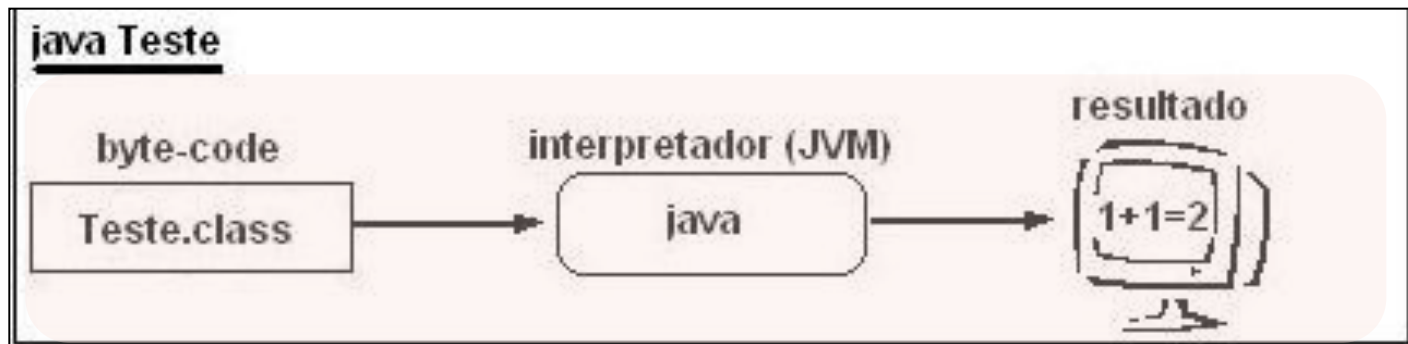
Exemplo:



Resumo JVM, JRE, JDK

- **JRE** (ambiente de execução java):
 - Consiste na **JVM** e na Java **API**, que são as bibliotecas de classes presentes em todos os ambientes de produção Java
- **JVM** (*Java Virtual Machine*):
 - Máquina virtual que interpreta e executa o código Java compilado (*bytecode*).

Exemplo:



Plataformas de Desenvolvimento

- Java conta com três plataformas de desenvolvimento conhecidas:

- **JSE,**
- **JEE,**
- **JME.**



- Cada um dessas plataformas possui suas bibliotecas (APIs) específicas.



Plataformas Java

JSE

Soluções para
escritório

- Kit básico
- Aplicações independentes
- Aplicações desktop, notebook

JEE

Soluções para
empresas

- Aplicações empresariais
- Comércio eletrônico
- Aplicações web

JME

Soluções para
dispositivos móveis

- Aplicações móveis ou portáteis
- Celulares, palmtops, pagers



IDE

- Existem vários ambientes integrados de desenvolvimento para Java

- **Replit**



- NetBeans



- Eclipse




- BlueJ





Link: <https://replit.com>



Log in to your account


Email or Username


Password


Log in →

New to Replit? [Sign up](#)

[Forgot password?](#)

 Continue with Google

 Continue with Github

 Continue with Facebook

By continuing, you agree to Replit's [Terms of Service](#) and [Privacy Policy](#), and to receiving emails with updates.

Trouble logging in? [Get help](#)



Exemplo 1

- Criar um programa em Java que imprima a mensagem:

Oi! Seja bem vindo!



Exemplo 1

```
public class Principal
{
    public static void main(String args[])
    {
        System.out.println("Oi! Seja bem vindo!");
    }
}
```



Exemplo 1

- Todo programa Java é formado por pelo menos uma declaração de classe.

```
public class Principal
{
    public static void main(String args[])
    {
        System.out.println("Ola! Seja bem vindo!");
    }
}
```



Exemplo 1

- Método main(): ponto de partida de um aplicativo Java.

```
public class Principal
{
    public static void main(String args[])
    {
        System.out.println("Ola! Seja bem vindo!");
    }
}
```



Instrução de Saída

- System.out: objeto de saída padrão.

```
public class Principal
{
    public static void main(String args[])
    {
        System.out.println("Ola! Seja bem vindo!");
    }
}
```



Instruções de Saída

- Qual a diferença entre as instruções de saída abaixo?
 - `System.out.print(...);`
 - `System.out.println(...);`
 - `System.out.printf(...);`



Exemplo 2

- Escreva um programa m Java para ler um número inteiro digitado pelo usuários. Em seguida, exiba o número na tela.

- Exemplo:

- Entrada

Digite um numero: 45

- Saída

Numero digitado: 45

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in) ;
        int num;
        System.out.print("Digite um numero: ");

        num = input.nextInt() ;

        System.out.println("Numero digitado: " + num) ;

        input.close() ;
    }
}
```

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

```
/*
```

- O **import** é usado para o compilador localizar uma classe.
- No exemplo acima, o comando faz o import da classe Scanner para que ela seja usada no programa.

```
*/
```



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

```
public class Principal{
```

```
    public static void main(String[] args)
    {
```

```
        /*
```

A classe Principal contém o método main() que é responsável por iniciar o programa Java.

```
        */
```

```
    }
```

```
}
```

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

```
public class Principal{
```

```
    public static void main(String[] args)
    {
```

```
        Scanner input = new Scanner(System.in) ;
```

```
        /*
```

A declaração acima **cria o objeto input** para ler os dados fornecidos pelo usuário.

```
        */
```

```
    }
```

```
}
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in) ;
        int num;

        /*
            A variável num será usada para armazenar um
            número inteiro.
        */

    }
}
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in) ;
        int num;
        System.out.print("Digite um numero: ");

        /*
            A instrução acima exibe uma mensagem na tela.
        */

    }
}
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in) ;
        int num;
        System.out.print("Digite um numero: ");

        num = input.nextInt() ;

        /*
        O método nextInt() lê o primeiro número
        digitado pelo usuário e atribui a variável num.
        */

    }
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in) ;
        int num;
        System.out.print("Digite um numero: ");

        num = input.nextInt() ;

        System.out.println("Numero digitado: " + num) ;

        /*
            A instrução acima exibe uma mensagem na tela.
        */
    }
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in) ;
        int num;
        System.out.print("Digite um numero: ");

        num = input.nextInt() ;

        System.out.println("Numero digitado: " + num) ;

        input.close() ;
    }
}
```

/*

O método **close()** fecha o
objeto input.

*/



Tipos de Dados em Java

- Tipo lógico: *boolean*
- Tipo textual: *char* e *String*
- Tipo ponto flutuante: *float* ou *double*
- Tipo inteiro: *byte*, *short*, *int* e *long*

8 bits	<i>byte</i>	$-2^7 \dots 2^7 - 1$
16 bits	<i>short</i>	$-2^{15} \dots 2^{15} - 1$
32 bits	<i>int</i>	$-2^{31} \dots 2^{31} - 1$
64 bits	<i>long</i>	$-2^{63} \dots 2^{63} - 1$



Pesquisar

- Instrução para ler um número inteiro (usando Scanner):

```
int num;
```

```
num = input.nextInt() ;
```

- Pesquise as instruções para ler:
 - Um dado do tipo float
 - Um dado do tipo char
 - Um dado do tipo String



Exemplo 3

- Escreva um programa em Java que leia dois números digitados pelo usuário. Em seguida, imprima os números em ordem crescente.

- Exemplo:

- Entrada

Digite um numero: 10

Digite outro numero: 5

- Saída

5 10

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

Vamos precisar desse
import?

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

```
public class Principal{
```

```
    public static void main(String[] args)  
    {
```

Vamos precisar dessa
estrutura?

```
}
```

```
}
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in) ;
    }
}
```

Temos que criar um objeto
da classe Scanner?

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in) ;
        int ...
```

De quantas variáveis do
tipo inteiro vamos precisar?

```
}
```

```
}
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in);
        int num1, num2;
```

De quantas variáveis do
tipo inteiro vamos precisar?

```
}
```

```
}
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in);
        int num1, num2;

        System.out.print("Digite um numero: ");
        num1 = teclado.nextInt();
```

Vamos ler o primeiro
numero...


```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in);
        int num1, num2;

        System.out.print("Digite um numero: ");
        num1 = teclado.nextInt();
        System.out.print("Digite outro numero: ");
        num2 = teclado.nextInt();
    }
}
```

Agora lendo o segundo
numero...

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        ...

        if (num1 < num2) {

        }
        else{

        }

    }
}
```

Usando o comando if-else

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        ...

        if (num1 < num2) {
            System.out.println(num1 + " " + num2);
        }
        else{

        }

    }
}
```

se num1 < num2

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        ...

        if (num1 < num2) {
            System.out.println(num1 + " " + num2);
        }
        else{
            System.out.println(num2 + " " + num1);
        }

    }
}
```

se num2 <= num1

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        ...

        if(num1 < num2){
            System.out.println(num1 + " " + num2);
        }
        else{
            System.out.println(num2 + " " + num1);
        }
        teclado.close();
    }
}
```

Fechando o objeto teclado



Exemplo 4

- Escreva um programa em Java que imprima a tabuada de multiplicação de um número inteiro no intervalo [1,10] .

- Exemplo:

- Entrada

Digite um numero: 7

- Saída

$$7 * 1 = 7$$

$$7 * 2 = 14$$

...

$$7 * 10 = 70$$

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

Vamos precisar desse
import?

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

```
public class Principal{
```

```
    public static void main(String[] args)  
    {
```

Vamos precisar dessa
estrutura?

```
}
```

```
}
```



```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in) ;
    }
}
```

Temos que criar um objeto
da classe Scanner?

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in);
        int num;
```

Vamos precisar de uma
variável do tipo inteiro?

```
}
```

```
}
```

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in);
        int num;

        System.out.print("Digite um numero: ");
        num = teclado.nextInt();
    }
}
```

Lendo o número digitado
pelo usuário...

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args){
        Scanner teclado = new Scanner(System.in) ;
        int num, cont;

        System.out.print("Digite um numero: ");
        num = teclado.nextInt();

        cont = 1;
        while(cont <= 10) {

        }

    }
}
```

Usando o comando
while

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args){
        Scanner teclado = new Scanner(System.in) ;
        int num, cont, mult;

        System.out.print("Digite um numero: ") ;
        num = teclado.nextInt() ;

        cont = 1;
        while(cont <= 10) {
            mult = num * cont;

        }

    }
}
```

Fazendo o cálculo da
mutiplicação

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args){
        Scanner teclado = new Scanner(System.in);
        int num, cont, mult;

        System.out.print("Digite um numero: ");
        num = teclado.nextInt();

        cont = 1;
        while(cont <= 10){
            mult = num * cont;
            System.out.println(num+ "*" +cont+"="+mult);

        }
    }
}
```

Exibindo a saída para o usuário do programa

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args){
        Scanner teclado = new Scanner(System.in);
        int num, cont, mult;

        System.out.print("Digite um numero: ");
        num1 = teclado.nextInt();

        cont = 1;
        while(cont <= 10){
            mult = num * cont;
            System.out.println(num+ "*" +cont+"="+mult);
            cont++;
        }
    }
}
```

Incrementando a
variável cont

```
import java.util.Scanner;

public class Principal{

    public static void main(String[] args){
        Scanner teclado = new Scanner(System.in) ;
        int num, cont, mult;

        System.out.print("Digite um numero: ");
        num1 = teclado.nextInt() ;

        cont = 1;
        while(cont <= 10) {
            mult = num * cont;
            System.out.println(num+ "*" +cont+"="+mult) ;
            cont++;
        }
        teclado.close() ;
    }
}
```

Fechando o objeto
teclado



While → For

- Reescreva o código abaixo usando a estrutura de repetição for.

```
cont = 1;
while(cont <= 10) {
    mult = num * cont;
    System.out.println(num+"*"+cont+"="+mult) ;
    cont++;
}
```



Praticar...

- Construa um programa em Java que leia um número inteiro e diga se ele é par ou ímpar.
- Faça um programa em Java que leia números inteiros enquanto não for digitado o número -1, e calcule e imprima a soma destes números
- Faça um programa em Java que calcule e imprima a soma dos 10 primeiros múltiplos de 3.



Referências

- Deitel, P. J.; Deitel, H. M. (2017). Java como programar. 10a edição. São Paulo: Pearson Prentice Hall.
- Barnes, D. J. (2009). Programação orientada a objetos com Java: uma introdução prática usando o BlueJ (4. ed.). São Paulo, SP: Prentice Hall.
- Boratti, I. C. (2007). Programação orientada a objetos em Java. Florianópolis, SC: Visual Books.