

Linguagem de Programação Plataforma JAVA

Profª Aline Brito

Adaptado do material da Profª Grace
Borges



Programa em Java

- Todo **programa Java** é uma classe:
 - Possui obrigatoriamente um método principal;
 - Método principal contém a seqüência lógica do programa;
 - Pode ter outros métodos;
 - Pode utilizar outras classes e objetos em suas instruções;
- Cuidado: nem toda classe é um programa;

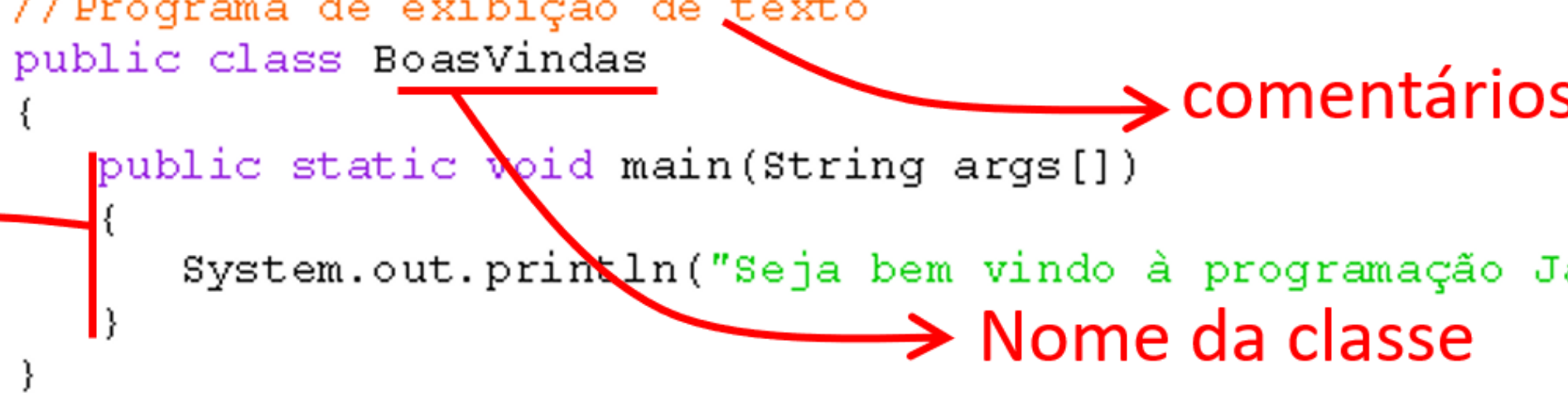
Estrutura de um Programa Java

```
public class Nome_da_Classe
{
    public static void main (String args[ ])
    {
        <bloco de instruções do programa>;
    }
}
```

Salvar como Nome_da_Classe.java

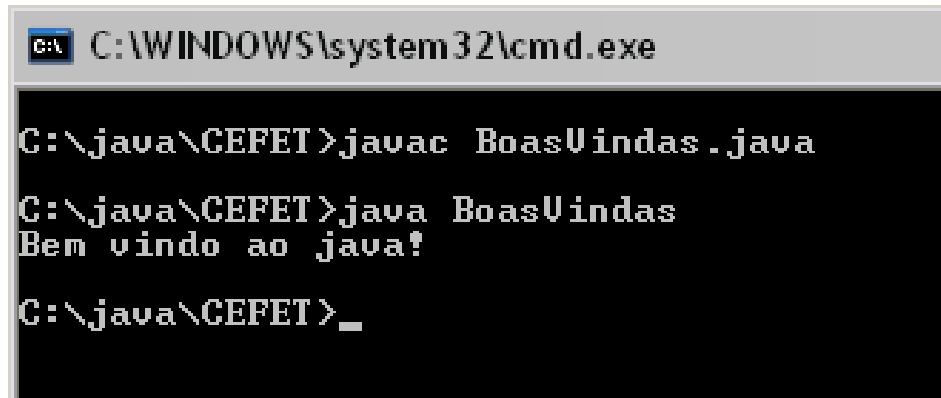
Programa BoasVindas.java

```
// Programa de exibição de texto
public class BoasVindas
{
    public static void main(String args[])
    {
        System.out.println("Seja bem vindo à programação Java");
    }
}
```



Exemplo de edição de programas

Primeiro programa JAVA



```
C:\WINDOWS\system32\cmd.exe

C:\java\CEFET>javac BoasVindas.java

C:\java\CEFET>java BoasVindas
Bem vindo ao java!

C:\java\CEFET>_
```

- Para compilar (NA LINHA DE COMANDO):
- <Drive e Diretório>\javac BoasVindas.java
- Para Interpretar (NA LINHA DE COMANDO):
- <Drive e Diretório>\java BoasVindas

```

1  /* Programa que imprime
2     uma mensagem de
3     boas vindas */
4
5  // definição da classe
6  public class BoasVindas
7  {
8      // método principal da classe
9      public static void main (String args[])
10     {
11         System.out.println ("Bem vindo ao java!");
12     }
13     } // fim do método principal
14
15 } // fim da classe

```

Linhas de comentário

- Usadas para documentar o programa
- Ignoradas pelo compilador
 - // comenta uma linha
 - /* múltiplas linhas */

Identificadores (classes e variáveis)

- Nome de classes
 - Por padrão, iniciar com maiúscula
 - Quando nome composto, iniciais maiúsculas
- Variáveis:
 - Iniciar com minúscula
- Regras gerais:
 - Caracteres permitidos: letras; dígitos; sublinhado; cifrão;
 - Não iniciar com dígitos;
 - Não conter espaço em branco;
- Ex.: BoasVindas1; \$value, _valor, cod_ent, botao7

SAÍDA PADRÃO: System.out

- Exibe caracteres/ dados na tela
- **string entre aspas ou variáveis**
- Formatos:
 - System.out.**print**(args); // cursor na mesma linha
 - System.out.**println**(args);// cursor para próx. linha

Seqüência de escape: caracteres especiais

- `System.out.println("Seja bem vindo\nà programação Java");`
- **Seja bem vindo**
- **à programação Java**

Seqüência de escape	Descrição
<code>\n</code>	Nova linha.
<code>\t</code>	Tabulação.
<code>\r</code>	Posiciona o curso no início da linha atual.
<code>\\</code>	Imprime uma barra invertida.
<code>\"</code>	Imprime aspas.

Entrada de dados – Classe Scanner

- Pertence ao pacote **java.util**
- Cláusula **import**: aparecer antes da definição da classe
- Se esquecermos, causa erro!

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

Exemplo

```
// declaração da variável input
```

```
Scanner input ;
```

```
// instanciando objeto
```

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

```
// chamando um método do objeto input
```

```
num1 = input.nextInt( );
```

```
num2 = input.nextFloat( );
```

```
num3 = input.nextDouble( );
```

```
nome = input.nextLine( );
```

Entrada de dados – Classe Scanner

```
1  /* Adicao.java
2  *Programa para adicao de dois numeros*/
3  import java.util.Scanner;
4
5  public class Adicao
6  {
7      public static void main(String args[])
8      {
9          Scanner entrada; // declara obj. Scanner
10         int num1, num2, soma; // declara variaveis inteiras
11
12         // instancia objeto entrada
13         entrada = new Scanner(System.in);
14
15         System.out.print("Digite o primeiro numero: ");
16         num1 = entrada.nextInt();
17
18         System.out.print("Digite o segundo numero: ");
19         num2 = entrada.nextInt();
20
21         soma = num1 + num2;
22         System.out.println(" A soma é| "+ soma);
23
24     } // fim do método principal
25 } // fim da classe
```

variável tipo Scanner

variáveis de tipo primitivo

Tipos primitivos do Java

TIPO	TAMANHO EM BITS	VALORES
char	16	'\u0000' a '\uFFFF' (0 A 65535)
byte	8	-128 a +127
short	16	-32768 a +32767
int	32	-2.147.483.648 a +2.147.483.647
long	64	-9.223.372.036.854.775.808 a +9.223.372.036.854.775.807
float	32	-3.4028234663852886E+38 a -1.40129846432481707E-45
double	64	-1.7976931348623157e+308 a -4.94065645841246544E-324
	boolean - 8 bits	true / false

Operadores e precedência

OPERAÇÃO JAVA	OPERADOR
adição	+
subtração	-
multiplicação	*
divisão	/
resto	%

OPERADORES
*, /, %
+, -

Entrada de cadeia de Caracteres – String

```
1  /* Programa para entrada de dados usando a classe Scanner */
2
3  import java.util.Scanner;
4
5  public class EntradaDados
6  {
7      public static void main(String args[])
8      {
9          Scanner entrada; // declara obj. tipo Scanner
10         String msg; // declara obj. tipo String
11
12         System.out.print("Digite uma mensagem: ");
13
14         // instancia objeto entrada
15         entrada = new Scanner(System.in);
16
17         // chama método para ler String
18         msg = entrada.nextLine();
19
20         // imprime String armazenada em msg
21         System.out.println(msg);
22
23     } // fim do método principal
24 } // fim da classe
```

variável do tipo
Scanner = objeto

Não é um tipo primitivo

Entrada de
valores reais
- double

```
1  /* AdicaoReais.java
2  L  *Programa para adicao de dois numeros reais*/
3  import java.util.Scanner;
4
5  public class AdicaoReais
6  {
7      public static void main(String args[])
8      {
9          Scanner entrada;           // declara obj. Scanner
10         double num1, num2, soma;    // declara variaveis reais
11
12         // instancia objeto entrada
13         entrada = new Scanner(System.in);
14
15         System.out.print("Digite o primeiro numero: ");
16         num1 = entrada.nextDouble();
17
18         System.out.print("Digite o segundo numero: ");
19         num2 = entrada.nextDouble();
20
21         soma = num1 + num2;
22         System.out.println(" A soma é "+ soma);
23
24     } // fim do método principal
25 } // fim da classe
```


Formatação de reais

```
▶▶ Digite o primeiro número: 4,5  
▶▶ Digite o segundo número: 5,5  
A soma é 10,000000
```

```
System.out.printf( "A soma é %5.2f\n", soma );
```

total de casas
(parte inteira + vírgula + casas decimais)

número de casas decimais

```
▶▶ Digite o primeiro número: 5,5  
▶▶ Digite o segundo número: 6  
A soma é 11,50
```

```

1 import java.util.Scanner;
2 public class TiposDistintos
3 {
4     public static void main( String args[] )
5     {
6         Scanner ent_Str = new Scanner( System.in );
7         Scanner ent_Int = new Scanner( System.in );
8         Scanner ent_Doub = new Scanner( System.in );
9
10        String nome;
11        int idade;
12        double salario;
13
14        System.out.print( "Digite o nome :" );
15        nome = ent_Str.nextLine();
16
17        System.out.print( "Digite a idade:" );
18        idade = ent_Int.nextInt();
19
20        System.out.print( "Digite o salário:" );
21        salario = ent_Doub.nextDouble();
22
23        System.out.printf( "Dados lidos:\n" );
24        System.out.printf( "Nome : %s\n", nome );
25        System.out.printf( "Idade : %d\n", idade );
26        System.out.printf( "Salário: R$ %7.2f\n", salario );
27    }
28

```

Entrada de
dados de
tipos
distintos

Operadores relacionais

Operador relacional	Resultado
$x == y$	Verdadeiro se x igual a y
$x != y$	Verdadeiro se x diferente de y
$x < y$	Verdadeiro se x menor que y
$x > y$	Verdadeiro se x maior que y
$x \leq y$	Verdadeiro se x menor ou igual a y
$x \geq y$	Verdadeiro se x maior ou igual a y

Operadores lógicos

- Usados em expressões lógicas

Operador	Resultado
!x	Verdadeiro se x for falso
x && y	Verdadeiro se x e y ambos verdadeiros
x y	Verdadeiro se x ou y (ou ambos) verdadeiros

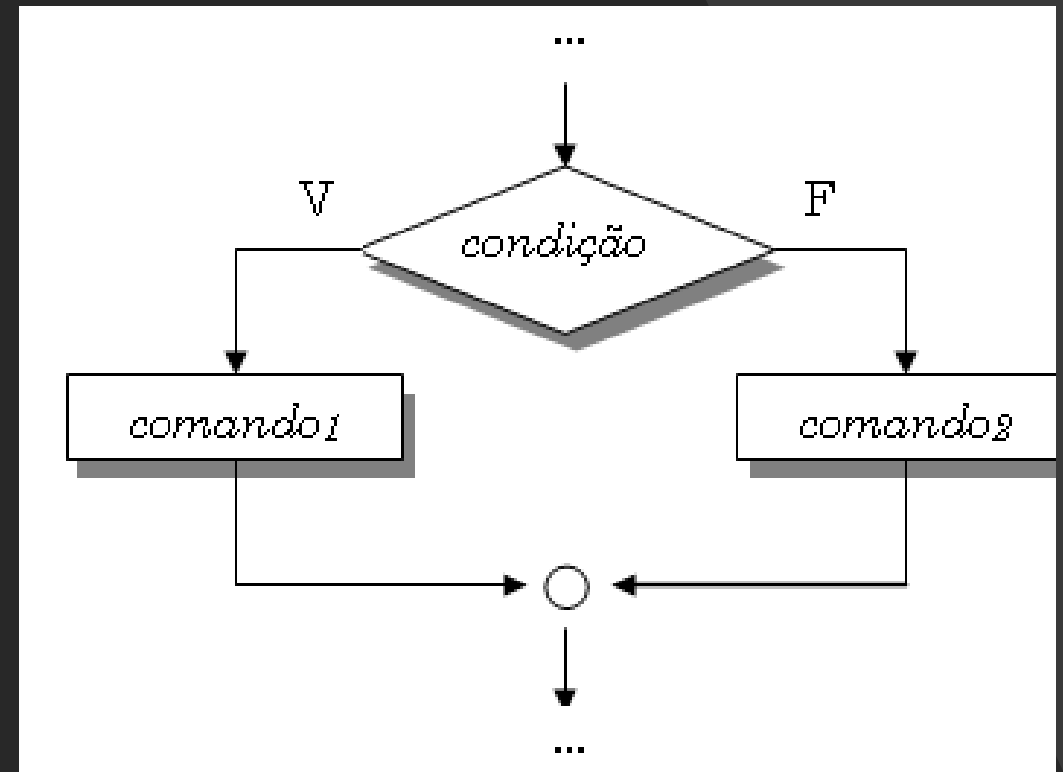
Estrutura de decisão simples

Em Java:

```
if (condição)
    comando1;
else
    comando2;
```

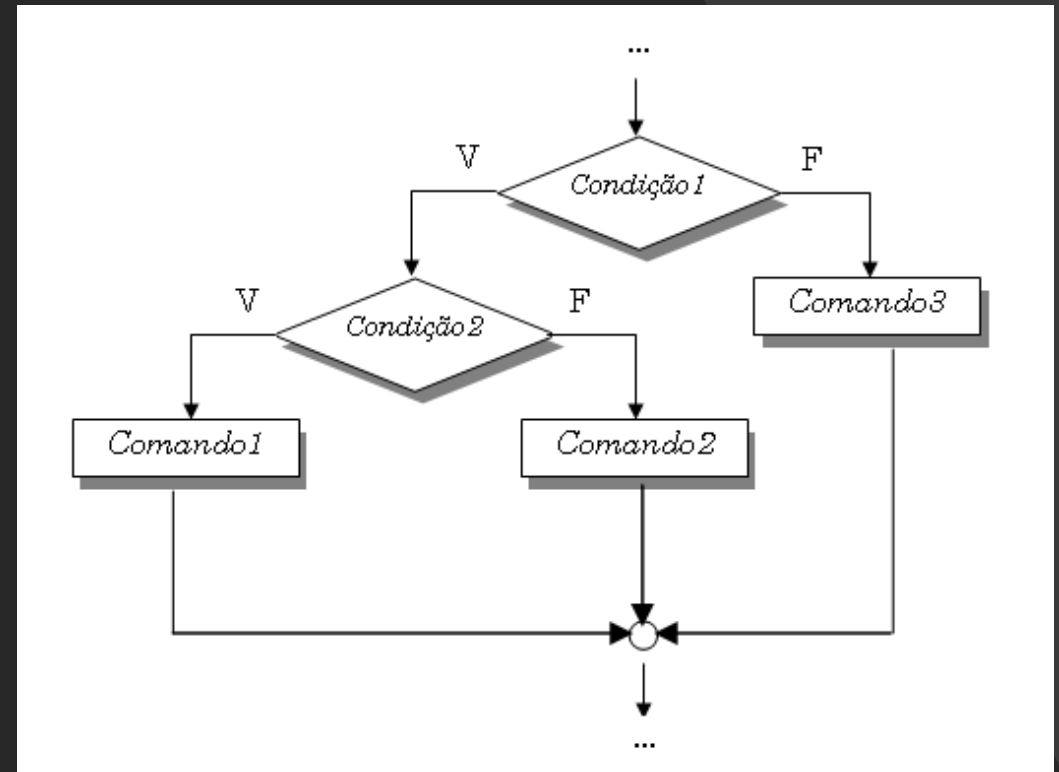
Exemplo:

```
if (m >= 0)
    System.out.println("Positivo");
else
    System.out.println("Negativo");
```



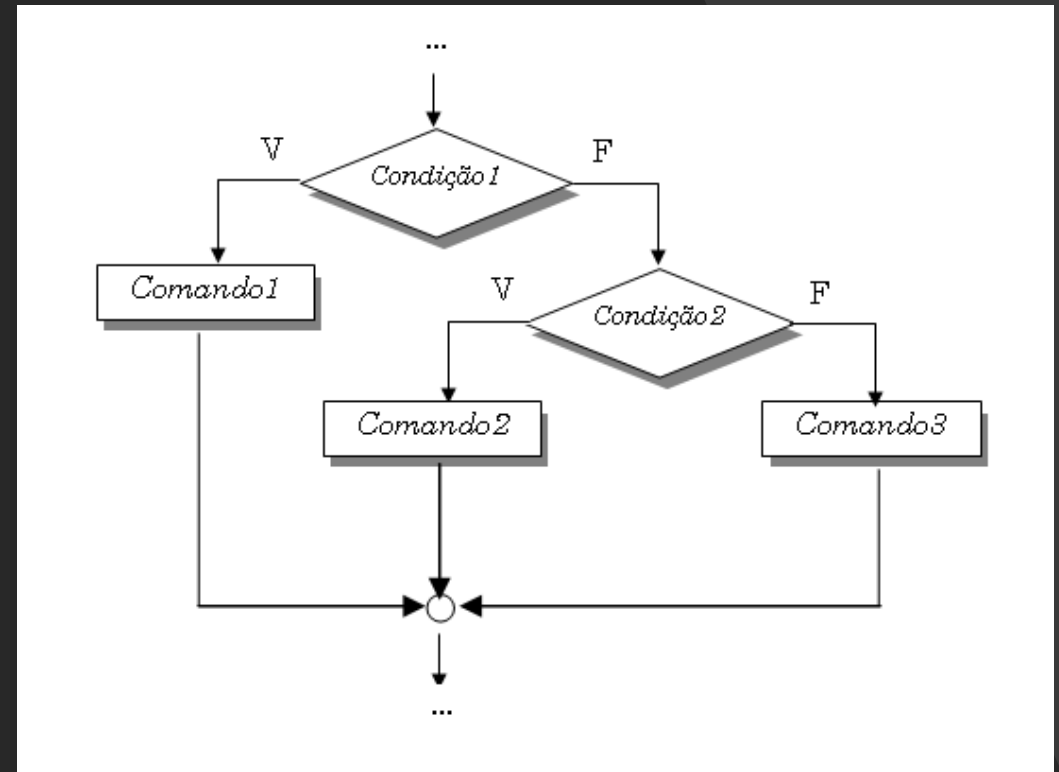
Condicional aninhada

```
if( condição1 )  
    if (condição2)  
        Comando1;  
    else  
        Comando2;  
else  
    Comando3;
```



Condicional encadeada

```
if( condição1 )  
    Comando1;  
else  
    if (condição2)  
        Comando2;  
    else  
        Comando3;
```



Exemplo - Encadeado

```
if (m > 0)
    System.out.println("Positivo");
else
    if (m = 0)
        System.out.println("Zero");
    else
        System.out.println("Negativo");
```


Casting

- Alguns valores são incompatíveis se você tentar fazer uma atribuição direta.
- Nesses casos é preciso uma conversão entre tipos (casting).
- Para atribuir tipos com maior capacidade para uma variável com menor capacidade esta conversão deve ser explícita.
- No caso contrário, o casting é implícito.

Casting

- No exemplo abaixo, um valor real é armazenado em uma variável inteira usando casting.
- Exemplo:

```
double d = 3.17;  
int i = d; //  
    erro  
int i = (int) d; //  
    casting  
System.out.println (" i: " + i );  
d = i; //  
    casting implícito  
System.out.println (" d: " + d );
```

Mais exemplos de casting

- Tipos float não podem armazenar double.
- É necessário fazer casting.

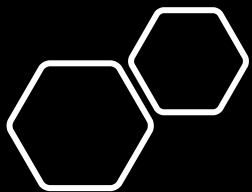
```
float x = 0.0;           // erro  
float x = 0.0f;          // conversão para  
float
```
- Outro exemplo:

```
double d = 5;  
float f = 3;  
float x = f + (float) d;
```

Condicional Simples

Dado um número real n ,
imprimir seu valor absoluto
(módulo).

```
1  import java.util.Scanner;
2
3  public class Absoluto
4  {
5      public static void main (String args[])
6      {
7          Scanner entrada = new Scanner(System.in);
8
9          System.out.println("Digite um numero:");
10         double n = entrada.nextDouble();
11         if (n < 0)
12             n = -n;
13         System.out.println("O modulo de n eh " + n);
14     }
15 }
```



Operador condicional ternário

- Operador para representar decisões simples de maneira mais compacta.
- Sintaxe:
condição ? expressão1 : expressão2
- Avalia a *condição*;
 - se for verdadeira, o resultado é o valor da *expressão1*;
 - senão, o resultado é o valor da *expressão2*.

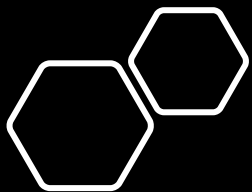
```
import java.util.Scanner;

public class Absoluto
{
    public static void main (String args[])
    {
        Scanner entrada = new Scanner(System.in);

        System.out.println("Digite um numero:");
        double n = entrada.nextDouble();
        System.out.println("O modulo de n eh " + (n<0 ? - n: n));
    }
}
```

Exemplo

Dado um número real n , imprimir seu valor absoluto (módulo) usando operador condicional ternário.



Estrutura de decisão múltipla

- Usada quando precisamos escolher uma entre várias alternativas previamente definidas;

```
switch(exp)  
{  
    case const1: comando1;  
        break;  
    case const2: comando2;  
        break;  
    ...  
    case constn: comandon;  
        break;  
    default: comando;  
}
```

Quais as
saídas nesse
exemplo?

```
import java.util.Scanner;

public class DecisaoMulti
{
    public static void main (String args[])
    {
        Scanner entrada = new Scanner(System.in);

        System.out.println("Digite um numero:");
        int n = entrada.nextInt();
        switch( n )
        {
            case 1: System.out.print('A');
                    break;
            case 3: System.out.print('B');
            case 4: System.out.print('C');
                    break;
            default: System.out.print('*');
            case 5: System.out.print('D');
        }
        System.out.print(' ');
    }
}
```


Outro exemplo

- Uso da estrutura de decisão múltipla para implementar uma simples calculadora;
- O usuário digita uma expressão da forma **val1** <enter> **operador** <enter> **val2** <enter> e o programa fornece-lhe seu valor como resposta.
- Os valores podem ser reais e os operadores aceitos são: +, -, * e /.

Solução (implementar)

```
import java.util.Scanner;
public class Calc
{
    public static void main (String args[])
    {
        Scanner eStr = new Scanner(System.in);
        Scanner eNum = new Scanner(System.in);

        System.out.println("Digite uma expressao:");
        double x = eNum.nextDouble();
        String s = eStr.nextLine();
        double y = eNum.nextDouble();

        char op = s.charAt(0);
        switch( op )
        {
            case '+': System.out.print("valor = "+ (x+y));
                       break;
            case '-': System.out.print("valor = "+ (x-y));
                       break;
            case '*': System.out.print("valor = "+ (x*y));
                       break;
            case '/': System.out.print("valor = "+ (x/y));
                       break;
            default : System.out.print("Operador inválido:" + op);
        }
    }
}
```