

PROGRAMAÇÃO ORIENTADA A OBJETOS

Introdução a Aplicativos Java (parte 2)

Prof. Emanuel Barreiros



Um programa um pouquinho mais complexo (parte 1)

```
1  import java.util.Scanner;
2
3  ▼ public class Soma {
4
5  ▼      public static void main(String[] args) {
6
7          //criar um novo objeto para obter entrada do usuário
8          Scanner scan = new Scanner(System.in);
9
10         //variáveis que serão usadas no código
11         int numero1;
12         int numero2;
13         int soma;
14
15         //solicita o primeiro número
16         System.out.print("Informe o número 1: ");
17         //obtem o primeiro número
18         numero1 = scan.nextInt();
```

Um programa um pouquinho mais complexo (parte 2)

```
20      //solicita o segundo número
21      System.out.print("Informe o número 2: ");
22      //obtem o segundo número
23      numero2 = scan.nextInt();
24
25      //soma os dois números obtidos
26      soma = numero1 + numero2;
27
28      //exibe a soma
29      System.out.println("Soma: " + soma);
30      |
31      scan.close();
32  }
33
34 }
```

Vamos analisar a parte 1

```
1  import java.util.Scanner;
2
3  ▼ public class Soma {
4
5  ▼      public static void main(String[] args) {
6
7          //criar um novo objeto para obter entrada do usuário
8          Scanner scan = new Scanner(System.in);
9
10         //variáveis que serão usadas no código
11         int numero1;
12         int numero2;
13         int soma;
14
15         //solicita o primeiro número
16         System.out.print("Informe o número 1: ");
17         //obtem o primeiro número
18         numero1 = scan.nextInt();
```

Declarações de import devem aparecer antes da declaração da classe

Criamos um novo objeto do tipo Scanner para obter dados do usuário.

Vamos analisar a parte 1

```
1  import java.util.Scanner;
2
3  ▼ public class Soma {
4
5  ▼      public static void main(String[] args) {
6
7          //criar um novo objeto para
8          Scanner scan = new Scanner
9
10         //variáveis que serão usadas no código
11         int numero1;
12         int numero2;
13         int soma;
14
15         //solicita o primeiro número
16         System.out.print("Informe o número 1: ");
17         //obtem o primeiro número
18         numero1 = scan.nextInt();
```

Declaração de variáveis

Solicitação do primeiro número

Obtém um inteiro a partir do teclado

Vamos analisar a parte 2

Exibe o valor armazenado na variável soma

Soma os dois números obtidos e atribui o resultado a outra variável

```
20 //solicita o segundo número
   System.out.print("Informe o n
   //obtem o segundo número
   numero2 = scan.nextInt();

24
25 //soma os dois números obtidos
26 soma = numero1 + numero2;
27
28 //exibe a soma
29 System.out.println("Soma: " + soma);
30 |
31 scan.close();
32 }
33
34 }
```

Liberar recursos não mais utilizados

Avaliação de expressões

- Em geral, expressões são escritas em uma única linha
- A precedência de operadores é a mesma da matemática:

Operador	Operação	Precedência
*	Multiplicação	Avaliados primeiro. Se houverem vários desses operadores, suas expressões são avaliadas da esquerda para a direita.
/	Divisão	
%	Resto	
+	Soma	Avaliados após os operadores acima. Se houverem vários desses operadores, suas expressões são avaliadas da esquerda para a direita.
-	Subtração	
=	Atribuição	Avaliado por último

Avaliação de expressões

- Parênteses mudam as regras do jogo
- Quando existirem parênteses, as expressões serão avaliadas do parêntese mais interno para o mais externo
- Ex: $(1 + 2) * 3 = 9$
- $1 + 2 * 3 = 7$

Operadores relacionais e de igualdade

- Condição: uma expressão cujo valor pode ser **true** ou **false**
- Condições em construções if podem ser criadas usando-se operadores condicionais ou de igualdade

Operadores relacionais e de igualdade

Operador Algébrico	Operador em Java	Exemplo	Significado
Operadores de Igualdade			
=	==	$x == y$	x é igual a y
≠	!=	$x != y$	x é diferente de y
Operadores relacionais			
>	>	$x > y$	x é maior que y
<	<	$x < y$	x é menor que y
≥	>=	$x >= y$	x é maior ou igual a y
≤	<=	$x <= y$	x é menor ou igual a y

O *if*

- Construção que permite a execução de um determinado trecho de código dada uma condição
- Só permite a avaliação de expressões que tenham como resultado um valor booleano
- Ex:

```
if ( <condicao> ) {  
    // código a ser executado  
}
```

Exercício

- 1) Escreva um programa em Java que solicite dois números inteiros ao usuário, e responda se esses dois números são iguais, se o primeiro é maior que o segundo, ou se o primeiro é menor que o segundo.
- 2) Escreva um programa em Java que solicite uma operação o usuário (soma, subtração, multiplicação ou divisão), depois solicite dois números, realize a operação solicitada pelo usuário e exiba o valor resultante na tela.