



Funções Aritméticas Operadores Relacionais e Lógicos

Prof. Luís Guilherme de S. Lopes



Agenda

Funções Aritméticas

Operadores Relacionais



Funções Aritméticas em Java

Funções e suas equivalências em Java:

- Valor Absoluto: `Math.abs(x)`
- Exponenciação: `Math.pow(base, expoente)`
- Valor Inteiro (parte inteira): Conversão com casting (ex.: `(int) x`)
- Raiz Quadrada: `Math.sqrt(x)`
- Pi: `Math.PI`
- Seno: `Math.sin(ânguloEmRadianos)`
- Cosseno: `Math.cos(ânguloEmRadianos)`
- Tangente: `Math.tan(ânguloEmRadianos)`
- Graus para Radianos: `Math.toRadians(ângulo)`



Exemplo “absoluto” em Java

```
public class Absoluto {  
    public static void main(String[] args) {  
        int a = Math.abs(-50);  
        System.out.println(a);  
    }  
}
```



Exemplo "exponencial" em Java

```
public class Exponencial {  
    public static void main(String[] args) {  
        double a = Math.pow(9, 5);  
        System.out.println(a);  
    }  
}
```



Exemplo

“raizquadrada” em Java

```
public class RaizQuadrada {  
    public static void main(String[] args) {  
        double a = Math.sqrt(81);  
        System.out.println(a);  
    }  
}
```



Exemplo “grauspararadianos” em Java

```
public class GrausParaRadianos {  
    public static void main(String[] args) {  
        double a = Math.toRadians(90);  
        System.out.println(a);  
    }  
}
```



Exemplo Combinando Funções Aritméticas (Conversor de Seno)

```
import java.util.Scanner;

public class Conversor {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Informe um ângulo: ");
        double angulo = scanner.nextDouble();
        double s = Math.sin(Math.toRadians(angulo));
        System.out.println("O seno de " + angulo + " é igual a " + s);
        scanner.close();
    }
}
```




PRATIQUE

Crie um programa que solicite ao usuário uma informação, utilize parênteses, operadores, funções aritméticas, leitura de dados e variáveis com tipos primitivos.



Operadores Relacionais

Operadores:

- Maior que: >
- Menor que: <
- Maior ou igual a: >=
- Menor ou igual a: <=
- Igual a: ==
- Diferente de: !=

Observação: Em Java, o operador de atribuição é = e o de igualdade é ==.



Operadores Relacionais

Comparando valores numéricos

Exemplo 1:

```
public class ExemploRelacional1 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(A > B);  
    }  
}
```

Exemplo 2:

```
public class ExemploRelacional2 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(A == B);  
    }  
}
```

Exemplo 3:

```
public class ExemploRelacional3 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(A != B);  
    }  
}
```



Operadores Relacionais

Exemplo 4:

```
public class ExemploRelacional4 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(C > 5);  
    }  
}
```

Exemplo 5:

```
public class ExemploRelacional5 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(C >= 5);  
    }  
}
```

Exemplo 6:

```
public class ExemploRelacional6 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(C == A + B);  
    }  
}
```



Operadores Relacionais

Exemplo 7:

```
public class ExemploRelacional7 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(C >= Math.pow(B, A));  
    }  
}
```

Observação: Em Java, para exponenciação utiliza-se Math.pow(), pois o operador ^ representa XOR (bitwise).

Exemplo 8:

```
public class ExemploRelacional8 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(C <= Math.pow(B, A));  
    }  
}
```

Exemplo 9:

```
public class ExemploRelacional9 {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(1 == C % 2);  
    }  
}
```

(% operador para obter o resto da divisão)



Operadores Lógicos

Os operadores lógicos também retornam resultados lógicos como verdadeiro ou falso, mas não são utilizados para comparar expressões ou números, servem para comparar outros resultados lógicos.

E (AND): &&

OU (OR): ||

NÃO (NOT): !



Operadores Lógicos

Operador E

```
public class ExemploLogicoOu {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println((A == B) && (C > A));  
    }  
}
```



Operadores Lógicos

Operador OU

```
public class ExemploLogicoOu {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println((A == B) || (C > A));  
    }  
}
```




Operadores Lógicos

Operador Não e OU

```
public class ExemploLogicoNaoOu {  
    public static void main(String[] args) {  
        int A = 2, B = 3, C = 5;  
        System.out.println(!((A == B) || (C > A)));  
    }  
}
```



Ordem de Precedência dos Operadores em Java

- Aritméticos: `()`, `*`, `/`, `+`, `-`
- Relacionais: `>`, `<`, `>=`, `<=`, `==`, `!=`
- Lógicos: `&&`, `||`, `!`



Operadores Lógicos

Crie um algoritmo solicitando que o usuário digite os 3 lados de um triângulo. Verifique se o triângulo é Equilátero ou Escaleno, e escreva:

O triângulo é EQUILÁTERO? (VERDADEIRO / FALSO)

O triângulo é ESCALENO? (VERDADEIRO / FALSO)



Exemplo: Triângulos em Java

```
import java.util.Scanner;

public class Triangulos {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Digite o primeiro lado: ");
        double L1 = scanner.nextDouble();
        System.out.print("Digite o segundo lado: ");
        double L2 = scanner.nextDouble();
        System.out.print("Digite o terceiro lado: ");
        double L3 = scanner.nextDouble();

        boolean EQ = (L1 == L2) && (L2 == L3);
        boolean ES = (L1 != L2) && (L2 != L3) && (L1 != L3);

        System.out.println("O triângulo é EQUILÁTERO? " + EQ);
        System.out.println("O triângulo é ESCALENO? " + ES);

        scanner.close();
    }
}
```

Exemplo: Triângulos com Teste de Formação

```
import java.util.Scanner;

public class TriangulosTeste {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Digite o primeiro lado: ");
        double L1 = scanner.nextDouble();
        System.out.print("Digite o segundo lado: ");
        double L2 = scanner.nextDouble();
        System.out.print("Digite o terceiro lado: ");
        double L3 = scanner.nextDouble();

        boolean TRI = (L1 < L2 + L3) && (L2 < L1 + L3) && (L3 < L1);
        boolean EQ = (L1 == L2) && (L2 == L3);
        boolean ES = (L1 != L2) && (L2 != L3) && (L1 != L3);

        System.out.println("Pode formar um TRIÂNGULO? " + TRI);
        System.out.println("O triângulo é EQUILÁTERO? " + EQ);
        System.out.println("O triângulo é ESCALENO? " + ES);

        scanner.close();
    }
}
```