

**Lista de Exercícios - Unidade 3: Decisões**

1. Qual o valor de cada variável dos trechos de código Java a seguir após o respectivo comando `if`?

a) `int n = 1; int k = 2; int r = n;`  
`if (k < n) { r = k; }`

b) `int n = 1; int k = 2; int r;`  
`if (n < k) { r = k; }`  
`else { r = k + n; }`

c) `int n = 1; int k = 2; int r = k;`  
`if (r < k) { n = r; }`  
`else { k = n; }`

d) `int n = 1; int k = 2; int r = 3;`  
`if (r < n + k) { r = 2 * n; }`  
`else { k = 2 * r; }`

Fonte: Horstmann (2013, p. 121, R3.1)

2. O que está errado em cada um dos fragmentos de código Java a seguir?

a) `if x > 0 then System.out.print(x);`

b) `if (1 + x > Math.pow(x, Math.sqrt(2))) { y = y + x; }`

c) `if (x = 1) { y++; }`

d) `x = in.nextInt();`  
`if (in.hasNextInt()) {`  
    `sum = sum + x;`  
`}`  
`else {`  
    `System.out.println("Bad input for x");`  
`}`

e) `String letterGrade = "F";`  
`if (grade >= 90) { letterGrade = "A"; }`  
`if (grade >= 80) { letterGrade = "B"; }`  
`if (grade >= 70) { letterGrade = "C"; }`  
`if (grade >= 60) { letterGrade = "D"; }`

Fonte: Horstmann (2013, p. 122, R3.3)

3. O que os fragmentos de código Java a seguir imprimem?

```
a) int n = 1;
    int m = -1;
    if (n < -m) { System.out.print(n); }
    else { System.out.print(m); }
```

```
b) int n = 1;
    int m = -1;
    if (-n >= m) { System.out.print(n); }
    else { System.out.print(m); }
```

```
c) double x = 0;
    double y = 1;
    if (Math.abs(x - y) < 1) { System.out.print(x); }
    else { System.out.print(y); }
```

```
d) double x = Math.sqrt(2);
    double y = 2;
    if (x * x == y) { System.out.print(x); }
    else { System.out.print(y); }
```

Fonte: Horstmann (2013, p. 122, R3.4)

4. Escreva um programa em Java que lê um valor inteiro e imprime se ele é negativo, zero ou positivo.

Fonte: Horstmann (2013, p. 126, P3.1)

5. Escreva um programa em Java que lê um número em ponto-flutuante e imprime: “zero”, se o número é zero. Caso contrário, imprime “positivo” ou “negativo”, conforme o caso. O programa também deve imprimir “pequeno”, se o valor absoluto do número for menor do que 1, ou “grande”, se o número exceder 1.000.000

Fonte: Horstmann (2013, p. 126, P3.2)

6. Escreva um programa em Java que lê as coordenadas de um ponto no plano cartesiano,  $x$  e  $y$ , e determine se este ponto corresponde à origem, se está sobre um dos eixos (eixo  $x$  ou eixo  $y$ ) ou em qual quadrante o ponto se encontra (primeiro, segundo, terceiro ou quarto quadrante).

Autor: Roland Teodorowitsch

7. Escreva um programa em Java que lê um inteiro e imprime quantos dígitos o número tem, verificando se o número é  $\geq 10$ ,  $\geq 100$ , e assim por diante. (Assuma que todos os inteiros sejam menores do que dez bilhões.) Se o número for negativo, primeiro multiplique ele por  $-1$ .

Fonte: Horstmann (2013, p. 126, P3.3)

8. Escreva um programa em Java que lê três números e imprime: “todos iguais”, se eles forem todos iguais; “todos diferentes”, se forem todos diferentes; e “nenhuma alternativa”, em caso contrário.

Fonte: Horstmann (2013, p. 126, P3.4)

9. Escreva um programa em Java que lê três números e imprime: “crescentes”, se eles estiverem em ordem crescente; “decréscuentes”, se eles estiverem em ordem decrescente; e “nenhuma ordem aparente”, em caso contrário. A ordem deve ser seguida estritamente, ou seja, cada valor deverá ser maior do que seu predecessor (ou menor, conforme o teste). A sequência 344, por exemplo, não deve ser considerada crescente.

Fonte: Horstmann (2013, p. 126, P3.5)

10. Escrever um programa em Java que lê 3 valores ( $a$ ,  $b$  e  $c$ ) e calcula e escreve a média ponderada com peso 5 para o maior dos 3 valores e peso 2,5 para os outros dois.

Adaptado de: Orth (2001, p. 39)

11. Escrever um programa em Java que lê dois valores ( $a$  e  $b$ ) e verifica se eles são múltiplos ou não. Escrever os números e uma mensagem (“são múltiplos” ou “não são múltiplos”).

Adaptado de: Orth (2001, p. 39)

12. Escrever um programa em Java que lê 4 números inteiros e imprime: “dois pares”, se na entrada houver dois pares de números iguais (em qualquer posição), ou “sem dois pares”, em caso contrário. Por exemplo, em “1 2 2 1” e “2 2 2 2” há “dois pares”. Para “1 2 2 3”, por outro lado, seria impresso “sem dois pares”.

Fonte: Horstmann (2013, p. 126, P3.8)

13. Escreva um programa em Java que leia 3 *strings* e ordene elas lexicograficamente. Por exemplo:

Forneça 3 strings: **Charlie Alfa Bravo**

Alfa

Bravo

Charlie

Fonte: Horstmann (2013, p. 127, P3.16)

14. Quando dois instantes de tempo são comparados, cada um dado em horas (variando de 0 a 23) e minutos, o seguinte pseudocódigo determina qual ocorre antes:

```
If hour1 < hour2
    time1 comes first.
Else if hour1 and hour2 are the same
    If minute1 < minute2
        time1 comes first.
    Else if minute1 and minute2 are the same
        time1 and time2 are the same.
    Else
        time2 comes first.
Else
    time2 comes first.
```

Escreva um programa em Java que solicite dois horários ao usuário (cada hora formada por dois valores inteiros: horas e minutos) e imprima o menor horário antes e o maior depois.

Fonte: Horstmann (2013, p. 127, P3.17)

15. O seguinte algoritmo descobre a estação do ano (Primavera, Verão, Outono ou Inverno) para determinado mês e dia.

```
If month is 1, 2, or 3, season = "Winter"
Else if month is 4, 5, or 6, season = "Spring"
Else if month is 7, 8, or 9, season = "Summer"
Else if month is 10, 11, or 12, season = "Fall"
If month is divisible by 3 and day >= 21
    If season is "Winter", season = "Spring"
    Else if season is "Spring", season = "Summer"
    Else if season is "Summer", season = "Fall"
    Else season = "Winter"
```

Escreva um programa em Java que solicite o dia e o mês ao usuário e então descubra e imprima a estação do ano. O algoritmo acima funciona para o hemisfério em que nos encontramos? Caso não funcione adapte-o.

Adaptado de: Horstmann (2013, p. 128, P3.18)

## REFERÊNCIAS

HORSTMANN, C. **Java for Everyone – Late Objectct**. 2. ed. Hoboken: Wiley, 2013. xxxiv, 589 p.

ORTH, Afonso Inácio. **Algoritmos e Programação com Resumo das Linguagens PASCAL e C**. Porto Alegre: AIO, 2001. 176 p.