Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Escola Politécnica
Disciplina: Fundamentos de Programação - Professor: Roland Teodorowitsch
4 de maio de 2023

Lista de Exercícios - Unidade 4: Laços

1. Escreva laços em Java usando `while` para imprimir

   a) Todos os quadrados menores do que `n`. Por exemplo, se `n` for 100, o laço deve imprimir 0 1 4 9 16 25 36 49 64 81.

   b) Todos os números positivos que são divisíveis por 10 e menores do que `n`. Por exemplo, se `n` for 100, o laço deve imprimir 10 20 30 40 50 60 70 80 90.

   c) Todas as potências de 2 menores do que `n`. Por exemplo, se `n` for igual a 100, o laço deve imprimir 1 2 4 8 16 32 64.

   Fonte: Horstmann (2013, p. 184)

2. Escreva laços em Java que calculem

   a) A soma de todos os números pares entre 2 e 100 (inclusive).

   b) A soma de todos os quadrados entre 1 e 100 (inclusive).

   c) A soma de todos os números ímpares entre `a` e `b` (inclusive).

   d) A soma de todos os dígitos ímpares de `n`. (Por exemplo, se `n` for igual a 32677, a soma será $3 + 7 + 7 = 17$.)

   Fonte: Horstmann (2013, p. 184)

3. Crie tabelas de depuração (teste de mesa) para os seguintes laços:

   a)
   ```
   int i = 0; int j = 10; int n = 0;
   while (i < j) { i++; j--; n++; }
   ```

   b)
   ```
   int i = 0; int j = 0; int n = 0;
   while (i < 10) { i++; n = n + i + j; j++; }
   ```

   c)
   ```
   int i = 10; int j = 0; int n = 0;
   while (i > 0) { i--; j++; n = n + i - j; }
   ```

   d)
   ```
   int i = 0; int j = 10; int n = 0;
   while (i != j) { i = i + 2; j = j - 2; n++; }
   ```

   Fonte: Horstmann (2013, p. 184)

4. O que os seguintes laços imprimem?

   a)
   ```
   for (int i = 1; i < 10; i++) { System.out.print(i + " "); }
   ```

   b)
   ```
   for (int i = 1; i < 10; i += 2) { System.out.print(i + " "); }
   ```

   c)
   ```
   for (int i = 10; i > 1; i--) { System.out.print(i + " "); }
   ```

d)
```java
for (int i = 0; i < 10; i++) { System.out.print(i + " "); }
```

e)
```java
for (int i = 1; i < 10; i = i * 2) { System.out.print(i + " "); }
```

f)
```java
for (int i = 1; i < 10; i++) { if (i % 2 == 0) { System.out.print(i + " "); } }
```

Fonte: Horstmann (2013, p. 184)

5. Escreva um programa em Java que imprime uma tabela de multiplicação, como a mostrada a seguir:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
...									
10	20	30	40	50	60	70	80	90	100

Fonte: Horstmann (2013, p. 191)

6. Escreva um programa em Java que inicialmente leia um número inteiro correspondente ao número de valores reais que deverão serem lidos a seguir, determinando: quantos destes valores são positivos, quantos destes valores são negativos, quantos são iguais a zero, qual o maior valor, qual o menor valor, qual a média, qual o somatório destes valores reais e qual o produtório destes valores reais.

Autor: Roland Teodorowitsch

7. A série de Fibonacci tem seus 2 primeiros termos definidos como 0 e 1. À partir deles, os demais termos são construídos pela seguinte regra:

$$t_n = t_{n-1} + t_{n-2}$$

Escreva um programa em Java que gera os 10 primeiros termos da série de Fibonacci.

Adaptado de: Orth (2001, p. 51)

8. Escreva um programa em Java que obtenha o mínimo múltiplo comum (MMC) entre dois números fornecidos.

Adaptado de: Forbellone e Eberspächer (2005, p. 64)

9. Escreva um programa em Java que obtenha o máximo divisor comum (MDC) entre dois números fornecidos.

Adaptado de: Forbellone e Eberspächer (2005, p. 64)

10. Escreva um programa em Java que calcule o valor de H, sendo que ele é determinado pela série

$$H = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + ... \frac{99}{50}$$

Adaptado de: Forbellone e Eberspächer (2005, p. 65)

11. Escrever um programa em Java que lê um valor inteiro e positivo (n), calculando o escrevendo o valor de E, que é dado pela fórmula:

$$E = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + ... + + \frac{1}{n!}$$

Adaptado de: Orth (2001, p. 53)

12. Um Clube de Futebol de uma cidade fez uma pesquisa entre seus sócios, coletando dados sobre o salário e o número de filhos dos sócios. O Clube deseja saber:

a) A média do salário dos sócios
b) A média do número de filhos
c) O maior salário
d) O percentual de pessoas com salário até R$400,00

Escreva um programa em Java para ler, para cada sócio, o seu salário e o seu número de filhos. O final de dados se dará quando um salário negativo for lido.

Adaptado de: Orth (2001, p. 53)

13. Em uma eleição presidencial existem quatro candidados. Os votos são informados por código. Os dados utilizados para a escrutinagem obedecem à seguinte condificação:

    - 1, 2, 3, 4 = voto para os respectivos candidatos;
    - 5 = voto nulo;
    - 6 = voto em branco.

    Escreva um programa em Java que calcule e escreva:

    - o total de votos para cada candidato e o seu percentual sobre o total;
    - o total de votos nulos e o seu percentual sobre o total;
    - o total de votos em branco e o seu percentual sobre o total.

    Como finalizador do conjunto de votos tem-se o valor 0.

    Adaptado de: Forbellone e Eberspächer (2005, p. 64)

14. Gustavo tem 1,40 metros e cresce 8 centímetros por ano, enquanto Juliano tem 1,10 metros e cresce 17 centímetros por ano. Construa um programa em Java que calcula e escreve quantos anos serão necessários para que Juliano ultrapasse Gustavo.

    Adaptado de: Orth (2001, p. 53)

15. Escrever um programa em Java que gera e escreve os 5 primeiros números perfeitos. Um número perfeito é aquele que é igual à soma de seus divisores, exceto ele próprio. Por exemplo:

$$6 = 1 + 2 + 3$$

$$28 = 1 + 2 + 4 + 7 + 14$$

    Adaptado de: Orth (2001, p. 54)

16. Escreva um programa em Java que lê `n`, inteiro e positivo, e calcula e escreve o termo de ordem `n` da sucessão a seguir:

Ordem:	1	2	3	4	5	6	7	8	9	10
Sucessão:	-1	0	5	6	11	12	17	18	23	24

    Adaptado de: Orth (2001, p. 57)

17. *The game of Nim.* This is a well-known game with a number of variants. The following variant has an interesting winning strategy. Two players alternately take marbles from a pile. In each move, a player chooses how many marbles to take. The player must take at least one but at most half of the marbles. Then the other player takes a turn. The player who takes the last marble loses.
    Write a program in which the computer plays against a human opponent. Generate a random integer between 10 and 100 to denote the initial size of the pile. Generate a random integer between 0 and 1 to decide whether the computer or the human takes the first turn. Generate a random integer between 0 and 1 to decide whether the computer plays *smart* or *stupid*. In stupid mode the computer simply takes a random legal value (between 1 and $n/2$) from the pile whenever it has a turn. In smart mode the computer takes off enough marbles to make the size of the pile a power of two minus 1 — that is, 3, 7, 15, 31, or 63. That is always a legal move, except when the size of the pile is currently one less than a power of two. In that case, the computer makes a random legal move.
    You will note that the computer cannot be beaten in smart mode when it has the first move, unless the pile size happens to be 15, 31, or 63. Of course, a human player who has the first turn and knows the winning strategy can win against the computer.

    Fonte: Horstmann (2013, p. 192)

REFERÊNCIAS

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de programação**: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Prentice Hall, 2005. 218 p.
HORSTMANN, C. **Java for Everyone – Late Objects**. 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.