
Capítulo #06 - Instruções de Repetição

Exercício #02 (p. 363)

Escreva instruções `for`, `do ... while` e `while` para calcular as seguintes somas e produtos:

1. $1 + 2 + 3 + \dots + 100$
2. $5 + 10 + 15 + \dots + 50$
3. $1 + 3 + 7 + 15 + 31 + \dots + (2^{20} - 1)$
4. $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{15}$
5. $1 \times 2 \times 3 \times \dots \times 20$
6. $1 \times 2 \times 4 \times 8 \times \dots \times 2^{20}$

Exercício #04 (p. 364)

Determine o que escreve o código seguinte *sem o executar*.

```
System.out.format("%4d", 234);  
System.out.format("%5d", 234);  
  
System.out.format("%s", "\n");  
  
System.out.format("$%6.2f", 23.456);  
System.out.format("%s", "\n");  
  
System.out.format("%1%3d+%1$3d=%2$5d", 5, (5+5));
```

Problema #06 (p. 365)

Escreva um programa para escrever os números de 10 a 49 na forma seguinte:

```
10 11 12 13 14 15 16 17 18 19  
20 21 22 23 24 25 26 27 28 29  
30 31 32 33 34 35 36 37 38 39  
40 41 42 43 44 45 46 47 48 49
```

Como vai resolver este problema? Repare nesta resolução limitada:

```
for (int i = 10; i < 50; i++) {  
    switch (i) {  
        case 19:  
        case 29:
```

```
        case 39: System.out.println(" " + i);    // move to the
            break;                               // next line
        default: System.out.print(" " + i);
    }
}
```

Este código não é bom porque **só funciona para escrever os números de 10 a 49**. Tente desenvolver código que pode ser facilmente aplicado a qualquer intervalo de valores. Pode fazer isso de duas maneiras: Com um ciclo `for` aninhado ou com aritmética modular (Se dividir um número por 10 e o resto for 9 então o número é 9, 19, 29, 39, etc).

Problema #07 (p. 365)

O inteiro n é um **número primo** se $n > 1$ e n for divisível **apenas** por n e por 1. Os primeiros sete números primos são 2, 3, 5, 7, 11, 13 e 17. Escreva um método que devolve `true` se (e só se) o seu parâmetro for um número primo. Usando este método, escreva um programa que repetidamente pergunta um número ao utilizador e escreve `Primo` ou `Composto` conforme o valor dado é, ou não, um número primo. O programa termina quando o valor dado for negativo.

Sugestão: Utilize uma estrutura semelhante ao `Hi-Lo`.

Problema #13 (p. 366)

Existem 25 números primos entre 2 e 100, e existem 1229 primos entre 2 e 10000. Escreva um programa que pergunta um inteiro positivo $N > 2$ e escreve o número de primos entre 2 e N inclusive. Use a técnica de *timing* explicada na secção 6.9 para mostrar quanto tempo demorou a calcular o resultado.

Opcional (Problemas #09, #11, ++)

Termine o código do jogo `Hi-Lo` (p. 351 — 361). Veja o código parcial nas pastas `Step1`, `Step2`, `Step3` e `Step4` do arquivo `Sample_Programs_Ch06` (este arquivo também está indicado no moodle).