

ACH2001 – Introdução à Ciência da Computação

PRIMEIRO SEMESTRE DE 2008

LISTA DE EXERCÍCIOS I

Os exercícios aqui apresentados são complementares aos exercícios presentes nos capítulos 3 a 16 do livro-texto da disciplina. Desta forma, recomenda-se que os exercícios presentes no livro também sejam resolvidos.

1. Escreva um método **int maior(int a, int b, int c)** que, dados três valores inteiros, apresente o maior de tais valor.
2. Escreva um método **void ordena(int a, int b, int c)** que, dados três valores inteiros, apresente os três valores em ordem crescente.
3. Escreva um método **int inverte(int n)** que retorna na ordem inversa um inteiro (na base 10) com quatro algarismos.

Observação: seja m (o algarismo do milhar), c (o algarismo da centena), d (o algarismo da dezena) e u (o algarismo da unidade), o inverso de um número mcd u de quatro algarismos é $u * 1000 + d * 100 + c * 10 + m$.

4. Escreva um método **double soma(int n1, int d1, int n2, int d2)** que soma duas frações ordinárias ($n1/d1$ e $n2/d2$), fornecendo o resultado em forma de fração ($n3/d3$), onde $n3 = n1 * d2 + n2 * d1$ e $d3 = d1 * d2$.
5. Escreva um método **boolean éPar(int n)** que verifica se um inteiro é par ou ímpar.
6. Escreva um método **void raizes (double a, double b, double c)** que determina as raízes, reais ou complexas, de uma equação do segundo grau ($ax^2 + bx + c$), dados os seus coeficientes (a, b, c).

Observação: Seja $\Delta = b^2 - 4 * a * c$, as afirmações abaixo são verdadeiras:

- se $\Delta > 0$, temos $x' = \frac{-b - \sqrt{\Delta}}{2 * a}$ e $x'' = \frac{-b + \sqrt{\Delta}}{2 * a}$
- se $\Delta = 0$, temos $x' = x'' = \frac{-b}{2 * a}$
- se $\Delta < 0$, temos $x' = \frac{-b}{2 * a} - \frac{\sqrt{-\Delta}i}{2 * a}$ e $x'' = \frac{-b}{2 * a} + \frac{\sqrt{-\Delta}i}{2 * a}$

Observação: O método **double Math.sqrt(double a)** retorna a raiz quadrada do número a.

7. Escreva um método **classificaTriângulo(double a, double b, double c)** que verifica se três números dados podem ser comprimentos dos lados de um triângulo, classificando o triângulo nos casos afirmativos (equilátero, isósceles e escaleno). Observação: 3 lados (a,b,c) formam um triângulo, se a soma de dois lados é maior que o terceiro lado. Isto é, as três condições seguintes são verdadeiras:

- $a + b > c$
- $a + c > b$
- $b + c > a$

8. No sistema imperial frequentemente são utilizadas as seguintes medidas lineares:

- 1 pé = 12 polegadas, 1 jarda = 3 pés e 1 milha = 1760 jardas

Baseado no fato de que 1 polegada equivale a 25.3995 milímetros no sistema métrico, desenvolva uma classe com métodos que convertam comprimentos expressos nos sistema imperial para o sistema métrico e vice-versa.

9. Um ciclone tropical consiste em uma tempestade muito violenta que, sob determinadas condições, adquire um movimento de rotação em torno de uma região de baixa pressão atmosférica. Um ciclone pode atingir um diâmetro de até 500 km. Furacões são os mais violentos desses ciclones e chegam a produzir ventos de até 320 km/h. Na tabela abaixo, são apresentados alguns dados relativos à classificação dos furacões segundo a escala Saffir-Simpson:

Classificação	velocidade dos ventos (km/h)
tempestade tropical	62 a 118
furacão de categoria 1	119 a 153
furacão de categoria 2	154 a 177
furacão de categoria 3	178 a 209
furacão de categoria 4	210 a 249
furacão de categoria 5	maior que 249

Escreva um método que dado a velocidade do vento aferida durante uma tempestade e apresente ao usuário a classificação do furacão caso a tempestade possa ser classificada como tal.

10. Escreva um método **void imprimeDivisoresInteiros(int n)** que dado um número positivo $n > 0$, achar todos os seus divisores inteiros diferentes de n .

Exemplo: Se a entrada do método fosse 308, então a saída é 1 2 4 7 11 14 22 28 44 77 154

11. O número 3025 possui a seguinte característica:

$$\begin{cases} 30 + 25 = 55 \\ 55^2 = 3025 \end{cases}$$

Escreva um método que imprima todos os números de 4 algarismos que apresentam tal característica.

12. Escreva um método **int somaNPares(int n)** que calcule a soma dos n primeiros números inteiros pares positivos para um $n > 0$ fornecido pelo usuário.

13. Diz-se que um número inteiro n é um quadrado perfeito se existirem m números ímpares consecutivos a partir do valor 1 cuja soma é igual a n . Exemplo: $16 = 1 + 3 + 5 + 7$ (16 é igual à soma dos quatro primeiros ímpares a partir de 1) e $16 = 4^2$. Logo 16 representa um quadrado perfeito. Faça um método **boolean éQuadradoPerfeito(int n)** que verifica se um valor inteiro positivo fornecido pelo usuário é um quadrado perfeito ou não.

14. Escreva um método **boolean éPotênciaDeDois(int n)** que, para um dado valor n inteiro positivo, verifica se tal valor é uma potência de 2. Em caso afirmativo, o método deve fornecer o valor da potência. Em caso contrário, uma mensagem apropriada deve ser gerada pelo método para o usuário.

15. Escreva um método **boolean éPotência(int n, int p)** que, para um dado valor n inteiro positivo, verifica se tal valor é uma potência de p . Em caso afirmativo, o método deve fornecer o valor da potência. Em caso contrário, uma mensagem apropriada deve ser gerada pelo método para o usuário.
16. Em criptografia, a Cifra de César, também conhecida como cifra de troca ou ainda código de César, é uma das mais simples e conhecidas técnicas de encriptação. É um tipo de cifra de substituição em que cada letra do texto é substituída por outra, que se apresenta no alfabeto rotacionado à direita ou esquerda um número fixo de posições k . Por exemplo, com uma troca de 3 posições, A seria substituído por D, B viraria E e assim por diante. Implemente o método **String encripte(int k, String in)** que dado uma String, retorna uma outra String com os caracteres presentes na String de entrada encriptados pela Cifra de César.

Exemplos:

encripte(3, "CIFRA DE CESAR") retorna "FLIUD GH FHVDU"

encripte(2, "ABCDEFGHJKLMNOPQRSTUVWXYZ") retorna "CDEFGHIJKLMNOPQRSTUVWXYZAB"

17. Faça um método **void quadrado(int n)** que gere para um valor $n \geq 1$ fornecido pelo usuário, um "quadrado" de n linhas e n colunas que tenha caracteres '*' nas bordas. Por exemplo, para $n = 5$ o método deve gerar

```
*****
*   *
*   *
*   *
*   *
*****
```

18. Faça um método **void quadradoDiagonal(int n)** que gere para um valor $n \geq 1$ fornecido pelo usuário, um "quadrado" de n linhas e n colunas que tenha caracteres ':' nas posições das diagonais e os caracteres '*' nas demais. Por exemplo, para $n = 5$ o método deve gerar

```
:***:
*:*:~*
**:*~*
*:*:~*
:***:
```

19. Faça um método **void linhaCrescenteAsteriscoEsquerda(int n)** que gere para um valor $n \geq 1$ fornecido pelo usuário, n linhas que tenha 1 caractere '*' (da esquerda para direita) na 1ª linha, 2 caracteres '*' na 2ª linha e n caractere '*' na n ésima linha. Por exemplo, para $n = 5$ o método deve gerar

```
*
**
***
****
*****
```

20. Faça um método **void linhaCrescenteAsteriscoDireita(int n)** que gere para um valor $n \geq 1$ fornecido pelo usuário, n linhas que tenha n caracteres '*' (da direita para esquerda) na 1ª linha, $n - 1$ caracteres '*' na 2ª linha e 1 caractere '*' na enésima linha. Por exemplo, para $n = 5$ o método deve gerar

```

      *
     **
    ***
   ****
  *****

```

21. Faça um método **void linhaDecrescenteAsteriscoEsquerda(int n)** que gere para um valor $n \geq 1$ fornecido pelo usuário, n linhas que tenha n caracteres '*' (da esquerda para direita) na 1ª linha, $n - 1$ caracteres '*' na 2ª linha e 1 caractere '*' na enésima linha. Por exemplo, para $n = 5$ o método deve gerar

```

*****
****
***
**
*

```

22. Faça um método **void linhaDecrescenteAsteriscoDireita(int n)** que gere para um valor $n \geq 1$ fornecido pelo usuário, n linhas que tenha n caracteres '*' (da direita para esquerda) na 1ª linha, $n - 1$ caracteres '*' na 2ª linha e 1 caractere '*' na enésima linha. Por exemplo, para $n = 5$ o método deve gerar

```

*****
****
***
**
*

```

23. Faça um método que gere a tabuada **void tabuada(int n)** de 1 até um valor $n \geq 0$ na forma de uma tabela tal que, na posição da linha i e coluna j da tabela, deve-se encontrar o valor $i * j$. Por exemplo, para $n = 6$ o método deve gerar

1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30
6	12	18	24	30	36