

Exercícios – Lista 4 (Repetição)

Disciplina: INE5603 (20072) - Programação Orientada a Objetos I **Turma:** 0138D

Professora: Carla Merkle Westphall

1. Escreva um main Java que lê números inteiros até que o número lido seja igual a zero. O programa deve mostrar a soma total dos números lidos.
2. Escreva um main Java que mostra na tela a tabuada de multiplicação de um número fornecido pelo usuário.
3. Faça um main Java que leia um valor, calcule seu fatorial e o exiba o seu valor.
4. Escreva um main Java que leia vários números inteiros e positivos e calcule o produto dos números pares. O fim da leitura será indicado pelo número -1.
5. Elabore um main Java que gere 100 números inteiros aleatórios (entre 0 e 9999) e escreva o menor e o maior valor gerado.
6. Quais os valores armazenados nas variáveis **a** e **b** após a execução do conjunto de instruções abaixo?

```
int a, b;  
a = 9;  
b = 0;  
while ( a > b ) {  
    a = a - 2;  
    b++; }  
}
```

7. Cite quais os valores armazenados nas variáveis **cont** e **s** após a execução do conjunto de instruções a seguir:

```
int cont = 0;  
int s = 0;  
do { cont = cont + 1;  
    if ( cont > 5 )  
        s = s + cont % 2;  
} while ( cont <= 10 );
```

8. O que o seguinte programa imprime?

```
1 public class Mystery  
2 {  
3     public static void main( String args[] )  
4     {  
5         int y;  
6         int x = 1;  
7         int total = 0;  
8  
9         while ( x <= 10 )  
10        {  
11            y = x * x;  
12            System.out.println( y );  
13            total += y;  
14            ++x;  
15        } // fim do while  
16  
17        System.out.printf( "O total é %d\n", total );  
18    } // fim de main  
19  
20 } // fim da classe Mystery
```

9. Elabore um programa que calcule o valor de $H(x)$, calculado da seguinte forma:
 $H(x) = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/x$. O valor de x deve ser lido pelo teclado.
10. Elabore um **método public static** `geraTermosDaSerieDeFibonacci(int n)` que gera e escreve os n primeiros termos da série de Fibonacci. Considere os dois primeiros termos da série de Fibonacci sendo 1(um). Os números seguintes sempre serão o resultado da soma dos dois números imediatamente anteriores. Assim, os 10 primeiros termos da série de Fibonacci seriam: 1,1,2,3,5,8,13,21,34,55. O método deve ser invocado no método `main`.
11. Uma empresa decidiu fazer um recrutamento para preencher algumas vagas. Os candidatos serão cadastrados por computador. Escreva um `main` Java para:
- Ler o número do candidato, a idade, o sexo e a experiência profissional (S/N).
 - Mostrar a idade média dos candidatos.
 - Mostrar o número total de candidatos e candidatas.
 - Mostrar os candidatos (homens e mulheres) maiores de idade que tenham experiência no serviço.
12. Considerando o código disponível em (<http://moodle.inf.ufsc.br/file.php/570/Topico3/repeticao1.zip>), que implementa as classes `Pessoa`, `Interface` e `TestePessoa`, modifique a classe `TestePessoa` – use o método `char temMais()` da classe `Interface` na repetição para pedir idades até que o usuário responda “N” a uma resposta “Tem mais (S/N)?”.
13. Modifique o exercício anterior de forma que também se calcule e escreva na tela:
- a quantidade de pessoas do sexo masculino;
 - o percentual de homens com mais de 70 anos;
 - a menor e a maior idade; e
 - a idade média das pessoas do sexo feminino.

Crie um novo método na classe `Interface` para mostrar esses resultados.

14. Considere o método `char temMais()` da classe `Interface`. Modifique este método de forma que caso o usuário digite um caractere que não seja S ou N, o método deve informar ao usuário que o caractere é inválido e deve solicitar que o usuário digite novamente.
15. Crie a classe `Expressao` que representa uma expressão matemática a ser calculada. Escreva um programa Java para ler o valor de N (atributo `n` da classe `Expressao`), e chamar os métodos `calcularS` e `calcularH` para calcular o resultado das expressões abaixo.

$$S = \frac{1}{N} + \frac{2}{N-1} + \frac{3}{N-2} + \dots + \frac{N-1}{2} + \frac{N}{1}$$

$$H = \frac{N}{1} + \frac{N-1}{2} + \frac{N-2}{3} + \dots + \frac{1}{N}$$

```
public class Expressao {
    private int n;

    public double calcularS() {} // completar
    public double calcularH() {} // completar
} // fim classe Expressao
```

16. Elabore e teste um **método “static” verificaPrimo** que receba uma variável inteira qualquer x (x deve ser lido no método `main`) e verifique se este valor corresponde a um número primo. O método deve retornar uma `String` com o valor “SIM” se o número for primo e com o valor “NÃO” caso contrário. Um número primo é aquele cujos únicos divisores inteiros são somente o número 1 e o próprio número.

17. Implemente o **algoritmo euclidiano** usando um **método static** para localizar o máximo divisor comum de dois inteiros positivos. Esse método transforma um par de inteiros positivos (m, n) em um par (d, 0), fazendo as seguintes operações repetidamente:

- Divide-se o maior inteiro pelo menor inteiro.
- Depois, o maior inteiro assume o valor do menor, e o menor assume o valor do resto da divisão do item a.
- Quando o resto é 0, o outro inteiro no par será o máximo divisor comum do par original (e de todos os pares intermediários).

Por exemplo, se m é 532 e n é 112, o algoritmo euclidiano reduz o par (532,112) a (28,0) por (532,112) -> (112,84) -> (84,28) -> (28,0).

Dessa forma, 28 é o máximo divisor comum de 532 e 112. Esse resultado pode ser verificado a partir do fato de que $532 = 28 \times 19$ e $112 = 28 \times 8$. A razão pela qual o algoritmo euclidiano funciona é que cada par na seqüência tem o mesmo conjunto de divisores, que são precisamente os fatores do máximo divisor comum. No exemplo apresentado, esse conjunto comum de divisores é { 1, 2, 4, 7, 14, 28 }.

18. Considere a **classe Ip** que representa um número inteiro positivo e possui um único atributo **int n**. Escreva a implementação de métodos de forma que um objeto tenha a capacidade de:

- Fornecer a soma dos números naturais ímpares menores que o valor representado.
- Informar se está representando ou não um quadrado perfeito. Um inteiro positivo é um quadrado perfeito se o mesmo for igual à soma de seus divisores, excluindo ele próprio.
Exemplo: $28 = 1+2+4+7+14$.
- Fornecer o valor do maior dígito que compõe o valor por ele representado.
- Sem usar a operação de multiplicação (*), fornecer o produto do valor por ele representado com o valor representado por outro objeto.
- Fornecer o somatório S_1 , dado por: $S_1 = 1 + 1/2 + 1/3 + \dots + 1/n$.
- Fornecer o fatorial do número que representa.
- Fornecer o valor do somatório S_3 , dado por: $S_3 = 1 + 1/2! + 1/3! + \dots + 1/n!$.
- Usando operações de multiplicação, fornecer o resultado do valor representado pelo número elevado à potência m, onde m é um inteiro positivo.
- Considere n o valor que o objeto representa, para um valor **double x**, fornecer o valor S dado por: $S = x^0 + x^1 + x^2 + \dots + x^n$.

19. Considerando ainda a classe **Ip**, escreva a implementação de um método que retorne a representação no sistema hexadecimal do valor representado por um objeto dessa classe. O sistema hexadecimal apresenta dezesseis dígitos: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. O dígito A corresponde ao valor 10, o dígito B corresponde ao valor 11 e assim por diante. Para obter a representação hexadecimal de uma quantidade inteira expressa no sistema decimal, basta efetuar divisões inteiras sucessivas por 16. A concatenação dos dígitos correspondentes aos restos obtidos, na ordem inversa da obtenção, corresponderá à representação do valor no sistema hexadecimal.

20. Considere a classe **Fracao**, declarada de forma incompleta a seguir. Implemente os métodos e uma classe de **Teste** que invoque os métodos de cálculo.

```
public class Fracao {
    private int numerador;
    private int denominador;

    public Fracao( int num, int den) {
        numerador = num;
        denominador = den; }

    public int getNumerador() { }
    public int getDenominador() { }
    public void setNumerador( int n ) { }
    public void setDenominador( int d ){ }
    public Fracao fornecaSuaSomaCom( Fracao outra ) { }
    public Fracao fornecaSuaSubtracaoCom( Fracao outra ) { }
    public Fracao fornecaSuaDivisaoCom( Fracao outra ) { }
```

```

public Fracao fornecaSeuProdutoCom( Fracao outra ) { }

// Método privado que fornece o mdc entre n e m. Usado pelo método reduza_se
private int mdc ( int n, int m) { }

// Método que simplifica a fração representada pelo objeto, usando o mdc
public void reduza_se(){ }
}

```

21. Implemente a classe `Data`, apresentada abaixo, completando os métodos que não estão implementados. Escreva também a classe `Teste` que invoca os métodos da classe `Data`. Peça o valor da data para o usuário no formato dd/mm/aa (ex.: 13/02/97), e gere como saída a data escrita na forma dd/mês escrito por extenso/aa (ex.: 13/fevereiro/97).

```

public class Data {
    private int dia;
    private int mes;
    private int ano;

    public Data( int vDia, int vMes, int vAno) {
        dia = vDia;
        if ( dia < 1)
            dia = 1;
        else
            if ( dia > 31 )
                dia = 31;
        mes = vMes;
        if ( mes < 1)
            mes = 1;
        else
            if (mes > 12)
                mes = 12;
        ano = vAno;
        if ( ano < 1900 )
            ano = 1900;
        this.ajusteDia();
    }

    public void ajusteDia() {
        switch ( mes ) {
            case 2 : if ( dia >=28 )
                    if ( (ano % 4 == 0 && ano % 100 != 0)
                        || (ano % 400 = 0) )
                        dia = 29;
                    else
                        dia = 28;
                    break;
            case 4 :
            case 6 :
            case 9 :
            case 11 : if ( dia == 31)
                    dia = 30;
        } // fim switch
    } // fim ajusteDia()

    public String fornecaMesExtenso() { }

    public int getDia() { }

    public int getMes() { }

    public int getAno() { }

    public boolean e_Maior_Igual( Data outra ){ }

    // método que incrementa em um dia a data representada pela instância
    public void incrementaUmDia(){ }
}

```