

Lista de Exercícios Extra

PRIMEIRA LISTA DE EXERCÍCIOS EXTRAS – LOGPROG – Prof. Menezes

1) Uma corrida de avestruzes possui 15 participantes numerados de 0 a 14. Imagine um programa que leia o tempo em segundos de uma volta para cada um destes 15 participantes, armazene estes tempos num array. Crie métodos para realizar os seguintes cálculos:

Índice, o qual coincide com o número

do avestruz 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Tempo de uma volta do avestruz 58 34 29 56 38 45 42 63 28 47 54 67 34 42 41

a) O número do primeiro colocado, ou seja, o número do que deu a volta em menor tempo (para os valores acima, a resposta seria 8, visto que o avestruz de número 8 fez a volta em 28 segundos).

```
int metodoA (int array [ ])
```

```
{
```

```
....
```

```
}
```

b) O número do SEGUNDO colocado, ou seja, o número do que deu a volta no segundo menor tempo (para os valores acima, a resposta seria 2, visto que o avestruz de número 2 fez a volta em 29 segundos).

```
int metodoB (int array [ ])
```

```
{
```

```
....
```

```
}
```

c) A diferença de tempo entre o primeiro e o último colocado (para os valores acima, a resposta seria 39, visto que o avestruz mais lento fez a volta em 67 segundos, o mais rápido em 28 segundos, portanto $67 - 28 = 39$).

```
int metodoC (int array [ ])
```

```
{
```

```
....
```

```
}
```

R:

```
import javax.swing.*;
```

```
public class Vetores
```

```
{
```

```
    private int avestruzes[] = new int[15];
```

```
    public Vetores()
```

```
    {
```

```
    }
```

```
    public void setAvestruzes()
```

```
    {
```

```
        String tempo = "";
```

```
        for (int i = 0; i<avestruzes.length;i++)
```

```
        {
```

```
            tempo = JOptionPane.showInputDialog("Digite o tempo do avestruz  
nº " + i);
```

```
            avestruzes[i] = Integer.parseInt(tempo);
```

```
        }
```

```
    }
```

```
    private void setTempo(int pos, int tempo)
```

```
    {
```

```
        avestruzes[pos] = tempo;
```

```
}
```

```
public int getWinner()
```

```
{
```

```
    int menor;
```

```
    int numAve;
```

```
    menor = avestruzes[0];
```

```
    numAve = 0;
```

```
    for(int i = 1; i < avestruzes.length; i++)
```

```
    {
```

```
        if(menor > avestruzes[i])
```

```
        {
```

```
            menor = avestruzes[i];
```

```
            numAve = i;
```

```
        }
```

```
    }
```

```
    return numAve;
```

```
}
```

```
public int getLoser()
```

```
{
```

```
    int maior;
```

```
    int numAve;
```

```
    maior = avestruzes[0];
```

```
    numAve = 0;
```

```
    for(int i = 1; i < avestruzes.length; i++)
```

```
    {
```

```
        if(maior < avestruzes[i])
```

```
        {
```

```
            maior = avestruzes[i];
```

```
            numAve = i;
```

```
        }
```

```
    }
```

```
    return numAve;
```

```
}
```

```
private int getWinnerTime()
```

```
{
```

```
    int menor;
```

```
    int numAve;
```

```
    menor = avestruzes[0];
```

```
    numAve = 0;
```

```
    for(int i = 1; i < avestruzes.length; i++)
```

```
    {
```

```
        if(menor > avestruzes[i])
```

```
        {
```

```
            menor = avestruzes[i];
```

```
            numAve = i;
```

```
        }
```

```
    }
```

```
    return menor;
```

```
}
```

```
private int getLoserTime()
```

```
{
```

```
    int maior;
```

```

    int numAve;

    maior = avestruzes[0];
    numAve = 0;

    for(int i = 1; i < avestruzes.length; i++)
    {
        if(maior < avestruzes[i])
        {
            maior = avestruzes[i];
            numAve = i;
        }
    }
    return maior;
}

public int getTime()
{
    int menor = getWinnerTime();
    int maior = getLoserTime();
    int diferenca = maior - menor;

    return diferenca;
}
}

```

2) Uma empresa possui 20 linhas com modems que podem ser conectados a um provedor de acesso. Faça um programa que leia a quantidade de minutos gastos pelos modems (os guarde em uma coleção, como um array ou um ArrayList) e o custo em reais de cada minuto de uso. Implemente em Java:

- a) um método que calcule o gasto total em minutos;
- b) um método que calcule o gasto total em reais;
- c) um método que calcule o gasto médio;
- d) um método que imprima a posição na coleção e o valor gasto do modem com o maior gasto.

R:

```

import javax.swing.*;

public class Exercicio02
{
    private int minutos[] = new int[4];
    private double valor;

    public Exercicio02()
    {
        String min;
        for (int i=0; i < minutos.length; i++)
        {
            min = JOptionPane.showInputDialog("Digite os minutos " + i);
            minutos[i] = Integer.parseInt(min);
        }
        valor = Double.parseDouble(JOptionPane.showInputDialog("Digite o
valor em reais de cada minuto."));
    }

    public int ValorTotalMinutos()
    {
        int soma = 0;
        for(int i = 0; i < minutos.length; i++)
        {
            soma = soma + minutos[i];
        }
    }
}

```

```

    }
    return soma;
}

public double ValorTotalReais()
{
    double soma = 0;
    for(int i = 0; i < minutos.length; i++)
    {
        soma = soma + (minutos[i] * valor);
    }
    return soma;
}

public double ValorMedia()
{
    double soma = 0;
    double media = 0;
    for(int i = 0; i < minutos.length; i++)
    {
        soma = soma + (minutos[i] * valor);
    }
    media = soma / minutos.length;
    return media;
}

public void MaiorGasto()
{
    int maior = minutos[0];
    int posicao = 0;
    for(int i = 1; i < minutos.length; i++)
    {
        if(maior < minutos[i])
        {
            maior = minutos[i];
            posicao = i;
        }
    }
    System.out.println("posição " + posicao + " minutos " + maior);
}

}

```

3) Escreva um método que calcule e exiba todos os divisores de um dado número inteiro positivo. Ex.:

a) Os divisores de 10 são: 1, 2, 5, 10.

b) Os divisores de 12 são: 1, 2, 3, 4, 6, 12.

c) Os divisores de 11 são: 1, 11.

DICA: Um número X é divisível por um Y se o resto da divisão de X por Y for zero.

R:

```

public class Exercicio03
{

    public Exercicio03()
    {
    }

    public void Divisores(int numero)
    {
        String mensagem = "...

```

```

String mensagem = "Os divisores de ";
for(int i = 1; i <= numero; i++)
{
    if(numero% i == 0)
    {
        mensagem = mensagem + ", " + i;
    }
}
System.out.println("Os Divisores de "+ numero + ": " +
mensagem);
}
}

```

4) Faça um programa que leia pelo teclado um array inteiro, de tamanho 10, e informe quantas TRIPLAS de números seguidos, que formam uma Progressão Aritmética, estão no conjunto dos dados teclados. Observe o exemplo abaixo:

3 1 0 2 0 3 0 7 2 4 6

Com estes números, a resposta seria "há 2 TRIPLAS de números seguidos, que formam uma Progressão Aritmética (P.A.)", porque:

$$20 = (10 + 30)/2 \text{ e}$$

$$4 = (2 + 6)/2$$

DICA: Para verificar se 3 números estão em P.A., basta testar se o segundo é igual à média aritmética do primeiro com o terceiro número.

R:

```

import javax.swing.*;
public class Exercicio04
{
    private int numeros[] = new int [10];

    public Exercicio04()
    {
        for(int i = 0; i < numeros.length; i++)
        {
            numeros[i] = Integer.parseInt(JOptionPane.showInputDialog("
Digite o numero: "));
        }
    }

    public void Triplas()
    {
        int soma = 0;
        for(int i =0; i < numeros.length-2; i++)
        {
            if(numeros[i+1] == (numeros[i] + numeros[i+2])/ 2 )
            {
                soma++;
            }
        }
        System.out.println("numero de triplas: " + soma);
    }
}

```

5) Digitado um número inteiro entre 0 e 100, informar o quanto ele está distante de um determinado número chave, carregado no próprio programa. Ex.: Número chave=20, número digitado=15, resposta=5 Número chave=17, número digitado=20, resposta=3 (Obs.: a resposta deverá ser sempre um número positivo).

Exemplo 1 - Supondo que o número chave é 20 e o número inteiro digitado é 15, temos:

$$20 - 15 = 5$$

Resposta: 5

Resposta: 3

Exemplo 2 - Supondo que o número chave é 17 e o número inteiro digitado é 20, temos:

$17 - 20 = -3$

Resposta: 3

Obs.: se os números não estiverem entre 0 e 100, o usuário será avisado e não será informada a diferença entre os dois números.

R:

```
public int exercicio05(int numero)
{
    int numeroChave = 36;
    int resposta = numeroChave - numero;

    if(resposta < 0)
    {
        resposta = resposta * (-1);
    }
    return resposta;
}
```

6) Criar um algoritmo que calcule e classifique clientes de seguro a partir de seus dados pessoais e histórico de utilização deste serviço com esta ou outras companhias. Importante lembrar que os parâmetros aqui utilizados são falsos, exclusivos para o exercício. A classificação seguirá estes intervalos:

Idade

Abaixo de 18 anos não há cobertura

18 - 23 -> 4 ptos

24 - 29 -> 3 ptos

30 - 45 -> 2 ptos

46 - -> 1 pto

Gênero

F -> 2 ptos

M -> 3 ptos

Utilização

Passeio -> 1 pto

Trabalho -> 2 ptos

Ambos -> 3 ptos

Garagem

sem garagem -> 6 ptos

Diurna -> 4 ptos

Noturna -> 2 ptos

Ambas -> 1 pto

Classificação até 7 pontos -> Baixo / desc. 10%

8 - 10 -> médio / --

11 - 14 -> Alto / acresc. 10%

15 - 16 -> Altíssimo/acresc. 40%

O importante deste exercício é testar os valores de entrada, acumular os pontos de risco e a classificar a proposta.

R:

```
public void exercicio06(int idade, String sexo, String utilizacao,
String garagem)
{
    int pontos = 0;
    boolean cobertura = true;
    if(idade < 18)
    {
        System.out.println("Não há cobertura para menores de 18
```

```
anos.");
    cobertura = false;
}
else if(idade < 24)
{
    pontos+= 4;
}
else if(idade < 30)
{
    pontos+= 3;
}
else if(idade < 45)
{
    pontos+= 2;
}
else
{
    pontos+= 1;
}

if(cobertura == true)
{
    if(sexo == "F")
    {
        pontos+= 2;
    }
    else if(sexo == "M")
    {
        pontos+= 3;
    }

    if(utilizacao == "PASSEIO")
    {
        pontos+= 1;
    }
    else if(utilizacao == "TRABALHO")
    {
        pontos+= 2;
    }
    else if(utilizacao == "AMBOS")
    {
        pontos+= 3;
    }

    if(garagem == "SEM")
    {
        pontos+= 6;
    }
    else if(garagem == "DIURNA")
    {
        pontos+= 4;
    }
    else if(garagem == "NOTURNA")
    {
        pontos+= 2;
    }
    else if(garagem == "AMBAS")
    {
        pontos+= 1;
    }

    if(pontos <= 7)
```

```

        System.out.println("Baixo Risco - Desconto de 10%");
    }
    else if(pontos <= 10)
    {
        System.out.println("Médio Risco - Sem Desconto");
    }
    else if(pontos <= 14)
    {
        System.out.println("Alto Risco - Acrescimento de 10%");
    }
    else if(pontos <= 16)
    {
        System.out.println("Altíssimo Risco - Acrescimento de
40%");
    }
}
}
}

```

7) A prefeitura de uma cidade está pesquisando o nível de analfabetismo da sua população. As faixas etárias são: 7 - 18 anos e 19 - 25 anos. Considerando um universo de N pessoas, crie um programa em Java para calcular, para cada faixa etária, a porcentagem de analfabetos e exibir na tela a quantidade e a porcentagem de pessoas que constitui cada faixa. Utilizar como sinalizador de fim o valor -1 para a idade.

R:

```

import javax.swing.*;

public class exercicio07
{
    private int faixaEtaria01total = 0;
    private int faixaEtaria02total = 0;
    private int faixaEtaria01analfabeta = 0;
    private int faixaEtaria02analfabeta = 0;

    public exercicio07()
    {
    }

    public void funcExercicio07()
    {
        int idade = 0;
        String alfabetizada = "";
        double porc01;
        double porc02;

        do
        {
            idade = Integer.parseInt(JOptionPane.showInputDialog("Digite
a idade da pessoa.));
            alfabetizada = JOptionPane.showInputDialog("A pessoa é
alfabetizada? (S/N)");

            if(idade >= 7 && idade <=18)
            {
                faixaEtaria01total++;
                if(alfabetizada.equals("N"))
                {
                    faixaEtaria01analfabeta++;
                }
            }
        }
    }
}

```

```

        else if (idade >= 19 && idade <= 25)
        {
            faixaEtaria02total++;
            if(alfabetizada.equals("N"))
            {
                faixaEtaria02analfabeta++;
            }
        }
    }while(idade > -1);

    if(faixaEtaria01total != 0)
    {
        porc01 = faixaEtaria01analfabeta / faixaEtaria01total * 100;
    }
    if(faixaEtaria02total != 0)
    {
        porc02 = faixaEtaria02analfabeta / faixaEtaria02total * 100;
    }
    System.out.println("Faixa etária 01 (7-18) total = " +
faixaEtaria01total);
    System.out.println("Faixa etária 01 (7-18) analfabeta = " +
faixaEtaria01analfabeta);
    System.out.println("Faixa etária 02 (19-25) total = " +
faixaEtaria02total);
    System.out.println("Faixa etária 02 (19-25) analfabeta = " +
faixaEtaria02analfabeta);
    System.out.println("");
}
}
}

```

8) Uma farmácia comercializa 5 produtos diferentes. Elaborar um programa-fonte em Linguagem Java que calcule o faturamento mensal, o qual é calculado pela somatória do faturamento diário que é obtido pela multiplicação do valor unitário de cada produto pela quantidade de produtos vendida diariamente. Considerar mês de 30 dias.

Exibir o resultado na tela com mensagens adequadas.

R:

```

import javax.swing.*;

public class exercicio08
{
    int produtosVendidos[];
    int precoProdutos[];
    int vendasMes[] = new int[30];

    public exercicio08(int qtdProdutos)
    {
        produtosVendidos = new int[qtdProdutos];
        precoProdutos = new int[qtdProdutos];

        for(int i = 0; i < precoProdutos.length; i++)
        {
            precoProdutos[i] = Integer.parseInt
(JOptionPane.showInputDialog("Digite o preço do          produto
" + (i+1)));
        }
    }

    public void efetuaCompra(int produto, int qtd)
    {
        if(produto-1 >= produtosVendidos.length && produto < 1)
        {

```

```

        System.out.println("Valor errado");
    }
    else
    {
        produtosVendidos[produto-1]+= qtd;
    }
}

public void mostrarVendaHoje()
{
    for(int i = 0; i < produtosVendidos.length; i++)
    {
        System.out.println("Produto " + (i+1) + " - " +
produtosVendidos[i] + " vendas");
    }
}

public void fecharCaixa(int dia)
{
    int total = 0;
    for(int i = 0; i < produtosVendidos.length; i++)
    {
        total = total + (produtosVendidos[i] * precoProdutos[i]);
        System.out.println("Produto " + (i+1) + " - R$" + total);
    }
    vendasMes[dia] = total;
    System.out.println("Total - R$" + total);
}

}

```

9) Desenvolva um método que:

a) vasculhe um array de números inteiros e que retorne quantos números pares há neste array.

```
int metodoA (int array [ ])
```

```
{.
```

```
...
```

```
}
```

b) vasculhe um array de números inteiros e que retorne quantos números há neste array entre 10 (inclusive) e 15 (inclusive).

```
int metodoB (int array [ ])
```

```
{.
```

```
...
```

```
}
```

c) vasculhe um array de números inteiros em busca de um número procurado. O retorno, caso ele seja encontrado, será o índice dele no array; caso ele não exista no array, o valor retornado será -1.

```
int metodoC (int array [ ])
```

```
{.
```

```
...
```

```
}
```

R:

```

public int ExercicioA(int array[])
{
    int soma = 0;
    for(int i =0; i < array.length; i++)
    {
        if(array[i] % 2 == 0)
        {
            soma++;
        }
    }
}

```

```

        return soma;
    }

    public int ExercicioB(int array[])
    {
        int soma = 0;
        for(int i =0; i < array.length; i++)
        {
            if(array[i] >= 10 && array[i] <= 15)
            {
                soma++;
            }
        }
        return soma;
    }

    public int ExercicioC(int array[], int n)
    {
        int soma = 0;
        for(int i =0; i < array.length; i++)
        {
            if(array[i] == n)
            {
                return i;
            }
        }
        return -1;
    }
}

```

10) Uma empresa de locação de veículos cobra pelo uso de seus veículos populares a partir da regra abaixo:

- 45 reais para cada dia de uso como custo fixo;
- 0.50 reais para cada quilômetro que exceder o total de quilômetros dados de cortesia. A cada dia o usuário tem direito a 60 quilômetros de cortesia.

O usuário efetua o pagamento somente ao devolver o carro à empresa de locação. Exemplificando, se o usuário ficou com o carro 5 dias e “rodou” 700 quilômetros, ele deve pagar $5 \cdot 45 + 0.5 \cdot (700 - 5 \cdot 60) = 425$. Se o usuário ficou com o carro 18 dias e “rodou” 18 quilômetros ele deve pagar $18 \cdot 45 = 810$.

Escreva um programa em Java que leia o número de dias que o usuário ficou com o carro e a quilometragem do mesmo e imprima o valor pago.

R:

```

public double Exercicio10(int dias, int km)
{
    int kmExcesso;
    kmExcesso = km - dias * 60;
    if(kmExcesso > 0)
    {
        return (dias * 45) + (0.5 * kmExcesso);
    }
    else
    {
        return (dias * 45);
    }
}

```

11) Faça um programa em Java que leia quatro números do teclado e mostre se houve ou não algum número repetido. Por exemplo:

Números digitados: 2, 2, 4, 7

Resultado: Houve número repetido.

Números digitados: 5, 1, 5, 5

Resultado: Houve número repetido.

Números digitados: 1, 1, 1, 1

Resultado: Houve número repetido.

Números digitados: 9, 1, 4, 3

Resultado: Não houve número repetido.

R:

```
public void numerosRepetidos(int n1, int n2, int n3, int n4)
{
    if(n1 == n2 || n1 == n3 || n1 == n4 || n2 == n3 || n2 == n4 ||
n3 == n4)
    {
        System.out.println("Existem números repetidos.");
    }
    else
    {
        System.out.println("Não existem números repetidos.");
    }
}
```