

## Programação Modular – Exercício de Nivelamento 00

### Questão 01 -

```
package exercicios.Nivelamento;

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Questao01 {
    public static void main(String[] args) {
        int largura = readNumber();
        int altura = readNumber();

        printCloseLines(largura);
        printEmptyLines(altura, largura);
        printCloseLines(largura);
    }

    private static void printEmptyLines(int altura, int largura) {
        for (int i = 0; i < altura - 2; i++) {
            System.out.print("X");
            for (int j = 0; j < largura - 2; j++) {
                System.out.print(" ");
            }
            System.out.print("X");
            System.out.println();
        }
    }

    private static void printCloseLines(int largura) {
        for (int i = 0; i < largura; i++) {
            System.out.print("X");
        }
        System.out.println();
    }

    public static int readNumber() {
        try {
            InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(isr);
            return Integer.parseInt(br.readLine());
        } catch (Exception ex) {
            System.out.println("Número invalido, digite novamente");
            return readNumber();
        }
    }
}
```

### Questão 02 -

```
package exercicios.Nivelamento;

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Questao02 {
```

```

public static void main(String[] args) {
    menuProgram();
}

private static void menuProgram() {
    printMenu();
    int opcode = Questao01.readNumber();
    System.out.println("=====");
    switch (opcode) {
        case 1:
            sumTwoNumbers();
            menuProgram();
        case 2:
            greatherThan();
            menuProgram();
        case 3:
            sumNNumbers();
            menuProgram();
        case 4:
            pairNumbers();
            menuProgram();
        case 0:
            System.exit(0);
        default:
            System.out.println("Opção Invalida");
            menuProgram();
    }
}

private static void pairNumbers() {
    System.out.println("Digite uma sequência de numeros separados por espaço: ");
    String sequencia = readLine();
    String[] numeros = sequencia.split(" ");
    int contador = 0;
    for (int i = 0; i < numeros.length; i++) {
        int number = numberTransform(numeros[i]);
        numeros[i] = String.valueOf(number);
        if (number % 2 == 0) {
            contador++;
        }
    }
    printSequence(numeros);
    System.out.println();
    System.out.println("Essa sequência tem: " + contador + " números pares");
}

private static void printSequence(String[] numeros) {
    for (String numero : numeros) {
        System.out.print(numero);
        System.out.print(" ");
    }
}

private static int numberTransform(String numero) {
    try {
        return Integer.parseInt(numero);
    } catch (Exception ex) {
        System.out.println("Número: " + numero + " invalido, digite novamente");
        return Questao01.readNumber();
    }
}

```

```

private static void sumNNumbers() {
    System.out.println("Quantidade de números a serem somados: ");
    int qnt = Questao01.readNumber();
    System.out.println("Digite o primeiro numero: ");
    int auxiliar = 0;
    for (int i = 0; i < qnt; i++) {
        auxiliar = auxiliar + Questao01.readNumber();
        System.out.println "[" + (i + 2) + "] Qual valor deve ser somado: ";
    }
    System.out.println("O resultado da soma dos números é: " + auxiliar);
}

private static void greatherThan() {
    System.out.println("Digite o primeiro número: ");
    int num1 = Questao01.readNumber();
    System.out.println("Digite o segundo número: ");
    int num2 = Questao01.readNumber();
    if (num1 >= num2) {
        System.out.println("O maior número é: " + num1);
    } else {
        System.out.println("O maior número é: " + num2);
    }
}

private static void sumTwoNumbers() {
    System.out.println("Digite o primeiro número: ");
    int num1 = Questao01.readNumber();
    System.out.println("Digite o segundo número: ");
    int num2 = Questao01.readNumber();
    System.out.println("Resultado: " + (num1 + num2));
}

private static void printMenu() {
    System.out.println("=====");
    System.out.println("1 - Somar dois números");
    System.out.println("2 - O maior de dois números");
    System.out.println("3 - Somar N números");
    System.out.println("4 - Contador de pares de uma sequência");
    System.out.println("0 - Sair");
    System.out.println("=====");
    System.out.println("Opção: ");
}

public static String readLine() {
    try {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(isr);
        return br.readLine();
    } catch (Exception ex) {
        System.out.println("Números invalidos, digite novamente");
        return readLine();
    }
}
}

```

```

package exercicios.Nivelamento;

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Questao03 {
    public static void main(String[] args) {
        System.out.println("=====");
        double[] exercicios = new double[4];
        double[] provas = new double[2];
        double[] trabalhos = new double[1];

        extrairNotas(4, "Digite a nota do Exercício ", exercicios, 100);
        System.out.println("=====");
        extrairNotas(2, "Digite a nota da Prova ", provas, 100);
        System.out.println("=====");
        extrairNotas(1, "Digite a nota do Trabalho ", trabalhos, 20);
        System.out.println("=====");

        double notaExercicio = getSomaTotal(exercicios);
        double notaProva = getSomaTotal(provas);
        double notaTrabalho = getSomaTotal(trabalhos);

        double notaGeral = ((notaExercicio / 4) * 0.2) + ((notaProva / 2) * 0.6) + (notaTrabalho *
0.2);
        String resultado = String.format("%.2f", notaGeral);
        System.out.println("Resultado: " + resultado);
    }

    private static void extrairNotas(int x, String x1, double[] exercicios, int total) {
        for (int i = 0; i < x; i++) {
            System.out.println(x1 + (i + 1) + ":");
            exercicios[i] = validateNotes();
        }
    }

    private static double validateNotes() {
        double valor = readDouble();
        if (valor > 100 || valor < 0) {
            System.out.println("Nota inválidas!");
            return validateNotes();
        }
        return valor;
    }

    private static double getSomaTotal(double[] exercicios) {
        double somaTotal = 0;
        for (double exercicio : exercicios) {
            somaTotal += exercicio;
        }
        return somaTotal;
    }

    public static double readDouble() {
        try {
            InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(isr);
            return Double.parseDouble(br.readLine());
        } catch (Exception ex) {
    
```

```

        System.out.println("Número inválido, digite novamente");
        return readDouble();
    }
}
}

```

Questão 04 -

```

package exercicios.Nivelamento;

public class Questao04 {
    public static void main(String[] args) {
        String entrada = Questao02.readLine();
        int colunas = getColunas(entrada);
        char[][] matriz = new char[colunas][5];

        populateMatriz(colunas, matriz, entrada);
        printMatriz(colunas, matriz);
    }

    public static void printMatriz(int colunas, char[][] matriz) {
        for(int i = 0; i < 5; i++) {
            for(int j = 0; j < colunas; j++) {
                System.out.print(matriz[j][i]);
            }
            System.out.print("\n");
        }
    }

    public static void populateMatriz(int colunas, char[][] matriz, String entrada) {
        int posicao = 0;
        for(int i = 0; i < colunas; i++) {
            for(int j = 0; j < 5; j++) {
                matriz[i][j] = getChar(entrada, posicao);
                posicao++;
            }
        }
    }

    public static int getColunas(String entrada) {
        int colunas;
        if(entrada.length() % 5 == 0) {
            colunas = entrada.length()/5;
        } else {
            colunas = entrada.length()/5+1;
        }
        return colunas;
    }

    public static char getChar(String entrada, int i) {
        try {
            return entrada.charAt(i);
        } catch (Exception e) {
            return ' ';
        }
    }
}

```

Questão 05 -

```
package exercicios.Nivelamento;

import java.util.Arrays;

import static exercicios.Nivelamento.Questao04.*;

public class Questao05 {

    public static void main(String[] args) {
        String entrada = Questao02.readLine();
        int colunas = getColunas(entrada);
        char[][] matriz = createMatriz(colunas);

        populateMatrizInvertida(colunas, matriz, entrada);

        for (int i = 0; i < colunas; i++) {
            for (int j = 0; j < 5; j++) {
                System.out.print(matriz[i][j]);
            }
        }
    }

    private static char[][] createMatriz(int colunas) {
        char[][] matriz = new char[colunas][5];
        Arrays.stream(matriz).forEach(c -> Arrays.fill(c, ' '));
        return matriz;
    }

    public static void populateMatrizInvertida(int colunas, char[][] matriz, String entrada) {
        int posicao = 0;
        for(int i = 0; i < 5; i++) {
            for(int j = 0; j < colunas; j++) {
                if (entrada.charAt(posicao) == '*') {
                    j = colunas;
                } else {
                    matriz[j][i] = entrada.charAt(posicao);
                }
                posicao++;
            }
        }
    }
}
```