

Avaliação 1º BIM - 2º DSM - VALOR 7,0

01) (VALOR 4.0) Calcular o valor total de uma compra com IPI, utilizando classes separadas para armazenar dados e realizar cálculos.

Requisitos:

- Criar uma classe **CalculadoraIPI** que:
 - Receba a porcentagem do IPI como construtor.
 - Possua um método **calcularValorTotalComIpi(valorTotalSemIpi)** que:
 - Receba o valor total sem IPI como parâmetro.
 - Calcule o valor do IPI.
 - Retorne o valor total com IPI.
- Criar uma classe **DadosCompra** que:
 - Armazene os dados da compra:
 - Código da peça 1.
 - Valor unitário da peça 1.
 - Quantidade da peça 1.
 - Código da peça 2.
 - Valor unitário da peça 2.
 - Quantidade da peça 2.
 - Possua um método **calcularValorTotalSemIpi()** que:
 - Calcule o valor total sem IPI.
 - Retorne o valor total sem IPI.
- Criar uma classe **Main** que:
 - Leia a porcentagem do IPI do usuário.
 - Leia os dados da compra do usuário.
 - Crie objetos das classes **CalculadoraIPI** e **DadosCompra**.
 - Calcule o valor total sem IPI.
 - Calcule o valor total com IPI.
 - Imprima os resultados:
 - Valor total dos produtos sem IPI.
 - Valor total dos IPI's.
 - Valor total dos produtos com IPI.

Exemplo de Entrada/Saída:

```
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ primeiroProjeto ---
Digite a porcentagem do IPI: 5
**Dados da Peça 1:**
Código: 001
Valor unitário: 15
Quantidade: 3
**Dados da Peça 2:**
Código: 002
Valor unitário: 25
Quantidade: 5

**Resultados:**
Valor total dos produtos sem IPI: R$ 170,00
Valor total do IPI: R$ 178,50
Valor total dos produtos com IPI: R$ 178,50
-----
```

02) (VALOR 2.0) Considere um vetor de 10 números inteiros positivos maiores que zero e um único número X inteiro, também positivo e maior que zero. Faça uma classe JAVA chamada **“MaioresMenoresIguais”** para:

- (a) ler pelo teclado os valores do vetor;
- (b) ler pelo teclado o número X;
- (c) dizer quantos números no vetor são maiores que X, menores que X e iguais a X

Exemplo de Entrada/Saída:

```
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ primeiroProjeto ---
Digite os 10 números do vetor:
10
2
5
8
1
3
6
9
4
7
Digite o número X:
5
Maiores que X: 5
Menores que X: 4
Iguais a X: 1
-----
```

03) (VALOR 1.0) Faça uma classe em JAVA chamada “**Temperatura**”, para que, com base em uma temperatura recebida pelo usuário em graus celsius, a converta e exiba em Kelvin (K), Réaumur (Re), Rankine (Ra) e Fahrenheit (F), seguindo as fórmulas: $F = C * 1.8 + 32$; $K = C + 273.15$; $Re = C * 0.8$; $Ra = C * 1.8 + 32 + 459.67$

Exemplo de Entrada/Saída:

```
-----[ jar ]-----  
  
--- exec-maven-plugin:3.0.0:exec (default-cli) @ primeiroProjeto ---  
Digite a temperatura em Celsius: 2,43  
Fahrenheit: 36,37°F  
Kelvin: 275,58K  
Réaumur: 1,94°Ré  
Rankine: 496,04°R  
-----
```

ALGUNS EXEMPLOS DOS EXERCÍCIOS EM SALA

```
import java.util.Scanner;  
  
You, ontem | 1 author (You)  
public class Fatorial {  
    Run | Debug  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print(s:"Digite um número inteiro positivo: ");  
        int n = scanner.nextInt();  
        int fatorial = 1;  
        for (int i = 2; i <= n; i++) {  
            fatorial *= i;  
        }  
        System.out.println("O fatorial de " + n + " é: " + fatorial);  
        scanner.close();  
    }  
}
```

```
import java.util.Scanner;
```

You, ontem | 1 author (You)

```
public class Fibonacci {
```

Run | Debug

```
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Digite um número inteiro positivo: ");
        int n = scanner.nextInt();
        int a = 0, b = 1, c;
        System.out.print("Sequência de Fibonacci até " + n + ": ");
        System.out.print(a + " " + b + " ");
        for (int i = 2; i <= n; i++) {
            c = a + b;
            System.out.print(c + " ");
            a = b;
            b = c;
        }
        scanner.close();
    }
}
```

```
import java.util.Scanner;
```

You, ontem | 1 author (You)

```
public class NumeroPerfeito {
```

Run | Debug

```
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Digite um número inteiro positivo: ");
        int n = scanner.nextInt();
        int somaDivisores = 0;
        for (int i = 1; i < n; i++) {
            if (n % i == 0) {
                somaDivisores += i;
            }
        }
        if (somaDivisores == n) {
            System.out.println(n + " é um número perfeito.");
        } else {
            System.out.println(n + " não é um número perfeito.");
        }
        scanner.close();
    }
}
```

```
public class NumerosPrimos2 {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Digite um número inteiro positivo: ");
        int n = scanner.nextInt();
        System.out.print("Números primos menores ou iguais a " + n + ": ");
        for (int i = 2; i <= n; i++) {
            int divisors = 0;
            for (int j = 2; j <= i / 2; j++) {
                if (i % j == 0) {
                    divisors++;
                }
            }
            if (divisors == 0) {
                System.out.print(i + " ");
            }
        }
        scanner.close();
    }
}
```

```
public class NumerosPrimos2 {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Digite um número inteiro positivo: ");
        int n = scanner.nextInt();
        System.out.print("Números primos menores ou iguais a " + n + ": ");
        for (int i = 2; i <= n; i++) {
            int divisors = 0;
            for (int j = 2; j <= i / 2; j++) {
                if (i % j == 0) {
                    divisors++;
                }
            }
            if (divisors == 0) {
                System.out.print(i + " ");
            }
        }
        scanner.close();
    }
}
```

```

public class SomaNumeros {
    Run | Debug
    public static void main(String[] args) {
        int soma = 0;
        for (int i = 1; i <= 100; i++) {
            soma += i;
        }
        System.out.println("A soma dos números de 1 a 100 é: " + soma);
    }
}

```

```

import java.util.Scanner;

You, ontem | 1 author (You)
public class TabuadaMultiplicacao {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Digite um número inteiro: ");
        int n = scanner.nextInt();
        System.out.println("Tabuada de multiplicação de " + n + ":");
        for (int i = 1; i <= 10; i++) {
            System.out.println(n + " x " + i + " = " + (n*i));
        }
        scanner.close();
    }
}

```

```

public class ContarParesImpares {
    You, há 1 segundo • Uncommitted
    Run | Debug
    public static void main(String[] args) {

        int[] vetor = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13};
        int pares = 0;
        int impares = 0;

        for (int i = 0; i < vetor.length; i++) {
            if (vetor[i] % 2 == 0) {
                pares++;
            } else {
                impares++;
            }
        }

        System.out.println("Quantidade de números pares: " + pares);
        System.out.println("Quantidade de números ímpares: " + impares);
    }
}

```

```
public class ConversaoDeDados1 {
    public static double quilometroParaMetro(double quilometro){
        return quilometro * 1000;
    }
    public static double litroParaDecilitros(double litro){
        return litro * 10;
    }
    public static double metroCubicoParaPesCubicos(double metroCubico){
        return metroCubico * 35.31;
    }
    public static double barrilParaDecalitros(double barril){
        return barril * 16.36;
    }
    public static double barrilParaLitros(double barril){
        return barril * 163.65;
    }
}
```

```
public class Teste1 {
    Run | Debug
    public static void main(String[] args) {

        double quilometro = 2.8;
        double litro = 4.5;
        double metroCubico = 2.5;
        double barril = 3.0;

        System.out.println(quilometro + " quilômetros correspondem a "
            + ConversaoDeDados1.kilometroParaMetro(quilometro) + " metros.");
        System.out.println(litro + " litros correspondem a "
            + ConversaoDeDados1.litroParaDecilitros(litro) + " decilitros.");
        System.out.println(metroCubico + " metros cúbicos correspondem a "
            + ConversaoDeDados1.metroCubicoParaPesCubicos(metroCubico) + " metros.");
        System.out.println(barril + " barris correspondem a "
            + ConversaoDeDados1.barrilParaDecalitros(barril) + " decalitros.");
        System.out.println(barril + " barris correspondem a "
            + ConversaoDeDados1.barrilParaLitros(barril) + " litros.");
    }
}
```

```

public class ConversaoDeDados2 {
    public double quilometroParaMetro(double quilometro){
        return quilometro * 1000;
    }
    public double litroParaDecilitros(double litro){
        return litro * 10;
    }
    public double metroCubicoParaPesCubicos(double metroCubico){
        return metroCubico * 35.31;
    }
    public double barrilParaDecalitros(double barril){
        return barril * 16.36;
    }
    public double barrilParaLitros(double barril){
        return barril * 163.65;
    }
}

```

```

public class Teste2 {
    Run | Debug
    public static void main(String[] args) {
        ConversaoDeDados2 obj = new ConversaoDeDados2();
        double quilometro = 2.8;
        double litro = 4.5;
        double metroCubico = 2.5;
        double barril = 3.0;

        System.out.println(quilometro + " quilômetros correspondem a "
            + obj.quilometroParaMetro(quilometro) + " metros.");
        System.out.println(litro + " litros correspondem a "
            + obj.litroParaDecilitros(litro) + " decilitros.");
        System.out.println(metroCubico + " metros cúbicos correspondem a "
            + obj.metroCubicoParaPesCubicos(metroCubico) + " metros.");
        System.out.println(barril + " barris correspondem a "
            + obj.barrilParaDecalitros(barril) + " decalitros.");
        System.out.println(barril + " barris correspondem a "
            + obj.barrilParaLitros(barril) + " litros.");
    }
}

```



```
//Classe Conta1 que contém uma variável de instância NOME
//e métodos para configurar/inserir e obter seu valor
public class Conta1 {
    private String nome; //variável de instância

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getNome() {
        return nome;
    }
}
```

```
public class TesteConta1 {
    Run | Debug
    public static void main(String[] args) {
        String name;

        //cria um objeto scanner para obter entrada a partir da janela de comando
        Scanner input = new Scanner(System.in);

        //cria um objeto a partir da classe Conta1() e atribui a minhaConta
        Conta1 minhaConta = new Conta1();
        You, há 4 semanas • first commit
        //exibe o valor inicial do nome (....)
        System.out.printf(format:"O nome inicial é: %s\n ", minhaConta.getNome());

        //solicita e lê o nome
        System.out.println(x:"Por favor, entre com o nome: ");
        name = input.nextLine(); //lê um linha de texto
        minhaConta.setNome(name); //insere name em minhaConta

        System.out.println();

        //exibe o nome armazenado no objeto minhaConta
        System.out.printf(format:"O nome do objeto em minhaConta é: %s\n", minhaConta.getNome());
    }
}
```

```
// a classe Conta2 com um construtor que inicializa o nome
public class Conta2 {

    private String nome;

    //o construtor inicializa "nome" com o mesmo nome do parâmetro
    public Conta2(String nome){
        this.nome = nome;
    }

    //método para recuperar o nome
    public String getNome() {
        return nome;
    }

    //método para inserir o nome
    public void setNome(String nome) {
        this.nome = nome;
    }

}

} // fim da classe Conta2
```

```
//usando o construtor de Conta2 para inicializar a instância nome no momento em que cada
//objeto Conta2 é criado
public class TesteConta2 {
    Run | Debug
    public static void main(String[] args) {
        //cria dois objetos Conta2
        Conta2 minhaConta1 = new Conta2(nome:"Alexandre");
        Conta2 minhaConta2 = new Conta2(nome:"Gomes");

        //exibe o valor de cada nome para cada Conta2
        System.out.printf(format:"Os nome dos objetos são: %s %s",
            minhaConta1.getNome(), minhaConta2.getNome());
    }
}
```

}

```
public static void main(String[] args) {
```

```
Conta1 primeiroObj = new Conta1();
```

```
String name = entrada.nextLine();
```

```
Conta2 segundoObj = new Conta2(nome: "Gomes");
```

```
Conta2 terceiroObj = new Conta2(nome:"Silva");
```

```
primeiroObj.getNome(), segundoObj.getNome(), terceiroObj.getNome());
```

}