



Aluno(a): _____

Turma: _____

Data: _____

Utilize atributos de instância privados sempre que possível.
Faça distinção clara entre maiúsculas e minúsculas.
Seja claro, formal e sucinto.
Códigos desnecessários e que reduzam o desempenho do sistema serão penalizados.
Utilize as boas práticas de programação.
LEIA AS QUESTÕES ATÉ O FINAL ANTES DE COMEÇAR.

Questão 1 (6.5) – Desenvolva o código conforme pedido abaixo:

A - Crie a interface `Eletronico` com dois métodos denominados `ligar` e `desligar`, que não possuem argumentos e que não retornam nada.

B₁ – Escreva uma classe `Produto` com 3 atributos: `id` (`String`), `cor` (`String`) e `preco` (`double`). Crie os getters e setters apenas se precisar. Crie em `Produto` APENAS UM construtor, que recebe o `id` como argumento. Implemente um método em uma classe chamada `Utils` com a seguinte assinatura: `public static boolean existe (List x, Produto y)`; considere que a lista possui objetos do tipo `Produto`, mas podendo possuir objetos de outros tipos. Escreva esse método de forma que seja verificada a existência do objeto `Produto` representado por `y` na lista representada por `x`, retorne verdadeiro se existir e falso se não existir. Considere que dois objetos `Produto` são iguais se possuem o **mesmo id**. Não é permitida qualquer iteração para realizar esse item, ou seja, não use `for`, `iterator`, etc.

B_{1.5} – Dado que `b` é uma referência a um objeto do tipo `Produto`, implemente o código que faz com que, caso essa referência seja impressa com o `System.out.println`, seja apresentado no console o `id`, `cor` e `preço` do produto no seguinte formato: `[id – cor - preço]`. Ex: `[COD001 – amarelo 3000.67]`.

B₂- Crie 1 subclasse da classe `Produto`: `Notebook`. Implemente a interface `Eletronico`.

C₁- Considere um método denominado **`Utils2.getLista(): List`** que retorne uma lista de objetos de diversos tipos, incluindo `Produtos`. Não crie nem a classe `Utils2` nem o método `getLista()`, suponha que já existe. (Não implemente nada nesse item, são apenas informações.)

C₂- Crie uma classe principal que receba do console o `id` de um `Produto`. Em seguida, verifique, com base no código desenvolvido no item B₁, se o `Produto` existe na lista retornada por **`Utils2.getLista()`**. Caso sim, exiba no console o `id`, `cor` e `preço` do `Produto` existente na lista retornada ao invocar o método **`Utils2.getLista()`**.

C₃ – Ainda na classe principal, crie um método denominado `transformaListEmArray` que receba uma lista `k` de objetos (de diversos tipos (ex: `Elefante`, `Carro`, `Conta`), incluindo `Produtos`) e

retorna um array de Produtos, contendo os objetos do tipo Produto na lista representada por **k**. O array DEVE ter o comprimento do número de Produtos em k. Ex: se em k existem 10 Produtos e 15 Carros, o array DEVE ter tamanho 10 e estar preenchido com os 10 Produtos.

D – Em seguida, crie um método para retornar a soma de todos os preços no array de Produtos criado no item C₃.

Questão 2 (1,0) Crie uma classe denominada **Media** com um método estático denominado **calcula (String numeros)** que retorna um **double**. Considere que o formato da String que esse método receberá é composto por N números separados por pelo caractere **#**. Ex1: 15#3#8; Ex2: 3#1#1#1#4

Retorne a média dos números inteiros presentes nessa String. Por exemplo, para o EX2, o retorno seria 2.

Questão 3 (0,25) Qual o método da classe String que, se invocado, retorna um array de caracteres representando essa String.

Questão 4 (0,25) Qual o método da classe String que, se invocado, retorna a String toda em letras minúsculas?

Questão 5 (2,0) Observe as classes abaixo e diga o que sai no console.

```
public class Lampada {
    private String cor;
    public static int teste;
    private int horasUteis;
    public Lampada() {
        ++teste;
    }
    public String getCor() {
        return cor;
    }
    public void setCor(String cor) {
        this.cor = cor;
    }
    public int getHorasUteis() {
        return horasUteis;
    }
    public void setHorasUteis(int idade) {
```

```

        this.horasUteis = idade;
    }
}

```

```

public class TesteLampada {
    public static void main(String[] args) {
        Lampada x = new Lampada();
        Lampada y = x;
        x.setCor("Branca");
        System.out.println(y.getCor());
        y.setCor("Rosa");
        x = new Lampada();
        y = x;
        System.out.println(y.getCor());
        y.setHorasUteis(2200);
        y.teste = y.teste + 15;
        System.out.println(x.teste);
        teste1(y, x);
        System.out.println(x.getCor());
        System.out.println(y.getCor());
        int k = 9;
        teste2(k, x.getHorasUteis(), x);
        System.out.println(x.teste);
        y.teste = y.teste + 3;
        System.out.println(x.teste);
        System.out.println(k);
        System.out.println(x.getCor());
        System.out.println(x.getHorasUteis());
    }
    public static void teste1(Lampada x, Lampada y) {
        x.setCor("Azul");
        x = new Lampada();
        x.setCor("Verde");
        y.setCor("Amarela");
        y = x;
        x = y;
        System.out.println(x.getCor());
    }
    public static Lampada teste2(int a, int b, Lampada x) {
        a = 11;
        b = 8;
        x.setCor("Laranja");
        x.teste = x.teste + 7;
        System.out.println("valor: " + a + b);
        return x;
    }
}

```

} }