



Aluno(a): _____

Turma: _____

Data: _____

Utilize atributos de instância privados sempre que possível.
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 (5,5) – Desenvolva o código conforme pedido abaixo:

A - Crie a interface Comestivel com apenas um método denominado comer, que não possui argumentos e que não retorna nada.

B₁ – Escreva uma classe Biscoito com 3 atributos: id (String), cor (String) e preco (int). Crie os getters e setters apenas se precisar. Crie em Biscoito 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, Biscoito y); considere que a lista possui objetos do tipo Biscoito, mas podendo possuir outros objetos. Escreva esse método de forma que seja verificada a existência do objeto Biscoito representado por y na lista representada por x, retorne verdadeiro se existir e falso se não existir. Considere que dois objetos Biscoito são iguais se possuem o **mesmo id**. Não é permitida qualquer iteração para realizar esse item, ou seja, não use *for*, *iterator*, etc.

B₂- Crie 2 subclasses da classe Biscoito: Negresco e Skiny. Implemente a interface Comestivel.

C₁- Considere um método denominado **Utils2.getLista(): List** que retorne uma lista de objetos de diversos tipos, incluindo Biscoitos. Não crie nem a classe Utils2 nem o método getLista(), suponha que já existe. Crie uma classe principal que receba do console o id de um Biscoito. Em seguida, verifique, com base no código desenvolvido no item B, se o Biscoito existe na lista retornada por **Utils2.getLista()**. Caso sim, exiba no console uma mensagem informando o id, cor e preço do Biscoito.

C₂ – Ainda na classe principal, crie um método denominado transformListEmArray que receba uma lista **k** de objetos (de diversos tipos (ex: Elefante, Carro, Conta), incluindo Biscoitos) e retorne um array de Biscoitos, contendo os objetos do tipo Biscoito na lista representada por **k**. O array DEVE ter o comprimento do número de Biscoitos em k. Ex: se em k existem 10 Biscoitos e 15 Carros, o array DEVE ter tamanho 10 e estar preenchido com os 10 Biscoitos.

D – Em seguida, crie um método para calcular a soma de todos os valores dos preços dos biscoitos no array de Biscoitos criado no item C₂. Imprima o total no console.

Questão 2 (1,7) Receba até 1000 Strings do teclado. Imprima uma saída informando que o formato da String deve ser: nome#peso#altura. Imprima outra saída informando que para encerrar a inserção de dados o usuário deve inserir “finalizar” no console.

Ex: Aline Ferraz#79.7#1.77

Em seguida, calcule a média dos pesos das pessoas inseridas.

Questão 3 (0,8) – Crie um método estático chamado inverter (String str) que receba uma String e retorne a mesma String invertida.

Questão 4 (2) Observe as classes abaixo e diga o que sai no console.

```
public class Arvore {  
    private String nome;  
    public int a;  
    public static int b;  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
}
```

```

public class TesteArvore {
    public static void main(String[] args) {
        Arvore v1 = new Arvore();
        System.out.println(v1.getNome());
        v1.setNome("Acacia");
        Arvore v2 = new Arvore();
        v2.setNome("Ipe");
        teste1(v1, v2);
        System.out.println(v1.getNome());
        System.out.println(v2.getNome());
        v1.a = 10;
        int x = v1.a;
        v1.a=5;
        v2.a = v1.a;
        System.out.println(x);
        teste2(v1.a, x, v1);
        System.out.println(v1.getNome());
        System.out.println(x);
        System.out.println(v1.a);
        System.out.println(v1.b);
        System.out.println(v2.a);
        System.out.println(v2.b);
        teste3(v2);
        System.out.println(v2.getNome());
        System.out.println(v2.a);
    }
    public static void teste1(Arvore v2, Arvore v1) {
        v1.setNome("Abacateiro");
        v1 = v2;
        v1.setNome("Eucalipto");
    }
    public static void teste2(int a, int x, Arvore arv) {
        a = 20;
        x=15;
        arv.b = 99;
        arv.setNome("Macieira");
        arv = new Arvore();
    }
    public static Arvore teste3(Arvore v2) {
        v2 = new Arvore();
        v2.setNome("Mogno");
        return v2;
    }
}

```