Centro Federal de Educação Tecnológica

**Disciplina: Programação Orientada a Objetos**

Prof. Gustavo Guedes

**Aluno(a):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Data: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Códigos desnecessários e que reduzam o desempenho do sistema serão penalizados.**

**Utilize as boas práticas de programação, sempre que possível. Vale lembrar que você deve declarar atributos de instância como privados.**

**LEIA AS QUESTÕES ATÉ O FINAL ANTES DE COMEÇAR.**

**Questão 1 (3.0) – Desenvolva o código conforme pedido abaixo:**

A – Crie a interface Ligavel com apenas um método denominado ligável, que não possui argumentos e que não retorna nada.

B – 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 de diversos tipos, inclusive do tipo Produto. 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.

B2 – Faça com que, se uma referencia a um objeto do tipo Produto seja exibida no console (ex: System.out.println(p)), seja exibido o id, cor e preco do produto.

C- Crie 2 subclasses da classe Produto: Televisao e Tapete. Faça Televisao implementar Ligavel.

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

D2 – Ainda na classe principal, crie um método denominado transformaListEmMap (List k): Map<String, Produto> que receba uma lista **k** de objetos (de diversos tipos, incluindo produtos) e retorna um mapa de produtos, contendo apenas os objetos do tipo Produto na lista representada por **k**. Considere a chave do mapa como o id dos produtos. Utilize o Iterator nessa questão.

D3 – Dado o mapa retornado no item D2, crie um método (na classe principal) transformaMap (Map<String, Produto> m): Produto [] que receba um mapa e retorne um array contendo os objetos Produto presentes no mapa.

**Questão 2 (3.0) – Desenvolva o código conforme pedido abaixo:**

Questão 2-a – Crie uma classe denominada Filme. Essa classe possui três atributos de instância: id(int), titulo (String) e ano (int). Desenvolva sua classe de maneira que você consiga ordenar (em ordem alfabética por nome) os filmes utilizando o método de ordenação de Collections.

2-b - Considere a existência de um método denominado LocAntiga.getFilmes():String []. Esse método retorna um array de Strings no seguinte formato: ID#titulo#ano. Por exemplo, considere os elementos desse array como (123#Morro dos ventos uivantes#1939, 443#Morro dos ventos uivantes#1939, 443#A ilha#2005, etc...). Esses valores representam id, titulo e ano dos filmes. Crie um método chamado Utils3.processa(String [] filmes):Set que irá receber como argumento o retorno de LocAntiga.getFilmes(). Em seguida, o método Utils3.processa irá processar e retornar um conjunto com objetos do tipo Filme, em que cada objeto Filme representa um elemento do array retornado por LocAntiga.getFilmes(). PS: observe que podem existir filmes com nome e ano iguais, porém, o id é diferente. Atribua o id, titulo e ano dos filmes.

2-c – Crie uma classe principal. Utilizando os itens 3-a e 3-b, imprima a quantidade total de filmes existente na locadora. Ex: existe um total de 1000 filmes na locadora. Imprima também a quantidade existente de cada título. Ex: Morro dos ventos uivantes: 2; A ilha: 1; etc. Em seguida, ordene os filmes e exiba a lista de filmes ordenados.

**Questão 4 (2) –** Considere a classe Utils que possui um método estático chamado retornaConversoesPalavras() que retorna um Map. O Map retornado contém chaves com palavras reduzidas e valores com as palavras completas. Como exemplo, temos a chave “vc” associada com o valor “você”, temos a chave “tb” associada com o valor “também”. Crie um sistema que recebe do console uma String, transforme as palavras reduzidas nas palavras completas utilizando o Map e imprima no console o texto com as palavras completas. a. Exemplo i. Entrada: vc vai na casa dela tb? -> saída: você vai na casa dela também?

**Questão 4 (2) – Observe a questão abaixo. O que sai no console?**

**public** **class** Estrela {

**private** String nome;

**public** **int** p1;

**public** **static** **int** *p2*;

**public** String getNome() {

**return** nome;

}

**public** **void** setNome(String nome) {

**this**.nome = nome;

}

**public** Estrela() {

p1++;

*p2*++;

}

}

**public** **class** TestaEstrela {

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

Estrela v1 = **new** Estrela();

v1.setNome("Sirius");

Estrela v2 = **new** Estrela();

Estrela v3 = **new** Estrela();

v2.setNome("Rigel");

*teste1*(v1, v2);

System.***out***.println(v1.getNome());

System.***out***.println(v2.getNome());

**int** x = 81;

v1.p1=11;

v1.p1 = v2.p1;

System.***out***.println(x);

*teste2*(v1.p1, x, v1);

System.***out***.println(v1.getNome());

System.***out***.println(x);

System.***out***.println(v1.p1);

System.***out***.println(v1.*p2*);

*teste3*(v2);

System.***out***.println(v2.getNome());

System.***out***.println(v2.p1);

v3 = *teste3*(v2);

v3.setNome("Altair");

System.***out***.println(v2.getNome());

}

**public** **static** **void** teste1(Estrela v2, Estrela v1) {

v1.setNome("Sol");

v1 = v2;

v1.setNome("Antares");

v2=v1;

}

**public** **static** **void** teste2(**int** x, **int** a, Estrela arv) {

a = 12;

x=18;

arv.setNome("Canopus");

arv = **new** Estrela();

arv.p1=46;

arv.*p2*=33;

}

**public** **static** Estrela teste3(Estrela v2) {

v2 = **new** Estrela();

v2.setNome("Vega");

**return** v2;

}

}

**BOA SORTE!**