Centro Federal de Educação Tecnológica
Disciplina: POO
Prof. Gustavo Guedes

Nota

Aluno(a): _____

Turma:_____ **Data: _____**

Códigos desnecessários e que reduzam o desempenho do sistema serão penalizados.
Utilize as boas práticas de programação, sempre que possível. Vale lembrar que, quando possível, declarar atributos de instância como privados são uma boa prática de programação.
LEIA AS QUESTÕES ATÉ O FINAL ANTES DE COMEÇAR.

Questão 1 (4,8) – Desenvolva o código conforme pedido abaixo:

A – Escreva uma classe Aluno com 3 atributos privados: id (String), nome (String) e nota (double). Crie os getters e setters apenas se precisar. Crie em Aluno 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<Aluno> x, Aluno y). Escreva esse método de forma que seja verificada a existência do objeto Aluno representado por y na lista representada por x, retorne verdadeiro se existir e falso se não existir. Considere que dois objetos Aluno são iguais se possuem o **mesmo id**. Não é permitida qualquer iteração para realizar esse item, ou seja, não use *for*, *iterator*, etc. Implemente o que for necessário para o funcionamento.

B – Crie um método em Utils com a seguinte assinatura public static void ordena (List<Aluno> x). Esse método deve ordenar os objetos Aluno em x por ordem alfabética de nome. Prepare a classe Aluno para que isso ocorra corretamente. Não é permitida qualquer iteração para realizar esse item, ou seja, não use *for*, *iterator*, etc (Utilize o método apropriado de Collections).

C – Ao utilizar o System.out.println em um objeto Aluno, deve sair no console o id, nome e nota da Aluno.

D- Crie 2 subclasses da classe Aluno: AlunoRegular e AlunoEspecial.

E - Dada a classe Utils, crie o método public Map<String, Aluno> retornaDados(Set conjuntoAlunos), utilizando o ITERATOR nessa questão.

Considere que o conjunto recebido como argumento (conjuntoAlunos) contém Strings no seguinte formato: id#nome#nota#tipo. Por exemplo, considere os elementos desse conjunto como (154-3#Simba Silva#3.7#R, 553-2#Juju Juba#10.0#E, etc.). Esses valores representam id, nome, nota e tipo do Aluno (AlunoRegular ou AlunoEspecial). Dessa maneira, implemente o método *retornaDados* de forma que seja retornado um mapa da seguinte forma: os elementos de *conjuntoAlunos* devem ser percorridos, o id de cada elemento (ex: 154-3) é a chave do Mapa e os valores do mapa são objetos do tipo AlunoRegular ou AlunoEspecial. Resumindo, você irá criar um objeto AlunoRegular (se o último caractere da string for R) ou AlunoEspecial representando cada elemento em *conjuntoAlunos* (se o último caractere da string for E) e adicionar ao mapa. Caso o último caractere não seja R nem E, lance a exceção RuntimeException. Preencha também os nomes e notas dos objetos Pessoa.

F – Crie uma nova classe, denominada ProgramaPrincipal. O programa deverá receber do console 100 strings no seguinte formato: id#nome#nota#tipo (i.e., mesmo formato explicado no item E). Utilizando os métodos já criados, exiba no console os nomes e notas dos alunos com os nomes ordenados em ordem alfabética.

Questão 3 – (0.7) – Imagine uma classe X que tenha um método chamado fazerAlgo que não retorne nada, tenha visibilidade apenas dentro do mesmo pacote, que não tenha argumentos e que não possa ser sobrescrito. Escreva a assinatura desse método.

Questão 4 – (2.5) Dado um Map com chave do tipo String e valor do tipo Integer, percorra as chaves do mapa de forma que se a chave iniciar por A, os valores são somados. Dado o método:

```
public static double contaA(Map mapa) { //não mude a assinatura do método
//implemente
}
```

Se passarmos o mapa abaixo, o valor de saída do método seria 28.

Chave	Valor
A1	5
A2	6.3
A3	7
BD21	3
B2	7.6
AC74	2
A5	8
B3	3
. .	
. .	

Questão 4 – (2.0) Diga 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++;
	}
}
```

```java
public class TestaEstrela {
	public static void main(String[] args) {
		Estrela v1 = new Estrela();
		v1.setNome("Sol");
		Estrela v2 = new Estrela();
		Estrela v3 = new Estrela();
		v2.setNome("Antares");
		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("Sirius");
		v1 = v2;
		v1.setNome("Rigel");
		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;
	}
}
```