

## 1- 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.

## 2- 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
..	
..	