



**Orientações iniciais:**

- A prova é individual e sem consulta;
- A prova vale **100 pontos**, que equivalem à nota **10**;
- A prova tem duração de mínima de **30 minutos** e duração máxima de **3 horas**;
- Caso você tenha alguma dúvida referente ao enunciado das questões durante a prova, pergunte em voz alta. O professor não irá atendê-lo individualmente.
- Crie uma pasta com o seu nome na pasta Documentos e armazene todas as implementações da prova dentro dessa pasta. Você pode utilizar um editor de texto comum ou o Eclipse. Crie dentro dessa pasta um arquivo chamado *README.txt*, que contém um descritivo de como você organizou os exercícios (1 projeto no eclipse, vários projetos no eclipse, utilizou editor de texto comum, etc.).

**BOA PROVA!**



## Prova 3

### Linguagem de Programação Orientada a Objetos — 2017

1. (6.0 pontos) *Tonio Raumus Peludus* é dono de um importante frigorífico no estado de Mato Grosso do Sul e, após as fraudes reveladas pela operação "Carne Fraca" da Polícia Federal, decidiu estabelecer um controle de qualidade mais rigoroso das carnes do seu frigorífico. Para isso, Tonio precisará de sua ajuda.

Diariamente, o frigorífico de Tonio (que trabalha apenas com carne bovina) recebe várias peças/pedaços de carnes, e para cada peça as seguintes informações são armazenadas: região do boi (picanha, maminha, alcatra, coxão mole, etc.), peso (em kg), origem (cidade e estado) e nome da fazenda de origem.

As informações referentes às peças recebidas estão armazenadas em um arquivo, onde cada linha corresponde a informações de uma peça recebida. Cada linha do arquivo obedece o seguinte formato: *região,peso,cidade\_origem,estado\_origem,fazenda*.

O arquivo abaixo é um exemplo de arquivo de entrada.

```
Picanha,2,Campo Grande,MS,Fazenda do seu Zé do Papelão  
Coxão Mole,3,Curitiba,PR,Recanto do Acido Ascórbico  
Maminha,3,Brasília,DF,Rancho não há o que temer  
...
```

Dada uma entrada com este formato, Tonio deseja que você forneça a ele os seguintes relatórios:

- Relatório 1: listagem ordenada de todas as peças contendo todas as informações. O nome da cidade deve ser usado como critério de ordenação;
- Relatório 2: listar a quantidade de peças de cada região do boi e o peso total das peças. Por exemplo: maminha -> 2 peças, total 15 kg; picanha -> 4 peças, total 22kg; etc.
- Relatório 3: listar, para cada fazenda, quais as peças que foram fornecidas. Por exemplo: Fazenda do seu Zé do Papelão -> Picanha, Maminha ; Recanto do Acido Ascórbico -> Alcatra, Bananinha; etc.

Neste exercício você **DEVE** utilizar **PELO MENOS** as seguintes classes e interfaces Java: Collections, List, Map e Set.

Para a leitura de arquivo, você pode utilizar as seguintes classes e métodos:

```
BufferedReader bf =  
new BufferedReader(new FileReader(new File(nomeDoArquivo)));  
bf.ready(); // pergunta se há linhas no arquivo  
bf.readLine(); // lê uma linha
```



2. (4.0 pontos) Implemente uma classe chamada *VetoriusInfinitus*< *T* >, capaz de modelar um vetor "infinito" utilizando um vetor comum. Sua classe deve fornecer métodos para adicionar um elemento no final, adicionar um elemento em uma posição específica (essa operação deve sempre ser possível), acessar um elemento em uma determinada posição (caso essa posição não exista, lance uma exceção não verificada) e retornar a quantidade atual de elementos no vetor. Além disso, objetos do tipo *VetoriusInfinitus* só podem armazenar números.

Crie uma classe de teste que instancie um objeto do tipo *VetoriusInfinitus* e valide seus métodos.