



**Orientações iniciais:**

- A prova é individual e sem consulta;
- A prova vale **100 pontos**, que equivalem à nota **10**;
- Desligue todos os equipamentos eletrônicos (Smartphones, Tablets, Smartwatches, etc...)
- Consulta à Internet é proibida.
- Distribua bem o tempo gasto em cada questão. A quantidade de pontos de cada questão corresponde ao tempo que você deveria gastar em cada uma delas;
- A prova tem duração de mínima de **40 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.
- Perguntas do tipo: Qual o nome daquela classe que faz tal coisa? Meu programa não está compilando, o que estou errando?. Não serão respondidas.
- 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 2

### Linguagem de Programação Orientada a Objetos — 2017

1. [50 pontos] Nesta semana o Brasil presenciou mais uma sessão pública na Câmara de Deputados referente a uma investigação envolvendo o Presidente da República. Nesse exercício você criará um sistema orientado a objetos com o objetivo de ajudar nossos caros Deputados a terem uma visualização mais organizada dos dados das votações. O sistema recebe como entrada um arquivo com o seguinte formato:

NOME DA VOTAÇÃO, NOME DO DEPUTADO, PARTIDO, VOTO

As linhas abaixo apresentam um exemplo de arquivo de entrada.

```
CASSAÇÃO DO TEMER, TIRIRICA, PR, SIM
CASSAÇÃO DO TEMER, BOLSONARO, PSC, NÃO
CASSAÇÃO DO TEMER, ALAN RICK, PRB, NÃO
CRIAÇÃO DO DIA DO POLÍTICO, FLAVIANO MELO, PMDB, SIM
CRIAÇÃO DO DIA DO POLÍTICO, ANGELIM, PT, SIM
CRIAÇÃO DO DIA DO POLÍTICO, TIRIRICA, PR, SIM
CRIAÇÃO DO DIA DO POLÍTICO, CÉSAR MESSIAS, PSB, SIM
```

Supondo um arquivo de entrada com este formato, crie um programa Java orientado a objetos que gere os seguintes relatórios:

- Relatório 1: lista de todos votos (da maneira como está apresentada no arquivo de entrada), ordenada pelo nome do Deputado;
- Relatório 2: para cada candidato, exibir quantas sessões de votação ele participou. Por exemplo: TIRIRICA -> 2, BOLSONARO -> 1, etc.;
- Relatório 3: para cada votação, exibir quais os partidos que participaram da votação. Por exemplo: CASSAÇÃO DO TEMER -> {PR,PSC,PRB}; CRIAÇÃO DO DIA DO POLÍTICO -> {PMDB,PT,PR}.

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

Para a leitura do arquivo, você pode utilizar o seguinte código:

```
// Leitura do arquivo
BufferedReader bf =
new BufferedReader(new FileReader(new File(nomeDoArquivo)));
bf.ready(); // pergunta se há linhas no arquivo
bf.readLine(); // lê uma linha
// Descobrir as palavras de uma String
String s = "CASSAÇÃO DO TEMER, TIRIRICA, PR, SIM";
String[] palavras = s.split(", ");
```



2. [50 pontos] Um Set é uma coleção que não contém elementos duplicados. Mais formalmente, um *Set* não contém um par de elementos  $(e1, e2)$  tal que  $e1.equals(e2)$ .

Crie uma classe concreta MySet capaz de modelar um *Set* de um tipo específico utilizando um *vetor* (eu disse **VETOR**, e não **ArrayList**). Essa classe deve conter PELO MENOS os seguintes métodos:

- *void add(T elemento)*: adiciona um elemento específico a este *Set*, se ele ainda não está presente. Se o elemento já estiver presente, seu método deve lançar uma exceção não verificada chamada *ElementoDuplicadoException*;
- *void clear()*: remove todos os elementos;
- *boolean contains(Object o)*: retorna *true* se o *Set* contiver o objeto *o*;
- *void remove(Object o)*: remove o objeto *o* do *Set*. Se o objeto *o* não estiver presente, seu método deve lançar uma exceção verificada ElementoNaoEncontradoException, e armazenar o elemento nessa exceção;
- *int size()*: retorna o número de elementos válidos no *Set*.

Logo após, crie uma classe TesteMySet que fornece um menu capaz de testar todos os métodos implementados na classe MySet.