UFES - CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA

Programação III

GABRIEL FERRARI
MARCELA VIEIRA

Processamento de dados da Eleição 2016 de vereadores

Vitória
11 de novembro de 2016

1 Introdução

O sistema eleitoral brasileiro utiliza dois métodos de eleição de candidatos para posições de liderança no País: majoritária e proporcional.
O programa criado analisa os resultados das eleições exportados pelo *software* Divulga 2016, disponibilizado pela Justiça Eleitoral, e apresenta relaórios de diferentes resultados alternando-se os dois métodos de escolha de candidatos e comparando-os.
Todos os dados provém do arquivo `CSV` exportado e são processados pelo programa escrito em linguagem `Java`.

2 Objetivos

Gerar relatórios na saída padrão a partir da leitura e manipulação de um arquivo de divulgação, determinado na linha de comando, acerca dos resultados de diversos Estados das Eleições 2016 de vereadores.

3 Metodologia

O programa lê o arquivo de entrada, cria listas de vereadores, cada qual com informações de nome, posição na divulgação final, partido, número de votos, entre outras, e listas de partidos e coligações.
Essas listas são ordenadas por número de votos e o programa gera novos *rankings* para as Eleições de acordo com os diferentes métodos de votação. É possível saber os diferentes resultados para os candidatos, partidos e coligações.

3.1 Candidato.java

Abstração dos candidatos à vereador cujos atributos são retirados da leitura individual das linhas do documento de entrada, exceto por `eleito`, que é um atributo booleano que facilita a identificação de candidatos eleitos em outras métodos do programa.
`partido` e `coligação` referem-se a instâncias específicas de Partido e Coligação. Esse *link* permite maior coesão entre as classes e aplicam-se bem ao contexto, pois todos os candidatos pertencem a um partido, os quais se relacionam dentro de coligações.

3.2 Partido.java

Abstração dos partidos contidos nas informações de cada candidato. Um partido tem nome, a quantidade total de votos recebidos pelos candidatos que o compõem e a lista de candidatos pertencentes a ele.

3.3 Coligacao.java

Abstração de coligações (conjunto de partidos), também referenciadas nas informações de entrada dos candidatos. Uma coligação tem nome, quantidade total de votos recebidos pelos candidatos dos partidos que a formam e a lista de partidos da coligação.

3.4 Eleicao.java

Esta classe contém as listas de partidos, vereadores e coligações de uma eleição e métodos para criar os diferentes relatórios descritos na especificação do trabalho.
As primeiras funções são de criação, adição de objetos e outras comuns ao uso de listas. Na seção seguinte do arquivo, temos métodos específicos para cada relatório, que retornam os resultados para os métodos de impressão de `Leitor.java`.

3.5 Leitor.java

É a classe principal do programa, com as chamadas de leitura, criação e escrita dos resultados. Os métodos foram separados entre `Eleicao` e `Leitor` para que a classe principal contesse apenas a `main` e métodos estritamente necessários, garantindo a organização e clareza no entendimento do programa.

4 Resultados e Avaliação

O programa, como mencionado, recebe um arquivo `CSV`, faz a leitura e, para cada linha, cria um candidato com as informações contidas nessa linha e o adiciona à lista de candidatos. O método de leitura do candidato se encarrega de adicionar os partidos e coligações às suas respectivas listas.
Os relatórios gerados pelo programa são:

Número de Vagas

A contagem do número de vagas para vereador de uma cidade é realizada pela contagem de candidatos que apresentam um asterisco (*) antes do seu respectivo índice no arquivo de entrada. Os asteriscos representam os candidatos eleitos, logo, a quantidade de eleitos é igual a de vagas.
Durante a leitura de candidatos, o objeto recebe o valor `true` no atributo `eleito` caso apresente o asterisco. A função `getNumVagas` passa por toda a lista de candidatos contando os candidatos classificados como "eleito".

Candidatos Eleitos

A função `getEleitos` cria uma lista com todos os candidatos eleitos (segundo o mesmo método citado anteriormente) e a retorna para impressão. Os candidatos da lista são escritos conforme a formatação abaixo:

```
1 - FABRÍCIO GANDINI (PPS , 7611 votos) - Coligação:  PPS /
                                PROS
```

Candidatos mais votados dentro do número de vagas

Para apresentar os candidatos mais votados, é necessário apenas ordená-los por número decrescente de vagas. Para reordená-los com esse critério sem alterar a lista original, a função `getMaisVotados` cria uma *cópia* da lista original com o método `clone()` e reordena a lista copiada com o comparador por número de votos sobrescrito em `Candidato.java`.
O índice de cada candidato também é alterado segundo a nova ordem da lista.

Candidatos não eleitos e que seriam eleitos se a votação fosse majoritária

Este relatório é gerado pela função `getPrejudicados`, que clona a lista de candidatos e a ordena pelo número de votos (de forma decrescente). Desconsiderando os candidatos que estão nas N primeira posições (em que N é o número de vagas), já que esses foram eleitos, analisa os próximos candidatos da lista que tem valor `false` no atributo `eleito`.

Candidatos que não seriam eleitos se a eleição fosse majoritária

Com a lista clonada ordenada por número decrescente de votos, a função `getBeneficiados` retorna os candidatos eleitos a partir da posição N+1 (em que N é a quantidade de vagas).

Total de votos por coligação/partido e número de candidatos eleitos

Este relatório envolve, simplesmente, a impressão da lista de coligações ordenada por número decrescente de votos na sua criação. Dentro de `toString`, a função `getNEleitos` soma a quantidade de candidatos eleitos fazendo uma iteração sobre a lista de partidos que fazem parte da coligação.

Total de votos por partido e número de candidatos eleitos

Funciona de forma análoga ao relatório anterior, mas utilizando a lista de partidos.

Total de votos nominais

A função `totalVotosNominais` percorre a lista de partidos somando os valores no atributo "votos"de cada partido e retorna o total.

Os testes realizados com os arquivos disponibilizados para três cidades mostram os resultados esperados.