

Universidade Federal do Espírito Santo
Departamento de Informática
Curso de Engenharia da Computação

Trabalho I - Sistema Eleitoral Brasileiro

André Luiz Piona de Araujo

Vitória-ES

Novembro de 2016

1 INTRODUÇÃO

O objetivo do trabalho é fazer um programa em linguagem Java utilizando o que foi ensinado em sala de aula.

O programa tem que ler qualquer lista de vereadores dos que estão disponibilizados para download no site do governo, e conseguir separar esses vereadores e organizar as suas informações, além de conseguir também organizar em partidos e coligações. O programa deverá imprimir listas de vereadores de maneiras específicas, e uma lista com as informações para todos os partidos e para as coligações.

2 DIFICULDADES ENCONTRADAS

- Desorganização ao começar a desenvolver, pois estava escrevendo os métodos do programa praticamente apenas na classe “Main”, dificultando a compreensão cada vez que avançava mais.
- Conseguir criar tipo “Vereador” com referência para seu partido, e tipo “Partido” com referência para a coligação, ao invés de String com o nome dos mesmos.
- Criar coligações de apenas um partido para conseguir sair os partidos sem coligação no método de imprimir listas de coligações, porém não imprimir essas coligações quando a lista de vereadores está sendo impresso.

3 SOLUÇÕES ENCONTRADAS

- Para o problema de desorganização foi necessário criar diferentes classes com os métodos necessários para o programa, e deixar a classe “Main” apenas para ler o arquivo (foi usado ‘nextLine’), e executar os métodos das outras classes. As classes criadas foram “Leitor”, que recebe a linha lida e divide em variáveis necessárias para o desenvolvimento do programa, a classe “EleicaoDeVereadores” para utilizar essas variáveis do “Leitor” para criar e listar os vereadores, partidos e coligações, e por último a classe “Saída” , que tem como objetivo organizar e imprimir as listas de forma necessária.
- O problema de referências foi resolvido criando uma ‘corrente’ de métodos. Ao utilizar o método que cria um novo “Vereador”, ele chamava o método que retorna o “Partido”, e dentro dele um outro método que retorna “Coligação”. Esses métodos procuram o partido ou coligação na lista, e retorna ou cria os mesmos, para serem usados como referência pelo método anterior (retorna “Coligação” para usar como referência para “Partido”, e “Partido” para usar em “Vereador”)
- No problema para criar coligações com um partido, a solução foi bem simples. Tanto o “Vereador” quanto “Partido” recebem a variável “coligacao” como ‘null’, porém ao terminar de ler o arquivo de leitura, um método chamado “adicionaColigacaoDeUmPartido” cria coligações usando as mesmas informações dos partidos sem coligação, e adiciona a lista de coligações.

4 TESTES

Os testes foram feitos de acordo com a classe. Para a classe “Leitor”, o objetivo era conseguir imprimir a mesma linha lida usando as variáveis adquiridas separando a linha.

Para a classe “EleicaoDeVereadores”, o objetivo do teste era conseguir imprimir a lista de vereadores separados pelo partido, e a lista de partidos separados por coligações, para assim confirmar que as referências estavam corretas. Além de imprimir também informações completas de cada um deles, para confirmar que estava correto.

E para a classe “Saida”, era conseguir imprimir as listas de forma que cumpria o objetivo do programa.

5 CONCLUSÃO

Antes de começar o programa, o principal problema aparente era conseguir dividir as informações de um mesmo vereador no arquivo de entrada. Foi a parte que deu bastante trabalho em C, porém no Java isso foi bem simples usando o método 'split'. Com as dúvidas que surgiam foi possível ver o vasto número de ferramentas que o Java possui. Inclusive o "Comparator" que poderia ter sido usado de forma simples no programa para ordenar as listas, se descoberto mais cedo.

Depois do programa finalizado, a parte que pode ser considerado mais difícil foi de desenvolver um modo de criar o "Vereador", "Partido" e "Coligacao" ao mesmo tempo, para assim poder utilizar referências. Infelizmente no começo do desenvolvimento do programa foi utilizado Strings ao invés de referências, e isso fez necessário modificar uma boa parte do mesmo para corrigir isso. Porém, graças a isso, ao refazer uma parte do programa foi possível melhorar a lógica e a organização do código.