



TEMA: ELEIÇÕES DE PREFEITOS

Este ano é especial porque é ano de eleições políticas no país e você foi contratado para elaborar um programa orientado a objeto, que respeite todas as propriedades fundamentais deste paradigma de programação. Por meio do seu programa será possível acompanhar informações consolidadas e importantes sobre os candidatos a prefeito de algumas cidades brasileiras.

O programa deverá primeiro registrar o nome de uma cidade brasileira e armazenar este nome em uma *string* mutável totalmente em maiúsculo para em seguida efetuar o cadastro de todos os candidatos a prefeito dessa cidade solicitando:

- **nome completo** do candidato que nunca poderá ser vazio, tendo sempre mais que 3 caracteres, e deverá preservar o formato como o usuário do programa registrou entre maiúsculas e minúsculas (só mostrará tudo em maiúsculo sem alterar como o usuário escreveu);
- **sigla do partido** ao qual o candidato está associado nessa eleição, sendo exigido pelo menos 2 caracteres e no máximo 10, mas não podendo existir qualquer pontuação ou espaço em branco na sigla, além de não ser repetido com outros candidatos que concorrerão nesta mesma eleição e cidade (sigla será única);
- **número da legenda** associado ao candidato para votação, sendo este número um valor inteiro com duas casas que poderá variar de maior que 10 até 99, devendo também ser único nesta cidade e eleição.

É importante esclarecer que não podem ser repetidas as siglas dos partidos de cada candidato e nem o número da legenda, pois cada um destes atributos possui valores únicos que não se repetem em uma eleição de uma mesma cidade. Não existe limite máximo para o cadastro de candidatos, mas sempre que o programa for executado ele exigirá o cadastro do nome de uma cidade e ao menos um candidato, caso contrário não poderá ser encerrada a execução do programa.

Análise este problema e elabore um programa orientado a objeto em ambiente gráfico que cadastre os dados dos candidatos a prefeito de uma cidade em específico, sendo validados todos os dados de entrada conforme as definições e restrições esclarecidas anteriormente. O valor 10 como limite para o número da legenda deverá estar definido na constante **MÍNIMO** que será sempre usada para verificação de um número de legenda válido, sendo maior que esta constante;

Sua solução deverá respeitar todas as propriedades já estudadas de orientação a objeto. Todos os dados informados neste enunciado são obrigatórios e serão apresentados em forma de tabela somente quando o usuário informar que terminou os cadastros de candidatos para aquela cidade específica. No entanto, essa forma de tabela deverá ser apresentada também usando um recurso gráfico de Java chamado **JTable** e não na console. O aspecto relacionado a pesquisa dessa primeira parte do Trabalho Final acontecerá sobre o componente **JTable** que deverá ser usado, obrigatoriamente, na apresentação dessa forma



de resultado tabelar para mostrar todos os candidatos cadastrados, estando no alto dessa tabela o nome da cidade. Todas as *string* apresentadas no **JTable** deverão estar completamente em maiúsculo.

Sua solução deverá oferecer para o usuário uma forma de menu de opções ou botões que propiciará ao usuário final escolher entre as opções relacionadas a seguir, estando todas elas sempre disponíveis para o usuário acionar a que desejar, até que seja escolhida a opção **Sair**:

- a) **Cadastrar** – conforme desejo do usuário um novo Candidato será cadastrado no programa, respeitando todas as validações de entrada de dados coerentes (indicadas anteriormente neste enunciado);
- b) **Listar** - todas os candidatos cadastrados de maneira tabelar no **JTable**, respeitando o formato tabelar com uma única linha de cabeçalho indicando todos os dados fornecidos corretamente nesta forma tabelar (relatório com todos os cadastros e todos os dados cadastrados);
- c) **Consultar** - uma pessoa específica por meio de seu número da legenda válido, em que todos os seus dados serão apresentados em uma **janela de diálogo** sem nenhum ícone (figura do tipo de mensagem), estando cada dado cadastral em uma linha desta janela por extenso. O título dessa janela de diálogo deverá ser o nome da cidade;
- d) **Pesquisar** - entre todas os candidatos, por meio do fornecimento de qualquer parte de seu nome, onde todos os registros que satisfizerem este nome serão recuperados e apresentados em ordem alfabética, de maneira tabelar aproveitando o mesmo **JTable** anterior (item **b**). Sua solução também deverá notificar ao usuário desta pesquisa a quantidade de registros que foram recuperados e mostrados no final dessa apresentação (na mesma janela **JTable**);
- e) **Sair** – encerrar o programa.

Diante das características do problema apresentado deverá ser elaborada a especificação da solução que será desenvolvida, individualmente, por cada estudante, respeitando, exatamente, o modelo definido por seu professor, além do código que implementará tal solução. Esta solução só será aceita se for desenvolvida no ambiente gráfico Java (**Swing**).

A solução deste projeto deverá estar documentada corretamente por meio dos **diagramas de Estudo de Caso e de Classes** que correspondam a implementação realizada, não podendo ser elaborada a leitura de dados através de Janelas de Diálogo (deverá existir uma janela **Swing** para o programa, correspondendo a console da aplicação).

Usando uma **JOptionPane** antes de terminar o programa sua solução deverá perguntar para o usuário final se ele deseja cadastrar outra cidade e seus respectivos candidatos. Se ele informar que **NÃO** o programa encerrará, mas se for **SIM** todo o processamento de sua solução deverá ser realizada de novo, ou seja, registrar o nome de uma cidade e os candidatos para prefeito dessa nova cidade (todos os dados anteriores de cidade e candidatos não poderão mais ser considerados/apresentados).

Apure todas as possibilidades de exceção em sua solução e as trate, pois o programa não poderá ser interrompido em nenhum momento até que o usuário final decida encerrá-lo.



Outro dado importante é que nunca se saberá inicialmente quantos candidatos estarão participando de cada eleição cadastrada.

Os nomes cadastrados nesta solução deverão ser apresentados sempre em maiúsculo e sem espaços em branco excessivos (antes ou depois do nome informado). É importante destacar que cada candidato NÃO poderá ser instanciado sem argumentos em seu método construtor. Assim, analise o problema proposto e elabore uma solução adequada a todo conteúdo estudado até o momento em POO, implementando serviços e recursos coerentes com uma solução adequada a este paradigma de programação, além de verificar sempre a entrada de dados para evitar o registro incorreto dos dados exigidos neste enunciado, com informações incoerentes, causando interrupção imediata e incorreta em sua execução. Observe com atenção quantas classes e pacotes são necessários para implementação adequada que está sendo exigida.

Observações sobre o Material a ser Entregue para Avaliação

- Este trabalho é individual e cada estudante deverá entregar um programa orientado a objeto em Java, diretamente no ambiente MOODLE da disciplina, assim como a especificação (documentação) deste projeto ([ver arquivo modelo disponível na Área de Compartilhamento](#));
- A entrega de projeto plagiado, parcial ou totalmente copiado, resultará na nota zero para todos os envolvidos, independente de qualquer justificativa. Ressalto que o Plano de Ensino da disciplina define que para aprovação é necessária obter a nota mínima de 4,0 (três pontos) neste Trabalho Final, onde a cópia já o reprova diretamente, independente de outras notas;
- Sua implementação deverá ser feita somente com **Swing**, ao menos nos itens em que o enunciado exija outra forma de interação e apresentação para o usuário (indicar explicitamente que poderá mostrar na **CONSOLE**);
- O nome do projeto a ser entregue para avaliação deverá possuir a expressão **TF1** seguida do primeiro e último nome do estudante, sem espaço em branco ou acento, além de sua matrícula. Por exemplo: a entrega pela aluna Ana Maria Braga, com matrícula 139874563, teria o nome do projeto **TF1AnaBraga**, enquanto o arquivo físico e compactado, somente no **formato ZIP**, seria **TF1AnaBraga_139874563.zip**;
- Atente a organização do código que será entregue e deverá estar estruturado de maneira eficiente com vários pacotes que sejam coerentes com sua implementação, além de respeitar o padrão exigido pela disciplina durante todo este semestre letivo;
- O arquivo de documentação deverá estar no projeto Java, na pasta **modelagem** que você criará através do Eclipse, com o mesmo nome do projeto, ou seja, **TF1AnaBraga.pdf** para o exemplo usado anteriormente.