Projeto 2

Nesse projeto trabalharemos os conceitos referentes a listas encadeadas e orientação a objetos. Para isso, implementaremos um programa para processar os dados dos candidatos e seus bens declarados à Justiça Eleitoral para o pleito de 2014.

Os dados a serem trabalhados são:

1. Candidatos:

http://agencia.tse.jus.br/estatistica/sead/odsele/consulta_cand/consulta_cand_2014 .zip 2. Bens:

http://agencia.tse.jus.br/estatistica/sead/odsele/bem_candidato/bem_candidato_2014 .zip

Os dados disponibilizados pelo TSE estão separados por UF. No entanto, eles deverão ser processados e armazenados como se estivessem em um único arquivo. A descrição das tabelas (dados) estão disponíveis no arquivo zip, especificamente no arquivo LEIAME.

Você deve processar os dados, criando representações computacionais conforme a descrição a seguir. Seu programa deverá implementar uma série de funcionalidades para permitir a análise dos dados, conforme item 4.

1 - Classe Candidato:

A classe Candidato deverá conter:

- Ano da eleição
- Sigla da UF
- Código do Cargo
- Descrição do cargo
- Nome do candidato
- ID do candidato (número sequencial do candidato gerado pelos sistemas eleitorais)
- Número na urna
- CPF
- Nome na urna
- Número do partido
- Nome do partido
- Sigla do partido
- Código de ocupação do candidato
- Descrição da ocupação
- Data de nascimento (armazenada como dia, mês e ano)
- Sexo do candidato
- Grau de instrução
- Estado civil

- UF nascimento
- Nome do município de nascimento
- Situação do candidato pós pleito (eleito, não eleito, suplente)
- Situação da candidatura (deferida ou indeferida)
- Lista de bens (objetos a serem detalhados a seguir)

Os atributos da classe devem ser obrigatoriamente privados. Dessa forma, deverão ser implementados métodos para acessar (alterar e obter) os valores desses atributos. Além disso, o objeto *Candidato* deverá possuir um método *incluirBem* que deve receber como parâmetro um objeto do tipo *Bem* (descrito a seguir) e deve inseri-lo na lista de bens do candidato (você deverá implementar a lista conforme descrição a seguir). Você também deve implementar os métodos __str__ e __repr__ de forma que a saída seja formatada conforme a seguir:

Nome da Urna -- Número da Urna -- Sigla do partido
Cargo disputado (UF) Município Nascimento (UF)
Resumo dos bens:

- Total declarado: (valor formatado como R$)
- Total por tipo de bem (imprimir Tipo: valor formatado em R$)

Você deve também implementar os métodos de comparação entre candidatos. A comparação deve ser feita com base no nome completo do candidato. Dois candidatos serão iguais somente se tiverem o mesmo nome e o mesmo CPF.

Finalmente, você deve incluir um método para exibir os bens declarados do candidato.

2 - Classe Bem

Os bens declarados de cada candidato deverão ser armazenados em objetos do tipo Bem. Esses objetos devem conter os seguintes atributos:

- Código do tipo de bem
- Descrição do tipo de bem
- Descrição detalhada do bem
- Valor do bem

Os atributos deverão ser todos privados. Dessa forma, você deverá implementar métodos de acesso aos valores armazenados. Você também deve implementar os métodos __str__ e __repr__ para exibir as informações do bem conforme abaixo:

Código -- Descrição do tipo -- Valor (formatado em R$) Descrição: Descrição detalhada
(A palavra descrição deve ser impressa e o texto da descrição deve ser formatado de maneira a não exceder 80 caracteres por linha; ver TextWrap)
Você também deve implementar métodos de comparação baseados no valor do bem. Os critérios de desempate são: primeiro o código do bem; segundo a descrição detalhada do bem. Dois bens são iguais se possuírem o mesmo valor e a mesma descrição detalhada.

3 - Classe Lista

Você deve implementar uma lista duplamente encadeada (com referências para o anterior e próximo) contendo todas as funcionalidades discutidas em sala. Além disso, você deve implementar um segundo método para inserir elementos de forma ordenada conforme uma função de comparação. Essa função de comparação deve possuir dois parâmetros (os objetos a serem comparados) e deve retornar -1, 0 ou 1 caso o primeiro objeto seja menor, igual, ou maior que o segundo objeto, respectivamente. Fora da classe lista, você deve implementar funções de comparação para os candidatos baseadas nos seguintes critérios:

- Alfabética pelo nome em ordem crescente
- Alfabética pelo nome em ordem decrescente
- Pelo total dos bens (crescente e decrescente)
- Pelo partido e nome (alfabética, crescente e decrescente)
- Pela data de nascimento (crescente e decrescente)

Além dessas funcionalidades, você deverá implementar os métodos para exibição do conteúdo na tela.

4 - Classe Controle

A classe Controle é a classe principal do projeto. Ela deve ter métodos para execução das funcionalidades listadas a seguir.

1. Carregamento dos candidatos a partir do arquivo (o caminho do arquivo deve ser um parâmetro do método). Os candidatos lidos do arquivo deverão ser armazenados em objetos do tipo Candidato e inseridos numa lista de candidatos;

2. Carregamento dos bens dos candidatos. Os bens deverão ser lidos do arquivo cujo caminho é passado como parâmetro para o método. Cada bem lido deverá ser armazenado em um objeto do tipo Bem e inserido na lista de bens de seu respectivo candidato. A correspondência deve ser feita pelo ID do candidato;

3. Deve-se implementar funções/métodos que recuperem a lista de candidatos de um determinado partido, UF, nascidos em um dado município, candidatos a um determinado cargo, cujo total de bens declarados esteja acima de um valor especificado pelo usuário, ou que tenham ou não sido eleitos no pleito;

4. Deve-se implementar uma função/método que permita exibir a lista obtida em 3

5. Crie funções que mostrem a média do total de bens dos candidatos por cargo, UF, partido, ocupação, ou ano de nascimento

6. Crie uma função que remova da lista todos os candidatos que satisfaçam um critério. Exemplo, remover todos os candidatos com a candidatura indeferida, ou que não tenham sido eleitos.

Obs.: Os arquivos com os dados pessoais do candidato e de seus bens estão no formato CSV. O formato CSV organiza os dados em linhas separados geralmente por vírgulas. Entretanto, como no Brasil a vírgula é o separador decimal dos números, é comum que os dados organizados nesse formato estejam separados por ponto-e-vírgula. Em geral, a primeira linha do arquivo contém um cabeçalho que descreve as respectivas colunas. Mas existem casos em que os rótulos das colunas estão descritos em um dicionário de dados anexo ao arquivo. Este é o caso dos arquivos do projeto.

Para a leitura dos dados, você deverá processar o arquivo sem o auxílio de bibliotecas específicas. Em outras palavras, você deve abrir o arquivo como de costume, e processar linha a linha, extraindo os campos mencionados acima para cada objeto.

5 - O que entregar?

Você deve entregar todos os arquivos com os códigos implementados para o projeto. **OS ARQUIVOS NÃO PODEM ESTAR COMPRIMIDOS DE FORMA ALGUMA**. Projetos entregues em arquivos comprimidos, particularmente no formato rar, serão penalizados.

Você também deve entregar um arquivo com o tempo de execução e saída para cada uma das funcionalidades listadas no item 4. Esse arquivo também deve ser um arquivo texto.

6 - Quando entregar?

A data limite para entrega é **25/11/19 às 23h59**.

7 - Considerações finais

Projetos copiados em parte ou integralmente de colegas e/ou sites da internet serão punidos com a perda total dos pontos. **Em caso de cópia entre colegas, todos os envolvidos serão punidos com a perda dos pontos**.