



***Centro Federal de Educação
Tecnológica de Minas Gerais***

CURSO: ENGENHARIA DE COMPUTAÇÃO
DISCIPLINA: Programação de Computadores II
Professora: Natália

Trabalho Prático 2
Povo Feliz

Gustavo Henrique Marques
Pedro Henrique Lopes

Belo Horizonte - MG
16/06/2016

Convenções de código utilizadas:

- Nomenclatura:
 - Pacotes:
 - Composto por letras maiúsculas e minúsculas. Iniciado com letra minúscula.
 - Classes:
 - Substantivo composto por letras maiúscula e minúsculas. Iniciado com letra minúscula.
 - Métodos:
 - Verbo composto por letras maiúscula e minúsculas. Iniciado com letra minúscula.
 - Variáveis:
 - Composto por letras maiúscula e minúsculas. Iniciado com um sufixo de 3 letras minúsculas que identificam o tipo da variável. Ex: strNome.
 - Constantes:
 - Composto somente por letras maiúsculas separadas, quando necessário, pelo caracter “_”.
- Organização de código:
 - Identação:
 - TAB - 4 espaços.
 - Comentários:
 - Classes:
 - Iniciadas com um bloco de comentário explicando brevemente qual o objetivo e as funcionalidades da classe.
 - Métodos:
 - Padrão JavaDoc.

Descrição das principais funcionalidades:

- Checar se algum parâmetro que deveria ser ÚNICO é realmente: conjunto de métodos que conferem se um atributo é unico. Sua funcionalidade consiste em repetir a entrada enquanto o valor não for único

```

Saída - PovoFeliz (run) %
*****
*1) Cadastro *
*2) Excluir *
*3) Listar *
*4) Pesquisar *
*5) Alterar *
*6) Voltar *
*****
Escolha uma opção: 1
*****
*          Cadastro de Partido          *
*****

Digite a Sigla do Partido: PT

Digite o nome do Partido:

```

Saída do programa no caso descrito.

- Testes no Dígito dos candidatos: a função testa se o código é unico e se contém 2 casas. Caso falso ele continua solicitando até atender as condições.

```

Saída - PovoFeliz (run) %
*2) Vereador *
*****
Escolha uma opção: 1
*****
*          Cadastro de Candidato          *
*****

Digite o Código do Candidato: 123

Código inválido!

Digite o Código do Candidato: 13
Já existe um prefeito com esse código!
Digite o Código do Candidato: 66

Digite o nome do Candidato:

```

Saída do programa no caso acima

- Funcionamento do método de votação: o método votação é o responsável por controlar as condições necessarias nesse processo.
 - Entrada do Número de título de eleitor: caso o número de entrada seja inválido(Não existe no sistema) o progama acusa um erro e pergunta se o

usuário deseja encerrar o processo de votação, ou seja, ninguém mais votará. Digite 's' para encerrar a votação.

```
Saída x
PovoFeliz (run) x PovoFeliz (run) #2 x
*1) Inicar votação *
*2) Listar Urnas *
*3) Voltar *
*****
Escolha uma opção: 1
*****
*          Processo Votação          *
*****

Entre com titulo Eleitor: 9

Titulo não existe no sistema

Deseja encerrar votação? |
```

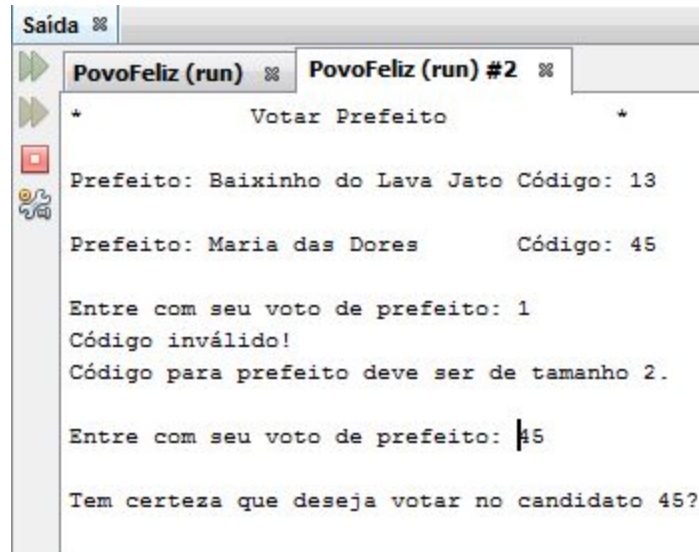
Saída do programa no caso acima

- Não existe presidente seção: Caso não exista presidente na seção e zona do eleitor que foi selecionado através do título: esse primeiro eleitor dessa zona e seção será o presidente da urna. Depois de definir esse primeiro eleitor como presidente ele pergunta horário de início da eleição.

```
Saída x
PovoFeliz (run) x PovoFeliz (run) #2 x
*****
Entre com titulo Eleitor: 9
Titulo não existe no sistema
Deseja encerrar votação? n
Entre com titulo Eleitor: 1
Esta zona e seção ainda nao possui presidente,
portanto este eleitor sera o presidente da seção
Entre com a Data e horario do inicio da seção <DD/MM/AAAA HH:MM> : |
```

Saída do programa no caso acima

- Votar Prefeito: Nessa parte primeiramente mostramos quais são os prefeitos disponíveis para voto (tirando os excedentes caso seja segundo turno). Caso voto não tenha 2 caracteres ele fica em loop solicitando o voto. Caso tenha 2 caracteres, o programa pergunta se o usuário tem certeza do voto. Caso não tenha certeza ele recomeça votação de prefeito. Caso tenha certeza contabiliza o voto.



```

Saída
PovoFeliz (run)  PovoFeliz (run) #2
*               Votar Prefeito               *
Prefeito: Baixinho do Lava Jato Código: 13
Prefeito: Maria das Dores      Código: 45

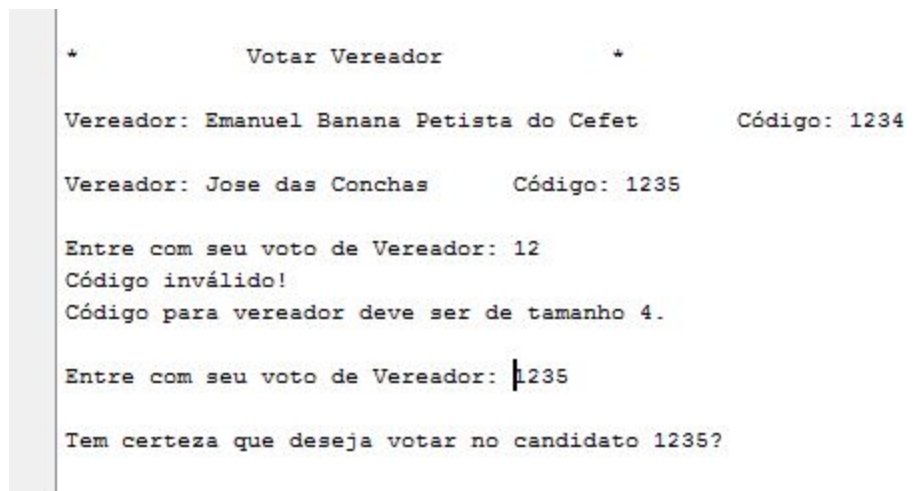
Entre com seu voto de prefeito: 1
Código inválido!
Código para prefeito deve ser de tamanho 2.

Entre com seu voto de prefeito: 45

Tem certeza que deseja votar no candidato 45?
  
```

Saída do programa no caso acima

- Votar Vereador: Caso seja primeiro turno: primeiramente mostramos quais são os vereadores disponíveis para voto. Caso voto não tenha 4 caracteres ele fica em loop solicitando o voto. Caso tenha 4 caracteres, o programa pergunta se o usuário tem certeza do voto. Caso não tenha certeza ele recomeça votação de Vereador. Caso tenha certeza contabiliza o voto.



```

*               Votar Vereador               *

Vereador: Emanuel Banana Petista do Cefet      Código: 1234
Vereador: Jose das Conchas      Código: 1235

Entre com seu voto de Vereador: 12
Código inválido!
Código para vereador deve ser de tamanho 4.

Entre com seu voto de Vereador: 1235

Tem certeza que deseja votar no candidato 1235?
  
```

Saída do programa no caso acima

- Encerrar Votação: Caso você responda 's' para pergunta "Deseja encerrar processo votação?", o programa solicitara a data do fim da eleição. Essa data será colocada em todas Urnas, além de imprimir o relatório.

```

Entre com a Data e horario do fim da seção <DD/MM/AAAA HH:MM> : 1/1/1 9:00

Resultado Vereador:

Nome: Emanuel Banana Petista do Cefet          50,00%          (1)
Nome: Jose das Conchas          50,00%          (1)

Votos Nulos:
Quantidade: 0   Porcetagem: 0,00%

Votos Branco:
Quantidade: 0   Porcetagem: 0,00%

Total de Votos: 2

Resultado Prefeitos

Nome: Baixinho do Lava Jato          0,00%          (0)
Nome: Maria das Dores          100,00%          (2)

Votos Nulos:
Quantidade: 0   Porcetagem: 0,00%

Votos Branco:
Quantidade: 0   Porcetagem: 0,00%

Total de Votos: 2

```

Saída do Programa no caso acima

- Escolher Relatório e mostrar: função que mostra opções de relatório e solicita um número ao usuário, caso seja número válido ele mostra o Relatório.
 - Relatório Prefeito:

```

Escolha uma opção: 4
*****
*                      Relatórios                      *
*****
*1) Procurar Relatório Prefeito          *
*2) Procurar Relatório Vereador        *
*3) Voltar                             *
*****
Escolha uma opção: 1
0: Turno: 1      Data Geração: Tue Jun 14 15:38:17 BRT 2016
Entre com o numero do relatorio que deseja ver:1
Entre com o numero do relatorio que deseja ver:0

Resultado Prefeitos

Nome: Baixinho do Lava Jato              0,00%          (0)

Nome: Maria das Dores                    100,00%         (2)

Votos Nulos:
Quantidade: 0   Porcetagem: 0,00%

Votos Branco:
Quantidade: 0   Porcetagem: 0,00%

Total de Votos: 2

```

Saída Relatório Prefeito

```

*****
*                      Relatórios                      *
*****
*1) Procurar Relatório Prefeito          *
*2) Procurar Relatório Vereador        *
*3) Voltar                             *
*****
Escolha uma opção: 2
0: Data Geração: Tue Jun 14 15:38:17 BRT 2016
Entre com o numero do relatorio que deseja ver:1
Entre com o numero do relatorio que deseja ver:0

Resultado Vereador:

Nome: Emanuel Banana Petista do Cefet    50,00%          (1)

Nome: Jose das Conchas                   50,00%          (1)

Votos Nulos:
Quantidade: 0   Porcetagem: 0,00%

Votos Branco:
Quantidade: 0   Porcetagem: 0,00%

Total de Votos: 2

```

Saída Relatório Vereador

Conceitos:

- **Pacotes:** O programa possui cinco pacotes.
 - principal: Contém a classe principal.
 - menu: responsável por realizar a gestão da estrutura de menus do programa.
 - moduloAdministrativo: responsável por realizar a gestão de candidatos, eleitores e relatório.
 - moduloUrnaEletronica: responsável por realizar a gestão da urna eletrônica.
 - moduloEleicao: contém todos os objetos necessários para o ato de votação e resultados.
- **Classe/Objeto/Método/Atributos:**
 - Classe: Candidato:
 - Características: Abstrata.
 - Atributos:
 - intCodigo
 - Tipo: Int;
 - Descrição: Código único do candidato para votação.
 - intVotos
 - Tipo: Int;
 - Descrição: quantidade de votos que esse candidato recebeu.
 - strEmail
 - Tipo: String;
 - Descrição: Endereço de email do candidato.
 - prtPartido
 - Tipo: Partido
 - Descrição: Representa o partido o qual o candidato é filiado.
 - Métodos:
 - receberVoto(): Soma um ao atributo votos.
 - toString(): Retorna uma string contendo o nome do candidato;
 - zerarVoto(): zera os votos.
 - Classe: Eleitor:
 - Características: Subclasse de Pessoa.
 - Atributos:
 - blnVotou:

- Tipo: Boolean;
 - Descrição: Define true se o eleitor votou e false caso não.
- intTitulo:
 - Tipo: Inteiro;
 - Descrição: Número do título de eleitor.
- intZona:
 - Tipo: Inteiro;
 - Descrição: Número da zona eleitoral.
- intSeccao:
 - Tipo: Inteiro;
 - Descrição: Número da secção.
- Métodos:
 - Votar: Modifica atributo blnVotou para true.
 - toString: Retorna String que contem número do título e o Nome do eleitor.
- Classe: Partido:
 - Características: Classe Concreta.
 - Atributos:
 - strNome:
 - Tipo: String;
 - Descrição: Representa nome do partido.
 - strSigla:
 - Tipo: String;
 - Descrição: Representa a sigla que o partido adota.
Ex: PT.
 - intCodigo:
 - Tipo: Inteiro;
 - Descrição: Código do partido.
 - Métodos:
 - toString: retorna uma string com a Sigla e o Codigo do partido.
- Classe: Pessoa:
 - Características: Superclasse.
 - Atributos:
 - strNome:
 - Tipo: String;
 - Descrição: Nome da Pessoa em questão.
 - dteDataNascimento:

- Tipo: Date;
 - Descrição: Data de nascimento da Pessoa.
- Métodos:
 - toString: retorna string com a Data de Nascimento e Nome da Pessoa.
 - converterDateToString(Date dtData): Converte um parametro do tipo Date em String(DD/MM/AAAA).
- Classe: Prefeito:
 - Características: Subclasse de Candidato e implementa Interface Politico.
 - Atributos:
 - vcpVice:
 - Tipo: Vice-Prefeito
 - Descrição: armazena dados de um vice prefeito já cadastrado.
 - blnEstaNoSegundoTurno
 - Tipo: boolean
 - Descrição: armazena true caso o candidato esteja no segundo turno.
 - Métodos:
 - verificarCodigo: faz os testes necessarios para verificar se o dígito para voto está de acordo com a especificação.
 - toString: retorna uma String contendo Prefeito, Código e Vice.
- Classe: Vereador:
 - Características: Subclasse de Candidato e implementa Interface Politico.
 - Métodos:
 - verificarCodigo: faz os testes necessarios para verificar se o dígito para voto está de acordo com a especificação.
 - toString: retorna uma String contendo o nome do Vereador e o código.
- Classe: VicePrefeito:
 - Características: Subclasse de Candidato.
- Classe: Urna:
 - Caraterísticas: Classe concreta.
 - Atributos:
 - IstPessoa: lista de pessoas

- blnTodosVotaram: Armazena true caso todos os eleitores tenham votado.
 - eltPresidente: Armazena o presidente.
- Métodos:
 - isBlnTodosVotoram: verifica se todos os eleitores votam
 - toString: retorna uma string contendo informações da urna
- Classe: CodigoInvalidoException
 - Características: Subclasse de Exception
- Classe: Relatório
 - Características: classe concreta;
 - Atributos:
 - intQtdVotosTotal:
 - Tipo: int
 - Descrição: armazena a quantidade total de votos
 - intQtdVotoBranco:
 - Tipo: int
 - Descrição: armazena a quantidade de votos em branco
 - intQtdVotoNulo:
 - Tipo: int
 - Descrição: armazena a quantidade de votos nulos
 - dteDataCriacao
 - Tipo: Date
 - Descrição: data de criação
 - Métodos:
 - mostrarRelatorioVotosNulosBranco: Exibe o relatório.
- Classe: RelatorioPrefeito
 - Características: subclasse de Relatorio
 - Atributos:
 - IstPrefeito
 - Tipo: ArrayList
 - Descrição: lista de prefeitos
 - intTurno
 - Tipo: int
 - Descrição: armazena o turno da eleição
 - Métodos:
 - montarRelatorio: monta o relatório
 - mostrarRelatorioVotos: exibe relatorios de voto
 - toString: Retorna String com Turno e Data de Geração

- Classe: RelatorioVereador
 - Características: Subclasse de Relatorio
 - Atributos:
 - IstVereadores:
 - Tipo: ArrayList
 - Descrição: lista de vereadores
 - Métodos:
 - montarRelario: monta relatório de voto
- Classe: Urna
 - Características: classe concreta
 - Atributos:
 - eltPresidente
 - Tipo: Eleitor
 - Descrição: Armazena o presidente, que também é um eleitor
 - intTurno
 - Tipo: int
 - Descrição: armazena o turno
 - blnTodosVotaram
 - Tipo: boolean
 - Descrição: verifica se todos os eleitores votaram
 - intVotosBrancoVereador
 - Tipo: int
 - Descrição: quantidade de votos em branco do vereador
 - intVotosNulosVereador
 - Tipo: int
 - Descrição: quantidade de votos nulos do vereador
 - intVotosNulosPrefeito
 - Tipo: int
 - Descrição: quantidade de votos nulos do prefeito
 - intVotosBrancoPrefeito
 - Tipo: int
 - Descrição: quantidade de votos em branco do prefeito
 - intQtdVotosTotalPrefeito
 - Tipo: int
 - Descrição: quantidade total de votos para prefeito
 - intQtdVotosTotalVereador

- Tipo: int
 - Descrição: quantidade de votos total para verador
- dtInicioVotacao
 - Tipo: Date
 - Descrição: Data de início da votação
- dtFimVotacao
 - Tipo: Date
 - Descrição: Data de fim da votação
- Métodos:
 - configurarSegundoTurno: configura o segundo turno.
 - toString: Retorna String com elementos dessa classe
 - verificarCodigoPrefeito: Função que verifica se o voto é válido baseado na quantidade de dígitos do código do candidato
 - verificarCodigoVereador: Função que verifica se o voto é válido baseado na quantidade de dígitos do código do candidato
 - votarPrefeito: Pesquisa no ArrayList o prefeito de codigo igual ao intVoto e vota em um prefeito
 - votarVereador: Pesquisa no ArrayList o vereador de codigo igual ao intVoto e vota em um prefeito
 - votoBranco: armazena o voto em branco
 - votoNulo: armazena o voto nulo
- Classe: main
 - Característica: classe principal
 - Atributos:
 - mMenu
 - Tipo: Menu
 - Descrição: Responsável por gerenciar os menus
 - blnSair
 - Tipo: boolean
 - Descrição: Enquanto for true, mantém o programa rodando
- Classe: Menu
 - Característica: classe concreta
 - Atributos
 - ctdControlador
 - Tipo: Controlador

- Descrição: Responsável por gerenciar todo o processo de votação

- Métodos:

- menuPrincipal: exibe o menu principal
- menuAdministrativo: Responsável por exibir o menu principal do ambiente administrativo
- menuCandidato: Responsável por exibir as opções disponíveis para a gestão dos Candidatos
- menuEleitor: Responsável por exibir as opções disponíveis para a gestão dos Eleitores
- menuEscolherCandidato: Responsável por realizar a gestão entre Prefeito e Vereador
- menuPartido: Responsável por exibir as opções disponíveis para a gestão dos Partidos
- menuRelatorio: Responsável por exibir as opções disponíveis para a gestão dos Relatórios
- menuUrna: Responsavel por exibir opcoes para processo de votação
- receberNumero: Função que usa tratamento de exceção para não permitir que o usuário atribua uma string para um atributo do tipo inteiro.

- Classe: Controlador

- Características: Classe concreta

- Atributos:

- IstPessoa:
 - Tipo: ArrayLista
 - Descrição: lista de pessoa
- IstPartido:
 - Tipo: ArrayLista
 - Descrição: lista de partido
- IstUrnas:
 - Tipo: ArrayLista
 - Descrição: lista de urnas
- IstRelatorio:
 - Tipo: ArrayLista
 - Descrição: lista de relatórios

- Métodos:

- alterarCandidato: Método responsável por realizar a alteração do Candidato

- alterarEleitor: Método responsável por realizar a alteração do eleitor
- alterarPartido: Método responsável por realizar a alteração do partido
- cadastraPartido: Método responsável por cadastrar um partido no ArrayList
- cadastrarCandidato: Método responsável por cadastrar um candidato no ArrayList
- cadastrarEleitor: Método responsável por cadastrar um eleitor no ArrayList
- cadastrarVicePrefeito: Método responsável por cadastrar um vice prefeito no ArrayList
- comecarVotacao: processo de votação
- converterDateToString: Recebe como parâmetro uma data e retorna uma string formatada (dd/MM/yyyy)
- converterStringToDate: Recebe uma data no tipo String e transforma no tipo Date
- criarPresidente: cria urna e define o presidente
- escolherRelatorioPrefeito: Opções do relatório de prefeito
- escolherRelatorioVereador: Opções do relatório de vereador
- excluiPartido: Método responsável por remover um Partido do ArrayList de Partidos
- excluirCandidato: Método responsável por remover um Partido do ArrayList de Candidato
- excluirEleitor: Método responsável por remover um Partido do ArrayList de Eleitor
- Iniciar: Cria objetos default
- listarCandidatos: Método responsável por listar os candidatos
- listarEleitores: Método responsável por listar os eleitores
- listarPartido: Método responsável por listar os partidos
- listarPrefeitos: Método responsável por listar os prefeitos
- listarUrnas: Método responsável por listar as urnas
- listarVereadores: Método responsável por listar os vereadores
- pesquisaPartido: Método responsável por pesquisar certo partido, pelo código
- pesquisarCandidato: Método responsável por pesquisar candidato

- `pesquisarEleitor`: Método responsável por pesquisar eleitor
- `receberNumero`: Função que usa tratamento de exceção para não permitir que o usuário atribua uma string para um atributo do tipo inteiro.

- Construtores:

- `public Candidato(int intCodigo, String strEmail, Partido prtPartido, String strNome, Date dteDataNascimenbto);`
- `Candidato();`
- `public Candidato(int intCodigo, int intVotos, Partido prtPartido, String strNome, Date dteDataNascimento);`
- `CodigoInvalidoException(String strMensagem);`
- `public Eleitor(int intTitulo, int intZona, int intSecao, String strNome, Date dteDataNascimento);`
- `public Partido(String strNome, int intCodigo,String strSigla);`
- `public Pessoa(String strNome, Date dteDataNascimento);`
- `Public Pessoa();`
- `public Prefeito(int intCodigo, String strEmail, Partido prtPartido, String strNome, Date dteDataNascimenbto, VicePrefeito vcpVice);`
- `public Prefeito();`
- `public Prefeito(VicePrefeito vcpVice, boolean blnEstaNoSegundoTurno, int intCodigo, int intVotos, Partido prtPartido, String strNome, Date dteDataNascimento);`
- `public Vereador(int intCodigo, String strEmail, Partido prtPartido, String strNome, Date dteDataNascimenbto);`
- `public Vereador();`
- `public VicePrefeito(int intCodigo, String strEmail, Partido prtPartido, String strNome, Date dteDataNascimenbto);`
- `public Relatorio(int intQtdVotosTotal, int intQtdVotoBranco, int intQtdVotoNulo);`
- `public RelatorioPrefeito(int intTurno, int intQtdVotosTotal, int intQtdVotoBranco, int intQtdVotoNulo,ArrayList<Pessoa> IstPessoa);`
- `public RelatorioVereador(int intQtdVotosTotal, int intQtdVotoBranco, int intQtdVotoNulo, ArrayList<Pessoa> IstPessoa);`
- `public Urna(Eleitor eltPresidente,Date dtelniDate) ;`

- Encapsulamento:

- Public: todos métodos que não são private ou protected. Foi utilizado public, pois todos métodos estão private.
- Private: todos atributos são private no programa, para melhor gerenciamento.
- Protected: os métodos para manipulação de Date, pois são utilizados somente dentro do package em questão.
- Composição:
 - Classe abstrata Candidato:
 - Composição em prtPartido:
 - Tipo: Partido;
 - Descrição: Representa o partido o qual o candidato é filiado.
 - Classe Pessoa:
 - Composição em dteDataNascimento:
 - Tipo: Date;
 - Descrição: Representa a data de nascimento da pessoa em questão.
 - Classe Prefeito:
 - Composição em vcpVice:
 - Tipo: Vice Prefeito;
 - Descrição: Dados do vice-prefeito que está se candidatando com este prefeito.
- Herança:
 - Pessoa: contém atributos e métodos que uma pessoa deve possuir.
 - Classes que herdam as características desta classe:
 - Candidato;
 - Eleitor.
 - Atributos:
 - strNome:
 - Tipo: String;
 - Descrição: Nome da pessoa.
 - dteDataNascimento:
 - Tipo: Date
 - Descrição: Data de nascimento da pessoa
 - Métodos:
 - public String getStrNome();
 - public void setStrNome(String strNome);
 - public Date getDteDataNascimento();
 - public void setDteDataNascimento(Date dteDataNascimento);

- public String toString();
- Interface:
 - Politico: contém os métodos que um político que recebe votos deve possuir.
 - Classes que implementam a interface:
 - Prefeito;
 - Vereador.
 - Métodos:
 - public boolean validarVoto(int intCodigo): verifica se o voto é válido baseado na quantidade de dígitos do código do candidato fornecido pelo usuário.
- Classe abstrata:
 - Candidato: contém atributos e métodos que um candidato deve possuir. Herda características da classe Pessoa.
 - Classes que herdam as características desta classe:
 - Prefeito;
 - Vice Prefeito;
 - Vereador.
 - Atributos:
 - intCodigo:
 - Tipo: Int;
 - Descrição: Código do candidato.
 - intVotos:
 - Tipo: Int;
 - Descrição: Quantidade de votos do candidato.
 - strEmail:
 - Tipo: String;
 - Descrição: Email do candidato.
 - prtPartido
 - Tipo: Partido;
 - Descrição: Partido do candidato.
 - Métodos:
 - public int getIntCodigo();
 - public String getStrEmail();
 - public void setStrEmail(String strEmail);
 - public Partido getPrtPartido();
 - public void setPrtPartido(Partido prtPartido);
 - public String toString();
 - public void receberVoto();

- public int getIntVotos().
- Coleções:
 - ArrayList:
 - Nome: ArrayList<Pessoa>
 - Função: Armazena Objetos de tipo Pessoa como: Candidatos e Eleitores.
 - Nome: ArrayList<Urna>
 - Função: Armazena Objetos de tipo Urna, para gerenciar as urnas existentes na eleição.
 - Nome: ArrayList<Partidos>
 - Função: Armazena Objetos de tipo Partido.
 - Nome: ArrayList<Relatorios>
 - Função: Armazena Objetos de tipo Relatório, para gerenciar os resultados obtidos nas urnas e consequentemente os resultados da eleição.
- Polimorfismo:
 - Sobrecarga:
 - Construtores das classes Candidato, Prefeito, Vereador e Pessoa.
 - Sobreposição:
 - Uso: métodos toString():
 - Descrição: exibir mensagens de acordo com cada classe
 - Uso: método verificarCodigo():
 - Descrição: Verifica se o código digitado pelo usuário é válido. Caso seja um prefeito, o código deve conter dois números inteiros, caso seja vereador a quantidade é quatro.
 - Inclusão:
 - Uso: ArrayList<Pessoa>:
 - Descrição: Foi utilizado polimorfismo de inclusão para verificar de qual tipo é cada elemento do ArrayList, ou seja, para ver se a Pessoa era Eleitor, Prefeito ou Vereador.
- Tratamento de exceção:
 - Foi criada a classe CodigoInvalidoException que estende a classe Exception. Não permite que o usuário digite um código inválido para prefeito ou vereador.
 - A função receberNumero da classe Controlador usa tratamento de exceção para não permitir que o usuário atribua uma string para um atributo do tipo inteiro.

