

Relatório de projeto LP2

Jessé Monteiro Ferreira - 118210282

O design do projeto foi desenvolvido de maneira que tornasse o código mais limpo e organizado, priorizando encapsulamento das informações para maior manutenibilidade e facilidade em alterações. Foi criada abstrações e uso de controller para melhor compreensão e facilitar a visualização dos atributos e métodos, evitando reescrita de código e priorizando a eficiência.

Foram utilizados os conceitos de herança e polimorfismo, assim como o uso exceções para o futuro tratamento das entradas inesperadas ou inválidas. Uma classe Util foi utilizada para armazenar métodos úteis e genéricos para o desenvolvimento, evitando reescrita de códigos.

As próximas seções detalham a implementação em cada caso.

Caso 1: Foi solicitado para implementar uma função para cadastro de pessoas. Foram desenvolvidas as classes Pessoa e ECO (Facade), como especificado foram implementados dois métodos, uma que recebe partido político como parâmetro e outra sem. As pessoas criadas são adicionadas a uma collection(ArrayList), foram respeitados os casos de exceção e foram lançadas as exceções solicitadas de maneira que todos os casos foram cobertos rodando o script de teste sem erros.

Caso 2: Foi solicitado a implementação de um método para cadastro de deputados. Deputado herda os atributos de Pessoa. Como condições para o sucesso do cadastro o deputado deveria ser uma pessoa com partido, condição que era verificada na ECO e lançada as exceções quando as entradas não eram válidas, os deputados eram armazenados em uma Collection (ArrayList).

Caso 3: Foi solicitado um método para exibir uma pessoas, antes de exibir era verificado se era político, as informações de retorno do método eram formatadas de acordo com o dados inseridos em cada pessoa. São utilizados os métodos buscaPessoa() e buscaDeputado() para buscar a partir do DNI que retorna a pessoa buscada e o método exibirPessoa() retorna todas as informações armazenadas dessa pessoa.

Caso 4: Foi solicitado o cadastro de partidos em uma base governista, foi desenvolvido a classe Partido para facilitar a manipulação das informações armazenadas, os novos partidos foram armazenados em uma Collection (ArrayList) através do método cadastrarPartido() e para exibir o método exibirBase() foi usado o método Sort() que ordena de maneira lexicográfica os partidos.

Caso 5: Foi solicitado o cadastro de comissões legislativas, o método cadastrarComissao() recebe o tema e os DNI's dos políticos que participam da comissão, é verificado os casos para lançamento de exceções para os casos de entrada inválida.

Caso 6: Foi solicitado o cadastro e a exibição das propostas legislativas, onde foi utilizado os conceitos de herança e polimorfismo. As classes filhas PL, PLP e PEC são do tipo Projeto (classe Pai), evitando a reescrita de código e facilitando a sua manutenção. Os

métodos `cadastrarPL()`, `casdastrarPLP()` e `cadastrarPEC()` foram implementados e os casos de exceção foram analisados e passaram nos testes. Métodos da classe `Util` também foram utilizados. O uso de `HashMap()` foi escolhido para ter o controle dos códigos dos projetos, onde a chave é o ano e valor é o índice do último projeto criado no devido ano.

Considerações: Os casos 7, 8, 9 e 10 não foram implementados devido a desistência dos outros membros do grupo, com esse entrave não foi possível terminar em tempo hábil o projeto.