



Orientação a Objetos - FGA0158

Semestre: 2021_1

Grupo 3

Detalhamento da implementação

Entidades:

Para o desenvolvimento do projeto as principais classes criadas foram todas no pacote entidade, dando corpo ao projeto.

- **Pessoa:** Classe criada onde são armazenados os dados de cada membro da república cadastrado. O método `getNome`, `getEmail` e `getRenda` são usados para se obter o nome, email e renda, respectivamente, da pessoa. Por outro lado, os métodos `setRenda` e `toString` são utilizados, para atualizar a renda e registrar textualmente os dados cadastrados.
- **Categoria:** Classe criada onde é armazenado o dado da categoria cadastrada. O método `getNome` é usado para se obter o nome da categoria.
 - **SubCategoria:** Classe criada através da herança da classe Categoria onde é armazenada o dado da subcategoria cadastrada. O método `getCategoria` é utilizados para se obter a classificação da subcategoria.
- **Despesa:** Classe criada para registrar as despesas da república onde `getDescrição`, `getValor` e `getCategoria` fornecem, respectivamente, as informações sobre a descrição, valor e categoria de cada despesa. O método `toString` é aplicado para registrar textualmente os dados cadastrados.
- **República:** É a classe principal das entidades, em programa, mantém os dados de referência para as instâncias do tipo Pessoa e Despesa, de forma a gerenciar a manipulação destes dados pelo programa principal. Os métodos de cadastro são utilizados para adicionar uma instância do tipo específico na lista de dados de programa da república, os métodos “ler” fornecem uma inicialização dos dados salvos nos arquivos de persistência, enquanto os métodos “gravar” atualizam estes dados.

Serviços:

Pacote criado para implementar o cálculo da dívida em duas opções: cálculo igualitário e cálculo proporcional.

- **CalculoDívida:** Classe criada que, através dos dados recebidos da república, é capaz de calcular a renda total e a despesa total através dos métodos `rendaTotal` e `despesaTotal`.
 - **RegraIgualitaria:** Subclasse capaz de calcular a dívida por pessoa, através do método `divida`, dividindo o valor total das despesas pelo número de pessoas na república.
 - **RegraProporcional:** Subclasse capaz de calcular a dívida por pessoa, através do método `divida`, dividindo o valor total entre os membros da república de forma proporcional com a renda de cada um.

App:

Pacote que contém o programa principal e os meios de interface com o usuário.

- **Main:** Classe responsável pela parte principal do programa que irá coletar informações sobre o mês e o ano, apresentando uma lista de opções para cada dado. Em seguida a classe fornece as opções de divisão das dívidas e inicia o menu do programa apresentando as opções:
 - **Cadastrar pessoa:** irá criar uma nova pessoa no programa através do método `criaPessoa` da UI e registrar estas através do método `cadastrarPessoa` da República.
 - **Cadastrar despesa:** irá criar uma nova pessoa no programa através do método `criaDespesa` da UI e registrar estas através do método `cadastrarDespesa` da República.
 - **Excluir pessoa:** Permite selecionar uma das pessoas cadastradas para ser removida do cadastro.
 - **Excluir despesa:** Permite selecionar uma das despesas cadastradas para ser removida do cadastro.
 - **Calcular dívida:** Permite escolher uma pessoa para calcular a dívida desta.
 - **Imprimir relatório:** Imprime o relatório das pessoas.
 - **Mudar Regra de cálculo:** Permite alterar o cálculo da dívida.
 - **Sair:** Fecha o programa

Por fim o método `gravarPessoa` e `gravarDespesa` irão registrar as pessoas e as despesas no arquivo `txt`.

- **UI:** Do inglês User Interface (Interface de Usuário), a classe mantém os métodos estáticos que fornecem o contato com o usuário de forma a enviar dados para o programa principal, ou exibir dados ao usuário.
 - O método selecionar é um método genérico que permite que o usuário selecione um dado dentro de uma lista pré-determinada por uma caixa de seleção. Além deste existem alguns métodos de selecionar mais específicos, que são utilizados para casos específicos, com uma personalização maior.
 - Os métodos criaPessoa e criaDespesa recolhem os dados necessários para criar uma instância do tipo especificado e retornam a referência a ela.
 - O método showMenu exibe a string menu, definida na classe Main, enquanto retorna o valor inteiro, passado pelo usuário, referente à opção da próxima funcionalidade realizada pelo programa.
 - O método mensagem recebe uma string como parâmetro, que é então exibida ao usuário. Os métodos de alerta são semelhantes, mas exibem uma caixa de alerta com a mensagem recebida, e pode ou não ter o título personalizado.
- **Exceções:** Serve para tratamento de eventuais erros que ocorrem no meio da aplicação, seja por valores inválidos inseridos pelo usuário, no caso da programa da republica foram implementadas algumas exceções para verificar o tipo de valor informado pelo usuário e assim impedindo que o programa quebre ou pare de funcionar, mostrando sempre uma mensagem de erro e requisitando uma nova ação do usuário de forma que mediação entre a logica do programa e o usuário ocorra tudo bem. Além dessas, foram implementas 5 exceções que foram exigidas no corpo do trabalho, que são:
 - DadosPessoaisIncompletosException: que verifica os dados pessoais inseridos são validos e retorna uma mensagem pedindo para inserir os dados novamente caso algum dado pessoal inserido seja invalido.
 - RendimentoInvalidoException: que verifica se o rendimento cadastrado é um numérico e positivo e retorna uma mensagem pedindo para cadastrar um rendimento válido caso não foi.
 - CategoriaNaolInformadaException: verificar se a categoria foi cadastrada e retorna uma mensagem pedindo para cadastrar uma categoria valida caso não foi.
 - DescricaoNaolInformadaException: verifica se a descrição informada é valida e retorna uma mensagem pedindo para cadastrar uma descrição valida caso não foi.