Descrição das Classes e Métodos

# 1. CategoriaDespesa

A classe 'CategoriaDespesa' é uma enumeração que define as categorias de despesas. Cada categoria possui um rótulo descritivo associado. As categorias definidas são: Alimentação, Transporte, Residência, Saúde, Educação, Entretenimento e Outras Receitas. A classe contém um atributo 'label' do tipo String, que é utilizado para armazenar o rótulo descritivo da categoria de despesa. A classe também possui um método 'toString', que retorna o rótulo da categoria.

## Métodos:

1. getLabel(): Retorna o rótulo da categoria.

2. toString(): Retorna o rótulo da categoria como uma String.

# 2. Despesa

A classe 'Despesa' é uma subclasse da classe 'Lancamento'. Ela representa uma despesa no sistema, onde o valor da despesa é sempre negativo para representar um débito. A classe sobrescreve o método 'setValor' para garantir que o valor sempre seja negativo.

## Métodos:

1. setValor(double valor): Caso o valor informado seja positivo, ele é convertido para negativo, garantindo que o valor da despesa seja registrado corretamente como débito.

# 3. LancamentoRepository

A interface 'LancamentoRepository' define as operações de repositório para gerenciar lançamentos financeiros. Ela permite salvar e carregar uma lista de lançamentos. As classes que implementam esta interface devem fornecer a lógica para a persistência de dados de lançamentos.

## Métodos:

1. salvar(List<Lancamento> lancamentos): Salva uma lista de lançamentos no repositório.

2. carregar(): Carrega todos os lançamentos armazenados no repositório.

# 4. LancamentoRepositoryImpl

A classe 'LancamentoRepositoryImpl' é uma implementação da interface 'LancamentoRepository'. Ela realiza a leitura e escrita de lançamentos em um arquivo CSV, permitindo salvar e carregar os dados de lançamentos financeiros. A classe utiliza o BufferedWriter para salvar os dados e o BufferedReader para carregá-los.

## Métodos:

1. salvar(List<Lancamento> lancamentos): Realiza a gravação dos lançamentos no arquivo CSV.

2. carregar(): Carrega os lançamentos financeiros do arquivo CSV.

# 5. LancamentoService

A interface 'LancamentoService' define as operações de serviço para manipulação de lançamentos financeiros. Ela inclui métodos para adicionar, listar e consultar saldos de lançamentos.

## Métodos:

1. adicionarLancamento(Lancamento lancamento): Adiciona um novo lançamento financeiro.

2. listarLancamentos(): Retorna uma lista de todos os lançamentos registrados.

3. consultarSaldoAte(Date data): Consulta o saldo total até uma data específica, considerando receitas e despesas.

4. consultarSaldoTotal(): Retorna o saldo total considerando todas as receitas e despesas.

# 6. LancamentoServiceImpl

A classe 'LancamentoServiceImpl' é a implementação da interface 'LancamentoService'. Ela fornece os métodos necessários para adicionar lançamentos, listar lançamentos, consultar o saldo até uma data específica e consultar o saldo total. Os lançamentos são armazenados em uma lista interna da classe.

## Métodos:

1. adicionarLancamento(Lancamento lancamento): Adiciona um lançamento à lista de lançamentos.

2. listarLancamentos(): Retorna a lista de lançamentos ordenados por data.

3. consultarSaldoAte(Date data): Retorna o saldo até a data fornecida.

4. consultarSaldoTotal(): Retorna o saldo total de todos os lançamentos.

# 7. DateUtils

A classe 'DateUtils' fornece métodos utilitários para manipulação de datas. Ela inclui métodos para converter uma string representando uma data em um objeto Date e vice-versa. O formato utilizado é 'dd/MM/yyyy'.

## Métodos:

1. stringToDate(String dateString): Converte uma string representando uma data para um objeto Date.

2. dateToString(Date date): Converte um objeto Date para uma string no formato 'dd/MM/yyyy'.

# 8. MainFrame

A classe 'MainFrame' é a interface gráfica do sistema, construída com a biblioteca Swing. Ela permite que o usuário adicione receitas e despesas, consulte saldos e visualize a lista de lançamentos registrados. A classe contém botões para interagir com o sistema e realizar operações com as receitas e despesas.

## Métodos:

1. adicionarReceita(): Solicita ao usuário os dados para adicionar uma receita.

2. adicionarDespesa(): Solicita ao usuário os dados para adicionar uma despesa.

3. consultarSaldoAtual(): Exibe o saldo atual até a data de hoje.

4. consultarSaldoTotal(): Exibe o saldo total considerando todos os lançamentos.

5. listarLancamentos(): Exibe uma lista de todos os lançamentos registrados.