

Documentação de Implementação

Trabalho Prático 2 - Programação Orientada a Objetos

Aluno: Lucca Garcia Leão - 2016020967

Introdução

Neste trabalho, foi implementado uma aplicação em Java que simula o funcionamento de uma operadora de telefonia celular. No pacote operadora, temos todas as classes necessárias para o funcionamento total do sistema, incluindo as classes de exceção que foram implementadas e a classe Operadora, que é a classe final que faz as operações com todos os celulares e clientes, e a única que se relaciona com a interface. No pacote Interface, temos apenas a classe Interface, que é a única classe de interação com o mundo externo. A idéia é que a aplicação tenha funcionamento totalmente independente da interface escolhida, por isso existe a separação entre os pacotes de Interface e operadora. Nesse trabalho, a interface é feita apenas por meio de impressões na tela, mas é importante ressaltar que qualquer outra interface poderia ser escolhida sem precisar de qualquer alteração no pacote operadora.

Classes

Para essa aplicação, foram implementadas algumas classes necessárias para todas as funcionalidades do sistema, que são:

- **Ligacao:** essa classe representa o registro de uma ligação de celular, seus campos contém um objeto do tipo LocalDateTime, que representa o dia e a hora da ligação, e um campo inteiro que representa o tempo (em minutos) da ligação.
- **Cliente:** o cliente é representado por campos do tipo String para o nome, CPF/CNPJ, e endereço. O cliente não depende de um celular para existir, podendo existir clientes sem nenhum celular.
- **Celular:** Essa é a classe base para qualquer plano de celular que a operadora pode ter. Possui um campo de cliente (do tipo Cliente), número (int) e uma lista de ligações (ArrayList<Ligacao>), pois todo Celular pertence a um cliente, possui um número, e pode fazer uma ou mais ligações. O número do celular é um número de 8 dígitos escolhido aleatoriamente pelo sistema no momento em que o construtor de Celular é chamado.

- Operadora: a classe Operadora é a classe que realiza todas as operações definidas no trabalho. Sua implementação e métodos serão abordados com maior detalhe a seguir.
- CelularPrePago: classe derivada de Celular, possui um campo inteiro credits, que representa o número de créditos disponíveis do celular, e um campo do tipo LocalDate que representa a data de vencimento dos créditos. Toda vez que créditos são adicionados, a data de vencimento é atualizada para dali a 180 dias. Para simplificar o funcionamento, foi decidido que cada minuto de ligação custa 1 crédito.
- CelularPosPago: classe derivada de Celular, possui um campo inteiro que representa os minutos de ligação registrados, e um campo LocalDate que representa a data de vencimento da fatura. O usuário deve escolher a data de vencimento da fatura, e ao ser feito o pagamento da mesma, a data de vencimento é atualizada para dali a 30 dias.
- Interface: classe responsável por fazer a comunicação com o mundo externo para as classes do pacote operadora.

Foram implementadas também as seguintes classes de exceção:

- ExcecaoCreditosInsuficientes: exceção lançada quando o usuário tenta registrar uma ligação de duração maior que o número de créditos disponíveis.
- ExcecaoCreditosVencidos: exceção lançada quando o usuário tenta registrar uma ligação mas a data de validade dos créditos está vencida.
- ExcecaoNumeroInexistente: exceção lançada quando o usuário informa um número de celular que não existe na lista para algum método.

O restante das situações de erro são tratadas genericamente através da classe Exception.

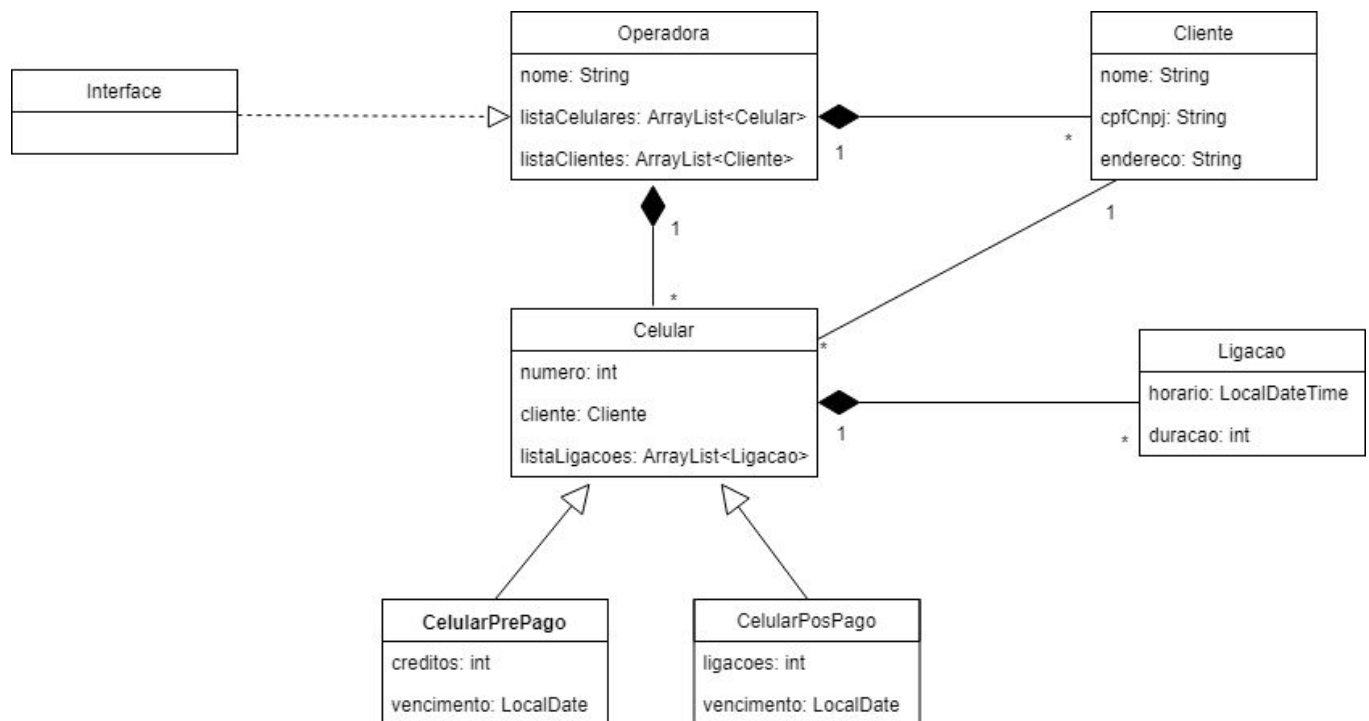
Classe Operadora

A classe Operadora possui um campo lista de clientes (ArrayList<Cliente>), um campo de lista de celulares (ArrayList<Celular>), e um campo de nome (String). O polimorfismo está presente nessa classe através da lista de Celulares. Os únicos celulares possíveis nessa operadora fazem parte do plano pré pago ou pós pago, e ambos celulares do plano pré pago e do plano pós pago podem fazer parte dessa lista, pois eles são derivados da classe base Celular. Algumas funções dessa classe devem ser aplicadas somente a celulares, como por exemplo a função adicionaCreditos(), que deve ser aplicada somente a celulares do tipo pré-pago. Para verificar que o Celular especificado realmente é do tipo requisitado pelo método chamado, é utilizado o

operador instanceof. Uma exceção é lançada caso o plano do celular seja Para testar se um celular é do tipo pré-pago, por exemplo:

```
if(cell instanceof CelularPrePago){ . . . }
```

Caso seja de interesse adicionar mais um plano de celular, basta criar uma classe que seja derivada de Celular, e sobrescrever seus métodos. Dessa forma, não será necessário reescrever as funções já implementadas na classe Operadora. O diagrama UML a seguir mostra a relação entre todas as classes:



Métodos da classe Operadora:

- `public Celular buscaCelular(int numero)`: método auxiliar implementada para facilitar a busca de um celular, visto que várias funções precisam fazer essa busca. Lança uma exceção `ExcecaoNumeroInexistente` se o celular procurado não for encontrado.
- `public void cadastrarCliente(Cliente c)`: adiciona o cliente `c` à lista de clientes da operadora.
- `public void habilitarCelularPosPago(Cliente c, int dia)`: cria um novo celular do plano pós pago com data de validade definida por dia.

- `public void habilitarCelularPrePago(Cliente c, int cred)`: cria um novo celular do plano pós pago com créditos iniciais `cred`, e `Cliente c`.
- `public void excluirCelular(int numero)`: exclui o celular especificado da lista de celulares. Lança exceções caso o celular não seja encontrado; caso o celular possua créditos (no caso do plano pré-pago), ou caso a fatura não tenha sido paga (no caso do plano pós-pago).
- `public void adicionaCreditos(int numero, int cred)`: adiciona créditos a um celular do tipo pré pago. Lança uma exceção caso o número não exista ou caso o celular não for do plano pré pago.
- `public void regLigacao(int numero, int d, int dia, int mes, int ano, int hora, int minuto)`: registra uma ligação ao número especificado com o horário e duração especificados.
- `public int getValorConta(int numero)`: retorna o valor da fatura de um celular do tipo pós pago. Lança uma exceção se o celular especificado não fazer parte do plano pós pago.
- `public int getValorCreditos(int numero)`: retorna o valor dos créditos de um celular do tipo pré pago. Lança uma exceção se o celular especificado não fazer parte do plano pré pago.
- `public LocalDate getDataVencimento(int numero)`: para celulares pré pagos, retorna a data de vencimento dos créditos. Para celulares pós pagos, retorna a data de vencimento da fatura.
- `public ArrayList<Ligacao> getExtrato(int numero, int dia, int mes, int ano)`: retorna o extrato de ligações de um celular a partir da data especificada.
- `public ArrayList<Cliente> getListaClientes()`: retorna a lista de clientes.
- `public ArrayList<Celular> getListaCelulares()`: retorna a lista de celulares.
- `public String getNomeOperadora()`: retorna o nome da operadora.