

IFRN - Instituto Federal do Rio Grande do Norte.
TADES - Tecnologia em Análise e Desenvolvimento de Sistemas.
Turma: 2017.1.
Professor: Plácido Netto.
Disciplina: Programação Orientada a Objetos.

Sistema de Vendas (Caixa) como exemplo de implementação no paradigma orientado a objetos

Módulos do Sistema

Saulo Daniel Ferreira Pontes

Wannderson Eduardo de Oliveira Nogueira

Natal/2017

Conteúdo dos pacotes

- *ifrn.edu.br.poo*:

Classes:

Main;
Operador;
Login (interface);
Admin;
ProductDataBase;
UserDataBase;
Usuario;
Venda;
Produto;
Operador;

- *libs*: SQLite e suas bibliotecas.

Descrição de Classes e Métodos

- **Main:** Classe principal Java, a classe que será executada pelo JRE. Contém todo o código do sistema que fará interação com o usuário.

+ static main(String[] args): void : Contém o código principal, com tratamento de erros e dinâmica de troca de telas.

+ static administradorLogin(): boolean :

+ static menuUsuario(): void : O menu onde aparece as operações em que o operador de caixa interagirá.

+ static menuAdministrador(): void : O menu onde aparece as operações em que o administrador do sistema interagirá.

- **Login:** Interface que oferece os métodos a ser implementados pela classe Usuário.

+ getPassword(): String

+ setPassword(String a): void

*Esses métodos são usados associados ao atributo *password* que receberá a senha.

- **Usuário:** Classe pai das classes Operador e Admin. Essa classe tem a função de fornecer atributos e métodos para ser usado pelas classes filhas e um identificador do operador (este identificador quando está **false** sinaliza que o login não foi feito pelo operador).

+ escolheOperador(): void

+ getOperador(): boolean

*Esses métodos são usados para alterar e revelar o valor boolean do atributo *operador*, que serve para identificar se é um administrador ou um operador de caixa.

- **Admin:** Identifica os requisitos que o administrador do sistema precisa para poder fazer configurações no sistema. Para ter acesso à administração do sistema é necessário uma senha, por isso essa classe usa o atributo *password* da classe Usuário através dos métodos *get* e *setPassword()* herdados da classe Usuário.

- **Operador:** É responsável por capturar a String com o nome do login do usuário (operador). Herda da classe Usuário os métodos *get* e *setPassword()* fazendo uso compartilhado desses métodos com a classe Admin. Ela tem métodos próprios *get* e *setLogin()* para manipular seu único atributo.

- **UserDataBank:** É a classe do banco de dados das contas de login do operador; usando ArrayList.

+ setOperator(Operador t): void : Adiciona uma conta de Login ao banco de dados.

+ searchLogin(Operador a): boolean : Procura um nome de usuário no banco de dados.

*Retorna **true** se o nome de usuário foi encontrada.

+ searchPassword(Operador a): boolean : Procura uma senha no banco de dados. *Retorna **true** se a senha foi encontrada.

+ apagaOperador(String string): boolean : Apaga uma conta determinada pelo parâmetro e checka se a conta foi apagada, se sim, retorna **true**.

- **ProductDataBank**: É a classe do banco de dados dos produtos; usando HashMap.

+ setItem(String chave, Produto item): boolean : Adiciona um objeto da classe Produto ao HashMap e associa a ele uma String.

+ getItem(String chave): Produto : Retorna um objeto da classe Produto ao receber como parâmetro uma String. Ele procurará o objeto com String igual ao parâmetro e retornará o Produto referente.

+ apagaItem(String chave): boolean : Apagará o objeto da classe Produto que tem associado a ele uma String igual a String usada como parâmetro e, caso sua operação tenha sido um sucesso, ela retornará **true**, senão, retornará **false**.

+ diminuiQtde(String code, int number): void : Diminui o número (**int**) do atributo *quantidade* e calcula o valor multiplicado para o atributo *valor* (**double**) da classe Produto.

- **Venda**: Realiza todas as operações concernentes ao caixa, função principal do sistema. Dessa forma, ela é responsável por guardar a informação dos produtos passados no caixa num HashMap, e calcula o total.

+ getTotal(): double : Retorna o valor do atributo *total*. Retorna o total da compra.

+ getVenda(String string): Produto : Retorna um objeto da classe Produto no HashMap, com chave igual à String passada como parâmetro.

+ setVenda(String chave, Produto product): boolean : Adiciona um objeto da classe Produto associado a uma String e retorna **true** se a operação foi bem sucedida.

+ calculaTotal(double tot): void : Atualiza o atributo *total* através de soma. Cada produto passado no caixa tem seu valor incrementado no atributo.