

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE - UFRN

Instituto Metr pole Digital
IMD0040 - Linguagem de Program  o 2

ATIVIDADE

Orienta  es:

- Separe as classes em pacotes diferentes de acordo com suas fun  es
- Crie o javadoc de no m nimo 3 classes, com coment rios da classe, do construtor, atributos, m todos e par metros.
- Por uma quest o did tica implemente todos os gets e sets dos atributos das classes

Tarefa 01:

Classes:

- Aplicacao (main)
- Funcionario:
 - Construtor
 - Atributos: matricula, nome e CPF
- Diretor (Herda de Funcion rio)
 - Construtor
 - Atributo: setor, senha
- Gerente (Herda de Funcion rio)
 - Construtor
 - Atributo: setor e senha
- Secretario(Herda de Funcion rio)
 - Construtor
- AuxiliarAdministrativo(Herda de Funcion rio)
 - Construtor
- Repositorio:
 - Atributo: banco (uma cole  o de funcion rios)
- RepositorioFuncionario: Implementa a interface IRepositorio

Interface:

- IRepositorio:
 - void inserirFuncionario(Funcionario f);*
 - void deletarFuncionario(int matricula);*
 - void atualizarFuncionario(int matricula);*
 - Funcionario buscarFuncionario(int matricula);*
 - void listarFuncionarios();*

Tarefa 02:

- Implemente todos os m todos da Classe RepositorioFuncionario utilizando a Classe Repositorio como base de dados, para nossa atividade devemos ter apenas um reposit rio.

- Na classe Aplicacao crie os seguintes métodos:
 - listarFuncionarios(ArrayList<Funcionario> f)
 - listarGerentes(ArrayList<Funcionario> f)
 - listarAuxiliarAdministrativo(ArrayList<Funcionario> f)
 - imprimirTodosNomesSetor(ArrayList<Funcionario> f)

Observação: A classe Aplicacao, deve conter a base de dados com inserções de diversos funcionários de tipos diferentes;

Tarefa 03:

- Crie mais duas classes:
 - Cliente (nome, senha)
 - Construtor
 - Fornecedor (nome, senha, cidade)
 - Construtor
- Clientes e fornecedores devem obrigatoriamente implementar o método autentica, por isso devem herdar da classe UsuarioAutenticavel. (O sistema da empresa não pode ser acessado por Funcionário)
 - public boolean autentica(int senha); // Este método apenas verifica se o Cliente e fornecedor podem acessar
- Apenas fornecedores da cidade de Natal podem autenticar.
- A classe UsuarioAutenticavel não pode ser instanciada.

Tarefa 04:

- Crie Classes de tratamento de exceções, de acordo com as informações abaixo:
 - Deletar Funcionário que não encontre a matrícula na base de dados
 - Atualizar Funcionário que não encontre a matrícula na base de dados
- Todas as classes de exceções devem ter uma mensagem personalizada da exceção, para isso devem utilizar o construtor do pai.

Tarefa 05:

- Crie Classes Testes de no mínimo 3 classes do sistema, crie testes de retornos e de conteúdos.

Tarefa 06:

- Crie um JAR que contenha um método chamado comparaUtils que recebe dois objetos e retorna true caso eles sejam do mesmo tipo ou false caso contrário. Exporte para que possa ser usado por outro projeto java.