



**Proposta:** Sistema de gerenciamento de empregados da empresa IMDCorp.

### **Funcionalidades e características:**

O sistema a ser desenvolvido deve ser capaz de criar e gerenciar empregados da empresa IMDCorp

### **Classes e enumerações para criação:**

#### **Classe Endereco**

Crie uma classe chamada “Endereco” com os atributos privados:

- rua (String);
- numero(Integer);
- bairro(String );
- cidade(String);
- cep(String);

Dentro da classe “Endereco” crie todos os métodos get/gets públicos para os atributos e um construtor com todos os atributos.

#### **Enumerações**

- Nivel (I, II, III, IV, V, VI, VII, VIII)
- Formacao (ESPECIALIZACAO, MESTRADO, DOUTORADO)
- Genero (MASC, FEM, OUTRO)

#### **Classe Pessoa (abstrata)**

Crie uma classe chamada “Pessoa” como os seguintes atributos privados:

- nome (String);
- cpf (String);
- dataNascimento (LocalDate);
- genero (Genero);
- endereco (Endereco);

Dentro da classe “Pessoa” crie um construtor parametrizado e todos os métodos get/sets públicos para os atributos..

### **Interface Funcionário**

Crie uma interface chamada “Funcionario” com os seguinte atributos e métodos

- matricula (Long);
- salario (Double);
- departamento (String);
- cargaHoraria (Integer);
- dataIngresso (LocalDate);
- Double calculaSalario();

### **Classe Professor**

Crie uma classe chamada “Professor” com os seguinte atributos e métodos:

- nivelProfessor (Nivel);
- formacaoProfessor (Formacao);
- disciplinas (List<String>);

Crie um construtor parametrizados para inicializar todos os atributos da classe e os métodos get/set (públicos). Essa classe de herdar Pessoa e Funcionario;.

### **Classe TecnicoADM**

Dentro desta classe crie os atributos privados :

- nivelTecnico (Nivel);
- insalubridade(Boolean);
- funcaoGratificada(Boolean);

um construtor parametrizado e os métodos get/set necessários (públicos).

### **Classe BancoDAO**

Crie uma classe chamada “BancoDAO” - singleton - que armazene e gerencie um ArrayList do tipo “Pessoa”:

- ArrayList<Pessoa> funcionario;

Dentro dessa classe crie os seguintes métodos públicos:

- `ArrayList<Pessoa>getArrayPessoa()`
- `BancoDAO getIntance()`

### **Classe Operacoes**

- `cadastrarProfessor()` - Cadastra um novo professor
- `cadastrarTecnicoADM()` - Cadastra um novo tecnicoADM
- `listarProfessores()` - Lista todos os professores por nome e disciplina;
- `listarTecnicosADM()` - Lista todos os tecnicosADM por nome e função;
- `deletarProfessor(int matricula)` - Apaga um professor a partir de sua matrícula
- `deletarTecnicoADM(int matricula)` - Apaga um TecnicoADM a partir de sua matrícula
- `buscarProfessor(int matricula)` - Localiza e exibe um professor a partir de uma matrícula passada como parâmetro.
- `buscarTecnicoADM(int matricula)` - Localiza e exibe um TecnicoADMa partir de uma matrícula passada como parâmetro.
- `CalcularSalario():` O salário base para professores é 4000,00 e de técnicos é 2500,00. No cálculo do salário de um professor deve ser levado em conta o nível e a formação (acrescente 5% a cada nível e 25% para especialização, 50% para mestrado e 75% para doutorado). O cálculo do salário do Técnico leva em conta o nível e a formação (acrescente 3% a cada nível e 25% para especialização, 50% para mestrado e 75% para doutorado) e acrescente 25% caso tenha insalubridade e 50% caso tenha função gratificada.

### **Atividades gerais do sistema**

- Essas informações devem ser armazenadas em um arquivo binário (Use serialização de classes). Ao abrir o programa faça a leitura dos dados e ao concluir o programa salve no arquivo.
- Crie no mínimo 10 professores e 10 tecnicosADM (20 funcionários)..

- Crie um menu interativo, via console, no qual é possível escolher qual função a chamar.
- O código deve ser modularizado, com todas as classes em pacotes/pastas próprias;

### **Critérios de avaliação**

- Código corretamente modularizado (1,0);
- Informações salvas nos arquivos (1,0);
- Utilização da herança, interfaces, collections, Date (2,0);
- Uso do singleton (1,0)
- Implementação correta de todos os métodos solicitados (2,0);
- Código organizado, comentado, de fácil entendimento e seguindo boas práticas (3,0);

### **Entregáveis**

O código deverá ser compactado em formato ZIP e enviado por meio do SIGAA (tarefa). Na pasta compactada, inclua obrigatoriamente um arquivo README.txt explicando como executar o código e o funcionamento das funções. A ausência desse arquivo resultará em um desconto de 1,0 ponto na nota.

Atenção: O uso evidente de inteligência artificial (IA) na realização desta atividade acarretará redução na nota.