



## **Projeto A - Sistema de RH da empresa Atacado dos Calçados**

### **Características da empresa.**

A ideia central do projeto é criar um sistema de RH (recursos humanos) de uma empresa de calçados chamada Atacado dos Calçados. A empresa dispõe de três tipos de profissionais:

- um gerente;
- cinco vendedores;
- três asg (auxiliar de serviços gerais);

Além desses há também o dono da empresa, o senhor Antônio Augusto de Souza Filho.

Todos os profissionais ASG têm direito a um adicional de insalubridade de 5% sobre o valor do salário pago no mês. Os vendedores, por sua vez, são classificados em A, B ou C, a depender do desempenho desses vendedores. Por ser uma empresa de pequeno porte, o senhor Antônio assume parte das atividades administrativas, sendo auxiliado pelo gerente Gustavo. As faltas ao serviço são descontadas do salário dos funcionários (vendedores, asg e gerente) no pagamento dos salários. A quantidade de filhos também influencia nos salários. Para cada filho o funcionário terá direito a um incremento de 100,00 reais em sua remuneração.

### **Implementação**

Crie em um arquivo denominado `util.hpp` duas struct Data (int ano, int mes e int dia) e Endereco (string cidade, string bairro, string rua, string cep, int numero). Utilize o `typedef` para definir o nome dessas struct.

Crie um classe concreta chamada Pessoa com os seguintes atributos:

privado

- nome (string)
- cpf (string)
- dataNascimento (Data)

enderecoPessoal (Endereco)

- estadoCivil (string) (possíveis: solteiro(a), casado(a), viuvo(a), divorciado(a))
- qtdFilhos (int) (Quantidade de filhos)

e os seguintes métodos:

público

- construtor vazio;
- construtor parametrizados (com todos os atributos);
- gets e set (para todos os atributos)

Crie uma classe abstrata chamada Funcionario com os seguintes atributos:

privado

- salario (string)
- matricula (string)
- ingressoEmpresa (Data) - Data que o funcionário entrou na empresa

e os seguintes métodos:

público

- gets e set (para todos os atributos)
- float calcularSalario(int diasFaltas) - virtual
- float calcularRecisao(Data desligamento) - virtual

Crie um classe concreta chamada Asg que herda de Pessoa e Funcionario (herança múltipla) com os seguintes atributos:

privado

- adicionalInsalubridade (float)

e os seguintes métodos:

público

- construtor vazio;
- construtor parametrizados (com todos os atributos);
- gets e set (para todos os atributos);
- métodos abstratos da classe Funcionario implementados.

o cálculo do salário do ASG deverá ser feito da seguinte forma:

1. As faltas deverão ser descontadas do salário base.
2. Após, adicionar o adicional de insalubridade
3. Por último, somar o adicional por filho do funcionário.

O cálculo da rescisão (demissão) deve ser feita da seguinte forma:

1. O funcionário ganha um salário base por ano trabalhado.
2. O tempo de trabalho será calculado em subtraindo os elementos da struct Data considerando a data de ingresso (classe Funcionario) e data de desligamento (passado como parâmetro da função calculaRecisão).
3. A dica é converter os meses e dias para frações de ano.

Crie um classe concreta chamada Vendedor que herda de Pessoa e Funcionario (herança múltipla) com os seguintes atributos:

privado

- tipoVendedor (char)

e os seguintes métodos:

público

- construtor vazio;
- construtor parametrizados (com todos os atributos);
- gets e set (para todos os atributos);
- métodos abstratos da classe Funcionario implementados.

o cálculo do salário do Vendedor deverá ser feito da seguinte forma:

1. As faltas deverão ser descontadas do salário base;
2. Após, adicionar uma gratificação que dependerá do tipo de vendedor (25% para vendedores A, 10% para vendedores B e 5% para vendedores C)
3. Por último, somar o adicional por filho do funcionário.

O cálculo da rescisão (demissão) deve ser feita da seguinte forma:

1. O funcionário ganha um salário base por ano trabalhado.
2. O tempo de trabalho será calculado subtraindo os elementos da struct Data considerando a data de ingresso (classe Funcionario) e data de desligamento (passado como parâmetro da função calculaRecisão).
3. A dica é converter os meses e dias para frações de ano.

Crie um classe concreta chamada Gerente que herda de Pessoa e Funcionario (herança múltipla) com os seguintes atributos:

privado

- participacaoLucros (float)

e os seguintes métodos:

público

- construtor vazio;
- construtor parametrizados (com todos os atributos);
- gets e set (para todos os atributos);
- métodos abstratos da classe Funcionario implementados.

o cálculo do salário do Vendedor deverá ser feito da seguinte forma:

1. As faltas deverão ser descontadas do salário base;
2. Após, some a participação no lucros (atributo da própria classe)
3. Por último, somar o adicional por filho do funcionário.

O cálculo da rescisão (demissão) deve ser feita da seguinte forma:

1. O funcionário ganha um salário base por ano trabalhado.
2. O tempo de trabalho será calculado subtraindo os elementos da struct Data considerando a data de ingresso (classe Funcionario) e data de desligamento (passado como parâmetro da função calculaRecisão).
3. A dica é converter os meses e dias para frações de ano.

Crie uma classe chamada Empresa com os seguintes atributos:

privado

- faturamentoMensal (float);
- nomeEmpresa (string);
- cnpj (string);
- dono (Pessoa);
- asgs (vector<Asg>);
- vendedores (vector<Vendedor>);
- gerentes (vector<Gerente>);

e os seguintes métodos:

público

- construtor vazio;
- construtor parametrizados (apenas com o nomeEmpresa, cnpj e faturamentoMensal);
- gets e set (faturamentoMensal, nomeEmpresa, cnpj));
- gets (dono, asgs, vendedores, gerentes);

- `carregaFuncoes()` - Esse é o principal método do programa, deve ler um arquivo chamado `funcoes.txt` e chamar cada uma das funções presentes nas linhas desse arquivo até a última linha.
- `carregarEmpresa()` - Lê o arquivo “`empresa.txt`” e carrega todos os dados da empresa (apenas: nome, faturamento, cnpj)
- `carregarAsg()` - Lê o arquivo “`asg.txt`” e carrega todos os dados desse arquivo no vector `asgs`.
- `carregarVendedor()` - Lê o arquivo “`vendedor.txt`” e carrega todos os dados desse arquivo no vector `vendedores`.
- `carregarGerente()` - Lê o arquivo “`gerente.txt`” e carrega todos os dados desse arquivo no vector `gerentes`.
- `carregaDono()` - Lê o arquivo “`dono.txt`” e carrega todos os dados desse arquivo em `dono`.
- `imprimeAsgs()` - Imprime todas os atributos de todos os `asgs`;
- `imprimeVendedores()` - Imprime todas os atributos de todos os `vendedores`;
- `imprimeGerentes()` - Imprime todas os atributos de todos os `gerentes`;
- `imprimeDono()` - Imprime todas os atributos de `dono`;
- `buscaFuncionario(int matricula);` - Caso não encontre deve imprimir a mensagem “Funcionário não encontrado no sistema”
- `calculaSalarioFuncionario(int matricula)` - Caso não encontre deve imprimir a mensagem “Funcionário não encontrado no sistema”
- `calculaTodoOsSalarios()` - Essa função deve salvar em um arquivo o resultado dos cálculos como o nome, cargo e salário de todos funcionários, salvando também o total de salários de `ASGs`, `VENDEDORES` e `GERENTES` e, por fim, a soma de todos os salários. Esses resultados também devem ser mostrados no console.
- `calcularRecisao(int matricula, Data desligamento)` - Calcula o valor da rescisão de um funcionário. Caso não encontre deve imprimir a mensagem “Funcionário não encontrado no sistema”

Os arquivos deverão ser modularizados (cada classe com seus arquivos `hpp` e `cpp`). A função `main()` deve conter apenas a criação dinâmica da Empresa e a chamada da função `carregaFuncoes()` seguindo o modelo abaixo:

```
Empresa *atacadoDosCalcado = new Empresa("Atacado dos Calcados",  
"40.101.588/0001-98", 156289.56 );
```

```
atacadoDosCalcado->carregaFuncoes();
```

Todas as funções que leem arquivos deverão ter tratamento de erros para lidar com possíveis erros nesse processo. Caso ocorra uma exceção imprima a mensagem "O arquivo

OBS: Não serão aceitas submissões de arquivos fora do sigaa, por meio de links ou fora do prazo estabelecido na atividade.