

## Lista 05 – Herança

1. Crie uma classe Pessoa com os atributos nome, CPF, endereço e data de nascimento. Esta classe deve possuir métodos para atualizar os dados (métodos modificadores) e retornar os valores necessários (métodos seletores).
2. Considere como subclasse da classe Pessoa, a classe Fornecedor. Considere que cada instância da classe Fornecedor tem, para além dos atributos que caracterizam a classe Pessoa, os atributos plafond (correspondente ao crédito máximo atribuído pelo fornecedor a determinada pessoa) e valorEmDivida (montante da dívida para com o fornecedor). Implemente na classe Fornecedor, para além dos usuais métodos seletores e modificadores, um método obterSaldo que devolve a diferença entre os valores dos atributos plafond e valorEmDivida. Escreva um programa de teste que permita verificar o funcionamento dos métodos implementados na classe Fornecedor e os herdados da classe Pessoa.
3. Considere como subclasse da classe Pessoa, a classe Empregado. Considere que cada instância da classe Empregado tem, para além dos atributos que caracterizam a classe Pessoa, os atributos numeroSecao, salarioBase (vencimento base) e INSS (percentagem retida para INSS). Implemente a classe Empregado com métodos seletores e modificadores e um método calcularSalario. Escreva um programa de teste para esta classe.
4. Considere como subclasse da classe Pessoa a classe Cliente. Considere que cada instância da classe Cliente tem, para além dos atributos que caracterizam a classe Pessoa, os atributos plafond (correspondente ao crédito máximo concedido ao cliente) e valorEmDivida. Implemente na classe Cliente, para além dos usuais métodos seletores e modificadores, um método obterSaldo que devolve a diferença entre os valores dos atributos plafond e valorEmDivida. Escreva um programa de teste para esta classe.
5. Implemente a classe Administrador como subclasse da classe Empregado. Um determinado administrador tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo ajudasDeCusto (ajudas referentes a viagens, estadias, ...). Note que deverá redefinir na classe Administrador o método herdado calcularSalario (o salário de um administrador é equivalente ao salário de um empregado usual acrescido das ajudas de custo). Escreva um programa de teste para esta classe.
6. Implemente a classe Operario como subclasse da classe Empregado. Um determinado operário tem como atributos, para além dos atributos da classe Pessoa e da classe

Empregado, o atributo `valorProducao` (que corresponde ao valor monetário dos artigos efetivamente produzidos pelo operário) e `comissao` (que corresponde à percentagem do `valorProducao` que será adicionado ao vencimento base do operário). Note que deverá redefinir nesta subclasse o método herdado `calcularSalario` (o salário de um operário é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste para esta classe.

7. Implemente a classe `Vendedor` como subclasse da classe `Empregado`. Um determinado vendedor tem como atributos, para além dos atributos da classe `Pessoa` e da classe `Empregado`, o atributo `valorVendas` (correspondente ao valor monetário dos artigos vendidos) e o atributo `comissao` (percentagem do `valorVendas` que será adicionado ao vencimento base do `Vendedor`). Note que deverá redefinir nesta subclasse o método herdado `calcularSalario` (o salário de um vendedor é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste para esta classe.

OBS.: Os programas de teste podem ser agrupados na função *main* de um arquivo teste.

Créditos:

Universidade da Madeira

DEPARTAMENTO DE MATEMÁTICA E ENGENHARIAS

PROGRAMAÇÃO ORIENTADA POR OBJECTOS

<http://www.jcarlosvale.com/wp-content/uploads/2009/11/ficha7.pdf>

```
struct endereco{
    string rua;
    int numero;
    string bairro;
    int CEP;
};

struct data{
    int dia;
    int mes;
    int ano;
};

class Pessoa {
    string nome;
    long int CPF;
    endereco end;
    data datanasc;

public:
    Pessoa(void);
    void setNome(string);
    void setCPF(long int);
    void setEndereco(endereco);
    void setDataNasc(data);
    string getNome(void);
    long int getCPF(void);
    endereco getEndereco(void);
    data getDataNasc(void);
};
```