UFPA – ICEN - FACULDADE DE COMPUTAÇÃO
Disciplina: Programação II CBCC

Aluno(a):_____Matrícula:_____

5ª LISTA DE EXERCÍCIOS (Polimorfismo em C++)

1) Crie uma classe chamada **Pessoa** que tenha como atributo protegido o nome da pessoa. Em seguida, crie duas outras classes chamadas **PessoaFisica** e **PessoaJuridica** que herdem da classe Pessoa.

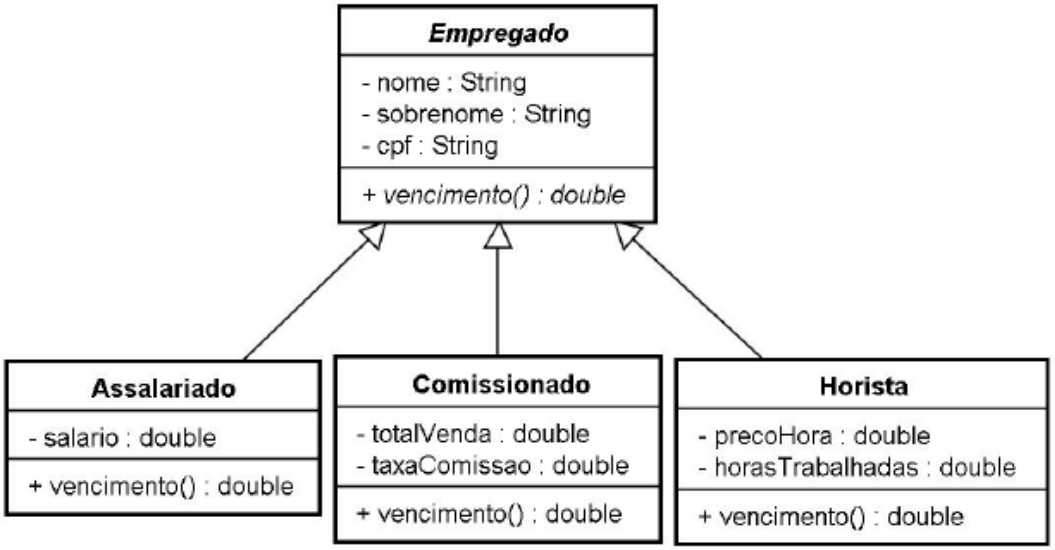
   A classe **PessoaFisica** terá como atributo privado o *CPF*, enquanto a classe **PessoaJuridica** terá como atributos privados o *CNPJ*, a *razão social* e o *nome fantasia*. Crie métodos *get, set* e *print* para todos os atributos das três classes.

   Crie uma classe chamada **Funcionario** que herda da classe **PessoaFisica**. Essa classe deverá ter como atributos privados a matrícula, o salário base do funcionário e a quantidade de horas trabalhadas no mês. Além disso, a classe terá um método público chamado *calculaSalarioBruto* que não terá nenhum parâmetro e deverá ser capaz de calcular e retornar o salario bruto através da seguinte equação: salarioBase x quantidadeHorasTrabalhadas. Por fim, crie métodos *get, set* e *print* para os atributos.

   Crie uma classe chamada **Cliente** herdeira da classe **PessoaFisica**. Essa classe deverá ter atributos privados que armazenem um telefone e um endereço. Crie métodos *get, set* e *print* para esses atributos.

   Crie uma classe chamada **Empresa** herdeira da classe **PessoaJuridica**. Essa classe deverá ter uma lista de funcionários e outra lista de clientes (pode ser vetor). Crie métodos para adicionar funcionários e clientes. Crie um método para imprimir matrícula, nome e salário bruto dos funcionários e outro para imprimir nome, telefone e endereço dos clientes. Crie também um método chamado *calcularFolhaDePagamento* que deverá calcular o salário bruto de todos os funcionários e retornar o total a ser pago aos funcionários.

2) Implemente a hierarquia de herança múltipla definida pelo diagrama UML abaixo:



3) Mostre o que será impresso pelos seguintes programas em C++. Em quais trechos dos códigos ocorrem polimorfismo?

a) Prog1.cpp

```
01: class A {
02:   public:
03:     virtual void f() { cout << "A::f()\n"; }
04:     void g() { cout << "A::g()\n"; }
05: };
06: class B: public A {
07: public:
08:     void f() { cout << "B::f()\n"; }
09:     void g() { cout << "B::g()\n"; }
10: };
```

```
11: int main() {
12:   B b;
13:   B *bp = &b;
14:   A *ap = &b;
15:   ap->f();
16:   ap->g();
17:   b.f();
18:   b.g();
19: }
```

b) Prog2.cpp

```
01: class A {
02: public:
03:   int x;
04: };
05: class B1: public A {
06: public:
07:   void f() {
08:     cout << x << endl;
09:   }
10: };
11: class B2: public A {
12: public:
13:   void f() {
14:     cout << x << endl;
15:   }
16: };
```

```
17: class C: public B1, public B2 {
18: public:
19:   void g() {
20:     B1::x = 4;
21:     B2::x = 5;
22:     B1::f();
23:     B2::f();
24:   }
25: };
26: int main() {
27:   C c;
28:   c.g();
29: }
```