

Unidad Nº4: Herencia

Preguntas orientadoras

- 1) Explique con sus palabras el concepto de Herencia.
- 2) ¿Cuál es la relación entre los conceptos de clase, subclase y superclase?
- 3) ¿En el lenguaje C++, la derivación es una forma de expresar la relación ____?
- 4) Una clase derivada es una _____ de una clase base.
- 5) ¿Para qué se utiliza la palabra clave *protected*?
- 6) Los datos y miembros protegidos son completamente _____ para las clases derivadas.
- 7) Las subclases pueden _____ el comportamiento de su superclase.
- 8) ¿Puede una clase derivada hacer que la función pública de una clase base sea privada?

Ejercicios

- 1) Modele una clase denominada "polar", que represente a los números complejos en forma polar.

Nota: "polar" deriva de "complejo", la clase que usted programó en el ejercicio 1 de la unidad III. En el proyecto se debe reutilizar la implementación de la clase complejo mediante el archivo binario (extensión .o). La definición de la clase "polar" deberá contener la interfaz y los atributos mostrados en la imagen que sigue:

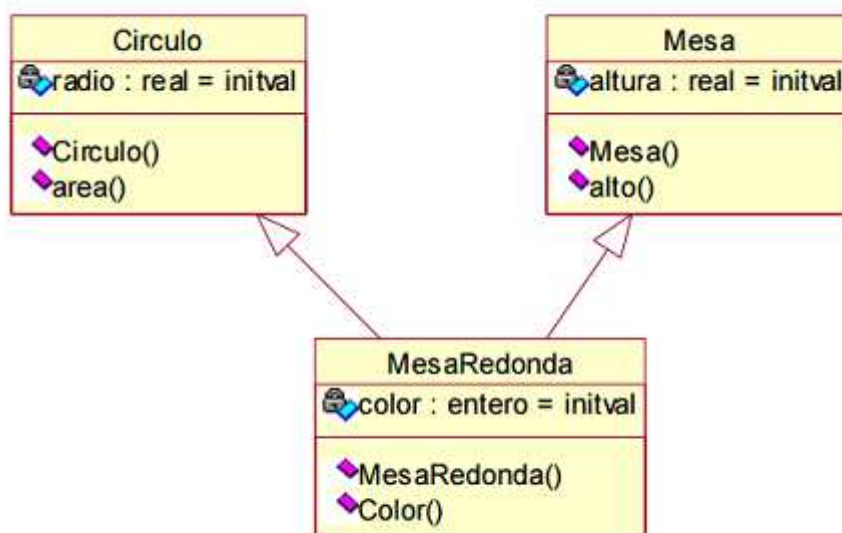
```

class polar : public complejo
{
public:
    polar();
    polar(float, float);
    float GetModulo();
    void SetModulo(float);
    float GetFase();
    void SetFase(float);
    float Modulo(complejo&);
    float Fase(complejo&);
    float Modulo();
    friend complejo Rectang(float, float);
    void operator= (complejo);
protected:
private:
    float fModulo;
    float fFase;
};

```

2) Implemente la clase "becario" vista en teoría. Analice todas las ambigüedades que puedan surgir al escribir el programa que prueba dicha clase.

3) Implemente en lenguaje C++ el siguiente diagrama UML.



4) Encuentre diferentes maneras de corregir el error del siguiente programa para que imprima el valor 10:

```

#include <iostream>
using namespace std;

class ClaseA {
public:
    ClaseA() : valorA(10) {}
    int LeerValor() const { return valorA; }
protected:
    int valorA;
};

class ClaseB {
public:
    ClaseB() : valorB(20) {}
    int LeerValor() const { return valorB; }
protected:
    int valorB;
};

class ClaseC : public ClaseA, public ClaseB {};

int main() {
    ClaseC CC;

    cout << CC.LeerValor() << endl;

    return 0;
}

```

5) Se desea modelar una Empresa con empleados. Una empresa conoce a todos sus empleados, y estos pueden ser de planta permanente o temporaria, además hay gerentes, que también son empleados de planta permanente, pero siguen un régimen salarial particular.

Cuando un empleado es de planta permanente cobra la cantidad de horas trabajadas por \$30, más antigüedad (\$10 por año de antigüedad), más salario familiar. Cuando es de planta temporaria, no cobra antigüedad y cobra la cantidad de horas trabajadas por \$20, más salario familiar. El salario familiar es \$20 por cada hijo, los empleados casados además cobran \$10 por su esposa/o. Un gerente cobra de manera similar a un empleado de planta permanente pero su hora trabajada vale \$40, por antigüedad se le pagan \$15 por año, mientras que el salario familiar es el mismo que el de los empleados de planta permanente y temporal.

a) Realizar un diagrama de clases.

b) Implementar el método `#montoTotal` en la clase `Empresa`, que retorna el monto total que la empresa debe pagar en concepto de sueldos a sus empleados (Definir e implementar todas las clases y métodos necesarios).