

Ejercicios Propuestos

A. Describa en UML la clase y dos objetos, para los siguientes objetos concretos y abstractos:

- 1) Numero Entero.
- 2) Numero Real
- 3) Numero Quebrado
- 4) Numero Binario Real
- 5) Numero Hexadecimal Real
- 6) Vector
- 7) Matriz
- 8) Cadena
- 9) Serie Fibonacci
- 10) Sumatoria = $1 + X + X^2 + X^3 + X^4 + \dots + X^N$
- 11) Programa de computadora
- 12) Función matemática
- 13) Ecuación de segundo grado
- 14) Conjunto Matemático
- 15) Ventana de Windows
- 16) Archivo de datos
- 17) Registro
- 18) Alumno
- 19) Docente
- 20) Computadora
- 21) Impresora
- 22) Lista de numeros
- 23) Pila
- 24) Cola
- 25) Cola Circular
- 26) Lista Enlazada
- 27) Lista Doblemente enlazada
- 28) Mouse de computadora
- 29) Tarjeta madre
- 30) Disco duro
- 31) Monitor
- 32) Automóvil
- 33) Figura geométrica
- 34) Reptil
- 35) Ave
- 36) Insecto

- 37) Pez
- 38) Fecha
- 39) Diccionario
- 40) Disco Compacto
- 41) Algoritmo computacional
- 42) Hospital
- 43) Universidad
- 44) País
- 45) Mesa
- 46) Bolígrafo
- 47) Polinomio
- 48) Integral
- 49) Circuito eléctrico
- 50) Componente eléctrico

B. Proponga tres ejemplos de objetos del mundo real:

- Para cada uno de ellos, determine la clase a la que pertenecen.
- Asóciele a cada clase un identificador descriptivo adecuado.
- Enumere varios atributos y operaciones para cada una de las clases.
- Represente gráficamente las clases utilizando la notación UML.
- A partir de los diagramas UML, escriba el código necesario para definir las clases utilizando el lenguaje de programación Java.

C. Rellene los huecos en las siguientes afirmaciones:

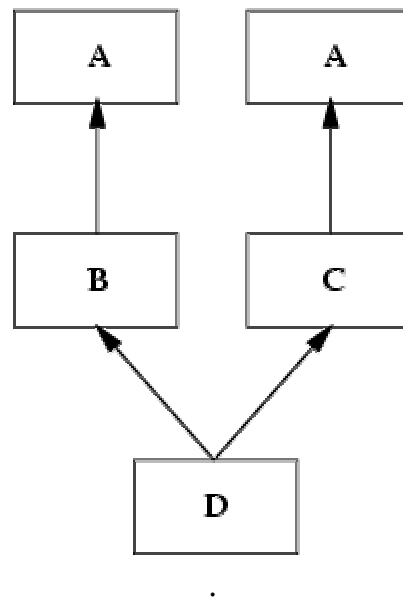
1. Los objetos encapsulan _____ y _____.
2. Los objetos se comunican entre sí pasándose _____.
3. Para comunicarse con un objeto concreto, no es necesario conocer su _____, basta con saber cuál es su _____.
4. Pueden existir varios tipos de relaciones entre clases: _____, _____ y _____.
5. Los lenguajes de programación orientada a objetos utilizan relaciones de _____ para derivar nuevas clases a partir de clases base.
6. _____ define una notación gráfica estándar para representar diseños orientados a objetos.
- 7.

D. ¿Cómo es posible crear objetos de una clase cuyos constructores son todos privados?

1. Definiendo un método estático público en la clase que cree un objeto de la clase y lo devuelva.
2. Definiendo una subclase y declarando públicos los constructores heredados.
3. Definiendo una superclase con constructores públicos.
4. No es posible.

E. Define la clase *Rectangle* por herencia a partir de la clase *Point*. El punto debería indicar la esquina superior izquierda del rectángulo. ¿Cuáles son los atributos de tu clase ? ¿Qué métodos incluiría ?

F. Herencia múltiple.



¿Qué conflictos de nomenclatura pueden ocurrir ? Trata de definir casos jugando con ejemplos de clases simples.

G. Emplea la siguiente estructura, para implementar un programa que manipule números complejos:

```
class Complex {  
    double _real,  
           _imag;  
  
public:  
    Complex() : _real(0.0), _imag(0.0) {}  
    Complex(const double real, const double imag) :  
        _real(real), _imag(imag) {}  
  
    Complex add(const Complex op);  
    Complex mul(const Complex op);  
    ...  
};
```