Unidad Nº5: Polimorfismo

Preguntas orientadoras

1) ¿Qué es el polimorfismo?

2) Según lo que se vió en clase, ¿por qué no hacer que todas las funciones de una clase sean virtuales?

3) ¿Qué es el enlace dinámico?

4) ¿Qué es una tabla v o VTABLE y para qué sirve?

5) ¿Qué es un destructor virtual?

6) ¿Qué ocurre si se está dentro de un constructor y se llama a una función virtual?

7) ¿Cómo se puede crear un constructor virtual de copia?

8) Si una clase base declara una función como virtual, y una clase derivada no utiliza el término virtual cuando redefina esa función, ¿seguirá siendo virtual cual la herede una clase de tercera generación?

9) ¿Por qué es malo cambiar el tipo de un objeto en tiempo de ejecución?

10) ¿Qué entiende por upcasting y downcasting?

11) ¿Investigue sobre el Object slicing?

12) ¿Cómo se puede prevenir el Object slicing?

13) ¿Qué significa filtrar la funcionalidad de manera ascendente?

14) ¿Es bueno utilizar siempre la filtración ascendente?

15) Si un rectángulo "redondo" tiene bordes rectos y esquinas redondeadas, y su clase *RectRedondo* hereda tanto de *Rectangulo* como de *Circulo*, y éstos a su vez heredan de *Figura* ¿cuántas *Figuras* se crearán cuando cree un *RectRedondo*?

Ejercicios

1) Escriba un programa que derive un *Auto* y un *Camión* de la clase *Vehículo*. Convierta a Vehículo en un ADT que tenga dos funciones virtuales puras. Haga que *Auto* y *Camión* no sean ADTs.

2) Modifique el programa del iniciso 1 para que *Auto* sea un ADT, y derive de *Auto* a *AutoDeportivo*, *Furgoneta* y *Sedan*. En la clase *Auto*, proporcione una implementación para una de las funciones virtuales puras de *Vehículo* y hágala no pura.

3) Modele una clase que represente una cuenta bancaria que tenga un saldo y un titular, y que pueda realizar operaciones de extracciones y depósitos. Una vez modelada la clase Cuenta, modele la clase Banco que contiene un conjunto cuentas, e implemente el mensaje que le permita saber cuál es el activo disponible del Banco. Es decir, la suma de los saldos de las cuentas. Existen dos tipos de cuentas bancarias: Cuentas corrientes y Cajas de ahorro.

Si revisamos el comportamiento nos encontraremos con las siguientes características en común:

- Ambas llevan cuenta de su saldo.
- Ambas permiten realizar depósitos.
- Ambas permiten realizar extracciones.

Pero cada una tiene un tipo de restricción distinto en cuanto a las extracciones:

- Cuentas corrientes: permiten que el cliente gire en descubierto (con un tope pactado con cada cliente).
- Cajas de ahorro: poseen una cantidad máxima de extracciones mensuales (para todos los clientes). No se permite girar en descubierto.
- Cuenta universitarias: permite extracciones de no más de 100$ diarios.

Modele estas clases y los métodos correspondientes.

4) Se pide que implemente el "calculador impositivo" con el que una persona puede calcular su impuesto. El calculador funciona así:

El usuario configura (inicializa) el calculador indicando su nombre y su categoría.

El usuario carga, una a una, las facturas de cada servicio que ha brindado, incluyendo detalle del servicio, número de factura y monto.

El usuario pide al calculador que calcule el impuesto para todas las facturas cargadas cuyos número se encuentren entre dos que él indica como parámetros.

Como resultado, se obtiene un objeto ReciboDePago que entiende los mensajes nombreDelContribuyente(), montoTotal(), montoFijo(), montoVariable(), primeraFacturaConsiderada(), ultimaFacturaConsiderada().

El impuesto se calcula así:

Todos los contribuyentes abonan un monto fijo por pago. Dicho monto es idéntico para todos y se actualiza regularmente. Inicialmente, el monto es de $5.

Además, cada contribuyente paga un monto variable, relativo a sus ingresos, de acuerdo a la categoría en la que se inscribió el contribuyente. Los contribuyentes inscriptos como "Limitado" son los que menos pagan. Ellos pagan un 0,5% de lo facturado. Los contribuyentes inscriptos en categoría "Completo" abonan un 50% por cada servicio facturado cuyo monto supere los $5000. Los que más pagan son los "Extendidos". Ellos pagan igual que los Completos, pero a eso se agrega $10 por cada servicio facturado.

1) Implemente en C++ todas las clases necesarias para proveer la funcionalidad antes descripta.

2) Construya un programa de test que demuestre:

1. Correcta inicialización del sistema.

2. Funcionamiento correcto de la carga de facturas.

3. Funcionamiento correcto de la generación de recibo de pago.

3) Construya un diagrama de clases UML donde quede claro que clases componen el sistema, de qué forma se relacionan, herencia, y atributos.