

PROGRAMACIÓN ORIENTADA A OBJETOS
PRACTICA # 7-8: Herencia y Polimorfismo

Objetivo General

- (7) Implementar los conceptos de herencia en un lenguaje de programación orientado a Objetos
- (8) Implementar el concepto de polimorfismo en un lenguaje de programación orientado a objetos.

Objetivo de clase

Realizar aplicaciones sencillas donde el estudiante pueda visualizar de manera clara los conceptos relacionados con herencia como clases abstractas e interfaces así como la importancia del polimorfismo y la sobreescritura de métodos

ACTIVIDADES PARA EL DESARROLLO DE LA PRÁCTICA

Ejercicio 1. Construyendo una jerarquía de Herencia,

Elabora un programa sencillo para la administración de una tienda de tecnología.

En el programa habrá una lista de productos disponibles para el usuario donde podrá ver las características principales y el precio de venta.

Habrán tres tipos de compradores para el programa, clienteVip, socio y estudiante

Al ingresar el programa, éste le preguntará al usuario qué tipo de comprador es, y dependiendo de la opción seleccionada accederá a productos dependiendo de su perfil

El usuario podrá navegar por la lista disponible y comprar algún producto

Las clases que tendrá el programa para utilizar al menos son las siguientes

Clases:

Comprador	Dispositivo Móvil
ClienteVip	Laptop
Estudiante	PC
Celular	Smartphone
Computadora	Tablet
Dispositivo Electrónico	Television

- Se pueden agregar clases adicionales, pero es obligatorio que aparezcan las que se muestran
- En la solución realizada se deberá considerar al menos una clase abstracta y al menos una interfaz (pueden ser más si el alumno lo considera)
- Se deberán establecer las jerarquías de herencia para las clases señaladas
- Codifica las clases con los atributos y métodos que consideres convenientes implementando herencia
- Todas las clases deberán utilizar un constructor diferente al constructor por defecto
- El diseño de la solución es libre y el alumno podrá diseñar el programa como mejor le convenga siempre y cuando establezca la jerarquía de herencia correctamente.
- En el reporte se debe incluir el análisis de la solución propuesta así como agregar las capturas de pantalla de las diferentes funcionalidades y opciones del programa

Ejercicio 2. Polimorfismo - Creando una colección de objetos

Elabora un programa sencillo en el cual se pueda establecer una jerarquía de herencia para almacenar una colección de “Animales”. El diseño de los objetos y de la jerarquía de herencia es libre, pero las clases descendientes deberán sobreescibir al menos 3 métodos de la clase padre.

El uso de clases abstractas, paquetes, modificadores de acceso no es obligatorio, pero si se implementa, se deberá realizar correctamente

Se deberá realizar un programa en el cual, en la clase principal se pueda crear una colección de animales, “comportándose” de maneras diferentes. El programa mostrará un menú de usuario con las siguientes opciones

- Agregar animal (en este caso el programa le mostrará al usuario un “catálogo de animales” para seleccionar el que se agregará a la colección)

- Mostrar un elemento de la colección (El usuario seleccionará una posición, lo cual mostrará el comportamiento del animal que ocupa esa posición)

- Mostrar la colección completa

- Salir

Nota: Se pueden agregar al programa o al menú de usuario, otros elementos que el alumno considere convenientes

Ejercicio Opcional – Para puntos extra

Codifica el programa realizado en el ejercicio de clase implementando el concepto de herencia en Vehículo, automóvil y auto compacto. Incluyendo las clases que consideraste como abstractas

Elabora un menú de usuario para crear, mostrar, y modificar vehículos

Nota: En este ejercicio no es necesario realizar un programa complejo con muchas opciones, el objetivo es codificar lo realizado en los ejercicios de clase y observar si ese diseño fue correcto o sufrió muchas modificaciones al llevarlo a la práctica

CONCLUSIONES

Escribe las conclusiones de tu práctica en función del cumplimiento de los objetivos y del aprendizaje de los temas de la práctica.