

Paradigmas de Programación

Práctica 2. Programación Orientada a Objetos (POO)

Objetivo:

Realizar la implementación del contenido de la Unidad 2 correspondiente a POO, mediante una aplicación en el lenguaje de programación Java.

La aplicación se puede llevar a cabo mediante algún IDE (Integrated Development Environment) como Netbeans, Eclipse, etc.; la práctica también puede ser realizada en consola.

Desarrollo:

La aplicación debe incluir los siguientes temas.

1. Clases
 - Encapsulamiento (uso de acceso *private*).
 - Constructores.
 - Sobrecarga de métodos.
 - Objetos.
 - Uso de *static*
2. Agregación de clases
 - Uso de objetos de clases agregadas
3. Jerarquía de clases
 - Jerarquía de clases de al menos tres niveles (no se acepta herencia en clases de la API)
 - Redefinición de métodos
 - Uso de *super*.
 - Uso de objetos de la jerarquía de clases

Sugerencias para la práctica 2:

- a) Escenario de un juego de mesa como el ajedrez: implementación de roles a partir de una jerarquía de clases para cada personaje (reina, rey, caballo, alfil, etc.), manejo de excepciones para validar los movimientos de las piezas del ajedrez.
- b) Gestión de Bancos: Modelar un banco con clientes y con diversos tipos de cuentas bancarias. Tener un registro de clientes con fotografía y datos personales como nombre, dirección, antigüedad como cliente del banco. Simular cajeros automáticos donde los clientes hacen transacciones que se conectan a la casa matriz (el banco). Usar una interfaz gráfica para hacer interacciones con el cajero automático. Hacer una interfaz gráfica para el banco en donde se pueda seleccionar a un cliente y aparezca su fotografía y datos personales así como generar reportes de cada cliente acerca del historial de transacciones bancarias hechas durante algún período establecido. Manejo de excepciones para validar movimientos de los clientes en sus cuentas.
- c) Control escolar: Modelar una universidad con alumnos (modalidad: presencial, a distancia, etc.), docentes (modalidad: titulares, invitados, medio tiempo), personal administrativo (encargados de laboratorio, analistas de datos, etc.). Usar clases como: materia (unidad de aprendizaje), grupo, calificación, etc. Realizar las operaciones de: inscribir un alumno a una materia y/o a un grupo, asignar profesor a un grupo, asignar materia a un profesor, reportar alumnos de un grupo, asignar calificación a un alumno, reportar calificaciones de un alumno, reportar grupos de un profesor, etc. validar datos personales con excepciones.

Nota: El proyecto **NO** deberá hacer uso de Bases de Datos; todo el diseño deberá contemplar el paradigma Orientado a Objetos.

Tabla de Evaluación:

Concepto	Valor
Diagrama UML (el diagrama debe corresponder con el código programado)	1
Clases: - Encapsulamiento (uso de acceso <i>private</i>) - Constructores - Sobrecarga de métodos - Uso de objetos - Uso de <i>static</i>	4
Agregación de clases - Clases que contienen a otras clases - Uso de constructores - Creación de objetos	2
Jerarquía de clases: - Diseñada por el estudiante (no será válido presentar jerarquía de API de Java). - Redefinición de métodos - Herencia simple - Uso de <i>super</i> en constructores, métodos o campos. - Al menos tres niveles de jerarquía - Uso de acceso <i>protected</i> .	3

Presentación de la práctica 2:

- Presentar el programa en ejecución.
- **Presentar un escenario de prueba para cada aspecto a evaluar, de forma que se cubran todos los requerimientos de esta práctica. No se revisará el proyecto si no se tiene el escenario de prueba.**
- Sustentar un breve examen oral acerca del código presentado y del programa en ejecución.
- Presentar individualmente o en equipo (máximo dos integrantes). Si la práctica se entrega en equipo, deberá ser el mismo equipo que entregó la práctica 1. No hay cambio de equipo.
- Prácticas copiadas serán canceladas.

Plazo de entrega:

Semana del martes 1 de junio al lunes 7 de junio del año en curso.

En la semana previa se establecerá el calendario de entrega en días y horarios.

No hay prórrogas bajo ninguna circunstancia, cuentan con el tiempo suficiente para hacer la entrega en tiempo y forma.