

NOMBRE DE LA ASIGNATURA		Programación de Computadores II						
NOMBRE DE LA ACTIVIDAD		Implementando herencia y polimorfismo en Java						
TIPO DE ACTIVIDAD		Sincrónica		Asincrónica	x	Individual	x	Grupal
TEMÁTICA REQUERIDA PARA LA ACTIVIDAD				OBJETIVOS				
Unidades 1 y 2 del plan de asignatura  Unidad 3. Herencia, polimorfismo e interfaces				Identificar de Superclases y subclases en problemas planteados.  Representar el mecanismo de herencia entre clases mediante UML  Solucionar problemas básicos mediante la implementación de herencia en Java.				
COMPETENCIAS				INSUMOS PARA EL DESARROLLO DE LA ACTIVIDAD / REFERENCIAS BIBLIOGRÁFICAS				
<ul style="list-style-type: none"> <li>Implementa los mecanismos de herencia, polimorfismo e interfaces para mejorar la estructura de soluciones de software orientadas a objetos</li> </ul>				<ul style="list-style-type: none"> <li>Material educativo de la “Unidad 1”</li> <li>Material complementario de la “Unidad 1”</li> <li>Material educativo de la asignatura “Unidad 2”</li> <li>Material complementario de la “Unidad 2”</li> <li>Material educativo y complementario de la “Unidad 3.</li> </ul>				
CONOCIMIENTOS PREVIOS REQUERIDOS								
Implementación de clases y objetos en Java								
ESPECIFICACIONES DE LA ACTIVIDAD								
<p><b>Actividad:</b> Utilizando el enfoque orientado a objetos y el mecanismo de herencia, implemente aplicaciones en Java utilizando Netbeans IDE para los problemas básicos que se formulan en la presente actividad.</p> <p>Para cada problema se debe diseñar el respectivo diagrama de clases UML, para representar las clases, atributos, métodos y relaciones (incluyendo herencia)</p> <p>Así mismo, se requiere que el código fuente de los programas desarrollados se almacenen en un repositorio GitHub.</p> <p><b>Problemas propuestos:</b></p> <ol style="list-style-type: none"> <li>Nos piden hacer que gestionemos una serie de productos. Todo producto tiene los siguientes atributos: nombre y precio. Sin embargo, solo tenemos dos tipos de productos: Perecedero: tiene un atributo llamado días a caducar, y No perecedero: tiene un atributo llamado iva.</li> </ol>								

Crea sus constructores, getters, setters y toString. Tendremos una función llamada calcular, a la cual le pasaremos un numero como parámetro, correspondiente a la cantidad de productos. Esta función, permitirá lo siguiente, según cada tipo de producto:

- En los Perecederos, reducirá el precio según los días a caducar:
  - Si le queda 1 día para caducar, se reducirá 4 veces el precio final.
  - Si le quedan 2 días para caducar, se reducirá 3 veces el precio final.
  - Si le quedan 3 días para caducar, se reducirá a la mitad de su precio final.
- En productos No Perecedero, aplicara el iva al precio.

En una clase principal, crea un array de productos y muestra el precio total de vender 5 productos de cada uno. Crea tú mismo los elementos del array.

2. Una agencia de renta de vehículos dispone de dos tipos de vehículos: autobuses y tractores. La renta de autobuses se factura por kilómetros. Debido a esto, los datos de la clase Autobús son: el precio por kilómetro, la cantidad de kilómetros que tiene el autobús cuando se renta, la cantidad de kilómetros que tiene el autobús cuando se devuelve. En cambio, la renta de tractores se factura por días. Debido a esto, los datos de la clase Tractor son: el precio por día, la fecha de su renta, la fecha de su devolución.

Cuando se rente un vehículo, se deben capturar sus placas, sus datos correspondientes (de acuerdo al tipo de vehículo), marcarlo como alquilado (asignando el valor booleano true a la propiedad respectiva). Cuando se devuelve un vehículo, se elimina la marca de alquilado (asignando el valor booleano false a la propiedad correspondiente), se calcula el importe a pagar por la renta y se muestran sus datos por consola.

3. Un banco requiere controlar las cuentas de sus clientes y para ello las clasifica en dos: cheques y ahorros. Todas las cuentas del banco tienen los siguientes datos: número de cuenta (entero), nombre del cliente (cadena), saldo (numérico real). Además, se realizan las siguientes operaciones con ellas:

- Consultar datos: Mostrar sus propiedades.
- Depositar: Incrementa el saldo con la cantidad de dinero que se deposita.
- Retirar: Antes de hacer el retiro, se debe verificar la suficiencia de saldo y en caso de aprobarlo, se disminuye el saldo.

Las cuentas de ahorros presentan las siguientes características:

- Fecha de vencimiento.
- Porcentaje de interés mensual.
- Método para depositar los intereses el primer día de cada mes.
- Solamente se puede retirar dinero el día de la fecha de vencimiento.

Las cuentas de cheques presentan las siguientes características:

- Comisión por uso de chequera.
- Método para calcular la comisión por emisión de cheques, la cual se descuenta directamente del saldo.

Elabore el diagrama de clases en UML e implemente dicho sistema con un proyecto. No se permiten componentes duplicados en las clases ni clases vacías (sin elementos).

4. Desarrolle un programa que permita hacer algunas manipulaciones básicas sobre círculos y rectángulos. Estas manipulaciones incluyen conocer la posición x e y del centro de la figura en un sistema de coordenadas cartesianas, mover la figura a otra posición x e y, así como redimensionar la figura. Además, se quiere calcular el área de cada una de las figuras y en el caso del rectángulo intercambiar altura por anchura.

También se pretende que cada figura responda al mensaje toString devolviendo una descripción de sus atributos.

Implemente un método main que pida al usuario introducir mediante teclado la información necesaria para crear un objeto Circulo y un objeto Rectángulo. En segundo lugar, el programa realizará las siguientes manipulaciones sobre las figuras creadas:

- Incrementará en una unidad la coordenada x y la coordenada y que definen la ubicación de cada figura.
- Pedirá al usuario que introduzca por teclado la información necesaria para redimensionar ambas figuras.
- Intercambiará el ancho por la altura del rectángulo.

El programa deberá mostrar por pantalla toda la información relativa a las figuras tanto antes como después de efectuar todas las manipulaciones.

5. Declare una clase VehiculoMotorizado que sirva como clase padre para vehículos de tipo Motocicleta, Automóvil y Camión. Todos los vehículos poseen un fabricante, modelo, año de fabricación y kilometraje. Los automóviles son de distintos estilos y las motocicletas se dedican a usos determinados. A su vez, los camiones pueden tener uno o varios remolques.

Las clases Automóvil y Camión deben implementar un método que determine si se cumplen las medidas de seguridad exigidas. Se considera que los automóviles y camiones cumplen las medidas de seguridad cuando su número de pasajeros no supera el número máximo de pasajeros autorizados.

Cree una clase llamada CapacidadLimite que debe incluir como atributos de clase constantes para el límite de pasajeros admitidos en automóviles y camiones. Los límites para automóviles deben incluir un límite de pasajeros para automóviles estilo "VAN" y otro límite de pasajeros para automóviles de cualquier otro estilo.

Con esta estructura de clases escriba un programa principal que construya un objeto de clase automóvil, motocicleta o camión según decisión del usuario. El programa deberá utilizar la clase Scanner para leer la información introducida por el usuario mediante teclado y finalmente imprimirá la información del vehículo considerado.

#### Lectura recomendada

- Consulta en la fuente bibliográfica de tu preferencia el concepto de **upcasting**, el usos de la instrucción **instanceof**

#### RECOMENDACIONES / OBSERVACIONES

Sin observaciones