

UNIDAD DOCE

ACTIVIDADES

Consulta en internet acerca de:

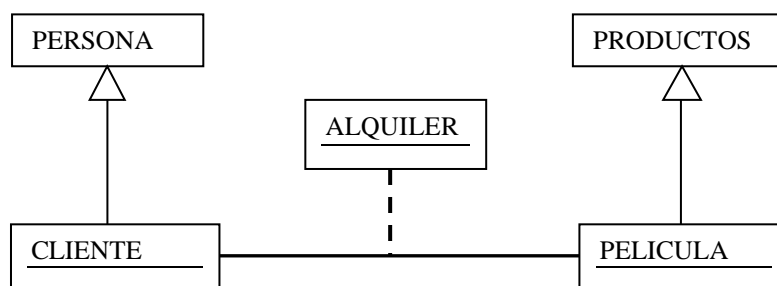
1. Agregación, composición, asociación de clases.
2. Metaclases en MOO. ¿Implementa Java este concepto?
3. Clases abstractas y jerarquía de Interfaces del paquete `java.util`
4. Recuerda las clases `Date`, `Calendar` y `GregorianCalendar` del paquete `java.util`. Utiliza estas clases en alguno de tus ejercicios.

Ejercicios

1. Se desea diseñar un programa para trabajar con figuras geométricas en el plano. El programa deberá mostrar en pantalla la información referente a la figura y calcular su área. Las figuras que se desea tratar son círculos, rectángulos y triángulos. Además se desea implementar una clase cuadrado como caso especial de rectángulo. A partir de la clase cuadrado, también deseamos derivar otra llamada `CuadradoEspecializado` que permita pintar un cuadrado con un carácter.

Implementa las clases necesarias y testéalas convenientemente.

2. Con la implementación de las clases del ejercicio 1, añadir las funcionalidades necesarias para realizar un programa que pinte en pantalla cuatro figuras de distintos tipos ordenadas de mayor a menor área.
3. Las clases asociación de un diagrama de clases UML son clases convencionales cuyo único papel consiste en relacionar objetos de otras clases, no tienen comportamiento propio. Dado el siguiente diagrama UML donde ALQUILER es una clase asociación:



Realizar un diseño e implementar las interfaces de cada una de las clases y verificarlas convenientemente.

Persona: propiedades comunes a cualquier conjunto de personas

Productos: Las características generales de los productos típicos de una tienda dónde podemos alquilar una película.

Cliente y Película: datos y funcionalidades interesantes para el ejercicio

Alquiler: Es una clase que relaciona cliente y película mediante una fecha.

4. Construir una clase final `Math2` que amplíe las declaraciones de métodos estáticos de la clase `Math` y que incluya funciones que devuelvan, respectivamente, el máximo, el mínimo, el sumatorio, la media aritmética y la media geométrica de un array de números reales dado como parámetro.

5. Construir una clase final **Math3** que amplíe las declaraciones de métodos estáticos de la clase **Math** y que implemente una interfaz llamada **Extremos** compilada con el siguiente código fuente:

```
public interface Extremos {
    int min(int [] a);
    int max(int [] a);
    double min(double [] a);
    double max(double [] a);
}
```

6. Cree una clase abstracta **Reloj** que permita obtener la hora actual del sistema y declare métodos abstractos para imprimir la hora en pantalla. El reloj debe cambiar la hora y establecer una alarma.

Cree una clase **RelojDigital** y una clase **RelojAnalogico** que hereden de la clase **Reloj**. Modifique el método de presentación de la hora en pantalla para adaptarlo a las necesidades de cada caso.

7. En un puerto se alquilan amarres para barcos de distinto tipo. Para cada **ALQUILER** se guarda el nombre y DNI del cliente, la fecha inicial y final de alquiler, la posición del amarre y el barco que lo ocupará.

Un **BARCO** se caracteriza por su matrícula, su eslora en metros y año de fabricación.

Un alquiler se calcula multiplicando el número de días de ocupación (incluyendo los días inicial y final) por un módulo función de cada barco (obtenido simplemente multiplicando por 10 los metros de eslora) y por un valor fijo (1.200 pesetas en la actualidad). Sin embargo ahora se pretende diferenciar la información de algunos tipos de barcos:

- número de mástiles para veleros
- potencia en CV para embarcaciones deportivas a motor
- potencia en CV y número de camarotes para yates de lujo.

El módulo de los barcos de un tipo especial se obtiene como el módulo normal más: el número de mástiles para veleros, la potencia en CV para embarcaciones deportivas a motor, la potencia en CV, más el número de camarotes para yates de lujo.

Utilizando la herencia de forma apropiada, diseñe el diagrama de clases y sus relaciones, con detalle de atributos y métodos necesarios. Programe en Java los métodos que permitan calcular el alquiler de cualquier tipo de barco.

8. Se quiere crear la jerarquía **ElementoGrafico** presentada en la figura. Deben crearse las clases necesarias para poder pintar por pantalla los elementos gráficos de la jerarquía. Para comprobar la funcionalidad proporcionada por esta jerarquía cree un array de objetos de la clase **ElementoGrafico** y haga que se muestren en pantalla.

