

## 7.H. Actividades propuestas.

Sitio: [Formación Profesional a Distancia](#)

Curso: Programación

Libro: 7.H. Actividades propuestas.

Imprimido por: Iván Jiménez Utiel

Día: lunes, 10 de febrero de 2020, 15:55

## Tabla de contenidos

- [1. ActividadUT07-1: Herencia con clase Punto.](#)
- [2. ActividadUT07-2: Clase abstracta ObjetoGeometrico.](#)
- [3. ActividadUT07-3: InterfazClase abstracta Geometria.](#)
- [4. ActividadUT07-4: Clase abstracta Figura.](#)
- [5. ActividadUT07-5: Clase Vehiculo.](#)

## 1. ActividadUT07-1: Herencia con clase Punto.

Crear una [clase](#) **Punto** para describir un punto en el plano, con atributos de tipo double relativos a las coordenadas X e Y de dicho punto.

Además dispondrá de un constructor general que reciba como parámetros los dos atributos anteriores, y un constructor de copia.

Incorporar métodos getters y setters.

Crear una [clase](#) **PuntoMejor** que extienda de la [clase](#) **Punto** y que disponga de un [método](#) de tipo estático para determinar la distancia entre dos puntos (documentate para determinar la fórmula matemática a usar). Dicho [método](#) recibirá como parámetros dos objetos de la [clase](#) **Punto**, entre los cuales querremos hallar la distancia.

Incorporar un nuevo [método](#) de [objeto](#), que calcule la distancia entre 2 puntos. Dicho [método](#) recibirá como parámetro un [objeto](#) **Punto**. La distancia se calculará entre ese punto y el punto que llame al [método](#).

Crear una [clase](#) **TestPunto** para probarlas.

## 2. ActividadUT07-2: Clase abstracta ObjetoGeometrico.

Crear una [clase](#) abstracta **ObjetoGeometrico**, la cual dispondrá de un único [atributo](#) llamado **nombre**. Además dispondrá de 2 métodos abstractos que serán **calcularArea()** y **calcularPerimetro()**. Además sobrescribirá el [método](#) **toString()** para mostrar el nombre del [objeto](#).

Crear una [clase](#) **Circulo** que extienda de la [clase](#) anterior. Su constructor incorporará un nombre y un double correspondiente al radio de dicho círculo. Crearemos también un constructor por defecto que cargue un nombre y el radio con valor 1. Además especificará los métodos abstractos de la [clase](#) padre según la fórmula correspondiente. También sobrescribirá su [método](#) **toString()** indicando el nombre y el radio.

Crear una [clase](#) **Prueba** para probar el ejercicio.

### 3. ActividadUT07-3: InterfazClase abstracta Geometria.

Crear una [interfaz](#) llamada **Geometria** con métodos **calcularArea()** y **calcularPerimetro()**.

Crear una [clase](#) **Circunferencia** con [atributo](#) **radio** y una [clase](#) **Rectángulo** con atributos **base** y **altura**. Ambas clases deben implementar la [interfaz](#) **Geometria**. Incorporar los constructores necesarios y los métodos getters y setters.

Crear una [clase](#) **TestGeometria** para probar el ejercicio.

#### 4. ActividadUT07-4: Clase abstracta Figura.

Diseña una [clase abstracta](#) denominada **Figura**, cuyos miembros de datos sean: **base** y **altura**. Además del constructor y los métodos set y get, la [clase](#) contará con un [método abstracto](#) que retorne el área de una figura. **Figura** contiene dos subclases denominadas **Rectangulo** y **Triangulo**. Escribe dos fuentes con el [método main](#). El primero de ellos declara dos objetos de tipo **Triangulo** y **Rectangulo** y visualiza en pantalla el área de la figura. El segundo empleará un array de referencias de tipo **Figura**.

## 5. ActividadUT07-5: Clase Vehiculo.

Diseña una [clase](#) denominada **Vehículo**. La información de un vehículo es: número de **ruedas** y **autonomía**. Además del constructor y los métodos **set()** y **get()**, tendremos un [método](#) denominado **verDatos()** que envíe a pantalla los datos de un vehículo. La [clase](#) **Vehículo** tiene dos subclases: **Coche** y **Camión**. Un coche además de la información de un vehículo almacena el **número de pasajeros**. Un camión tendrá además el **límite de carga**. Las subclases redefinen el [método](#) denominado **verDatos()** añadiendo información específica de ese tipo de vehículo. Crear una [clase](#) **TestVehiculo** para probar el ejercicio.