

**Practica 1**1. Desarrollar una clase llamada *Punto* que:

- Tenga dos atributos *private* de tipo *double*.
- Tenga un constructor con dos parámetros de tipo *double* que inicialice los dos atributos.
- Tenga un constructor por defecto (sin parámetros) que inicialice los dos atributos al valor que se quiera.
- Tenga un *getvalor* para cada uno de los atributos.
- Tenga un método *calcularDistanciaDesde* que recibe un parámetro de tipo *Punto* y que devuelve un *double*.

2. Desarrollar una clase llamada *Circulo* que:

- Tenga dos atributos *private*:
  - El centro del círculo: *Punto*
  - El radio del círculo : *double*
- Tenga un constructor con dos parámetros de tipo *Punto* y *double* que inicialice los dos atributos.
- Tenga un constructor por defecto (sin parámetros) que inicialice los dos atributos al valor que se quiera.
- Tenga un constructor con tres parámetros de tipo *double* que inicialice los dos atributos.
- Tenga un *getvalor* para cada uno de los atributos.
- Tenga un método *calcularDistanciaDesde* que recibe un parámetro de tipo *Punto* y que devuelve un *double*.
- Tenga un método *calcularArea* que no recibe ningún parámetro y devuelve un *double*.
- Tenga un método *calcularPerímetro* que no recibe ningún parámetro y devuelve un *double*.

3. Desarrollar una clase llamada *Triangulo* que:

- Tenga tres atributos *private* de tipo *Punto*.
- Tenga un constructor con tres parámetros de tipo *Punto* que inicialice los tres atributos.
- Tenga un constructor por defecto (sin parámetros) que inicialice los tres atributos al valor que se quiera.
- Tenga un constructor con seis parámetros de tipo *double* que inicialice los tres atributos.
- Tenga un *getvalor* para cada uno de los atributos.

- Tenga un método `calcularDistanciaDesde` que recibe un parámetro de tipo `Punto` y que devuelve un `double`.
  - Tenga un método `calcularArea` que no recibe ningún parámetro y devuelve un `double`.
  - Tenga un método `calcularPerimetro` que no recibe ningún parámetro y devuelve un `double`.
4. Desarrollar una clase llamada `Practica_1` que en su método `main`:
- Cree e inicialice dos objetos de la clase *Punto* y muestre la distancia entre ambos.
  - Cree un objeto de la clase *Circulo* y muestre su área, perímetro y distancia a uno de los dos puntos creados al comienzo.
  - Cree un objeto de la clase *Triangulo* y muestre su área, perímetro y distancia a un nuevo punto.

Nota 1: No seremos rigurosos con las matemáticas y supondremos que los triángulos siempre son rectángulos.

**Practica 2**

1. Desarrollar una clase llamada Asignatura que:

- Tenga dos atributos `private` de tipo `int` (el identificador) y de tipo `double` (la calificación).
- Tenga un constructor con un parámetro de tipo `int`.
- Tenga un `getvalor` para cada uno de los atributos.
- Tenga un `setvalor` para la calificación.

2. Desarrollar una clase llamada Alumno que:

- Tenga tres atributos `private` de tipo `Asignatura`.
- Tenga un constructor con tres parámetros de tipo `Asignatura` que inicialice los tres atributos.
- Tenga un constructor con tres parámetros de tipo `int` que inicialice los tres atributos.
- Tenga un `getvalor` para cada uno de los atributos.

3. Desarrollar una clase llamada Profesor que:

- Tenga un método `ponerNotas` que recibe un parámetro de tipo `Alumno` y que no devuelve nada. Pondrá una calificación aleatoria a cada una de las asignaturas del alumno.
- Tenga un método `calcularMedia` que recibe un parámetro de tipo `Alumno` y devuelve un `double`.

4. Desarrollar una clase llamada `Practica_2` que en su método `main`:

- Cree e inicialice tres `Asignaturas`
- Cree un `Alumno` con las tres `Asignaturas`.
- Cree un `Profesor` que le ponga calificaciones al `Alumno` y muestre por pantalla la media del `Alumno`.

**Practica 3**

5. Desarrollar una clase llamada Motor que:

- Tenga dos atributos `private` de tipo `int` (litros de aceite) y de tipo `int` (CV)
- Tenga un constructor con un parámetro de tipo `int` para los CV. Los litros de aceite por defecto serán 0.
- Tenga un `getvalor` para cada uno de los atributos.
- Tenga un `setvalor` para los litros.

6. Desarrollar una clase llamada Coche que:

- Tenga un atributo `private` de tipo `Motor`, un atributo de tipo `String` (marca), otro de tipo `String` (modelo) y otro de tipo `double` con el precio acumulado con las averías.
- Tenga un constructor con dos parámetros de tipo `String` que inicialice la marca y el modelo.
- Tenga un `getvalor` para cada uno de los atributos.
- Un método `acumularAveria` que incrementará el importe gastado en averías.

7. Desarrollar una clase llamada Garaje que:

- Tendrá tres atributos, un coche, un `String` con la avería asociada y el número de coches que ha ido atendiendo.
- El garaje solo podrá atender a un coche en cada momento. Controlar esta premisa.
- Tenga un método `acceptarCoche` que recibe un parámetro de tipo `Coche` y la avería asociada. El garaje solo podrá atender a un coche en cada momento. Si ya está atendiendo uno, que devuelva un `false`.
- Tenga un método `devolverCoche` que dejará al garaje en estado de aceptar un nuevo coche.

8. Desarrollar una clase llamada Practica3 que en su método `main`:

- Cree un garaje.
- Cree 2 coches.
- El garaje irá cogiendo los coches y devolviéndolos, acumulando un importe aleatorio (`Math.random()`) de la avería tratada.
- Si la avería del coche es "aceite" incrementar en 10 los litro de aceite.
- Los coches entrarán al menos 2 veces en el garaje.
- Mostrar la información de los coches al final de `main`.