

1 - Cree un nuevo proyecto en C# con tres clases más otra clase para probar la lógica del código. Las clases principales del programa son las siguientes clases:

- Persona
- Estudiante
- Profesor
- Las clases de Estudiante y Profesor heredan de la clase Persona.
- La clase Estudiante incluirá un método público Estudiar() que escribirá en pantalla Estoy estudiando.
- La clase Persona debe tener dos métodos público Saludar() y SetEdad(int edad) que asignará la edad de la persona.
- La clase Profesor incluirá un método público Explicar() que escribirá en pantalla Estoy explicando.

Además crea un método público VerEdad() en la clase Estudiante que escriba en pantalla Mi edad es: x años.

Debe crear otra clase de prueba llamada EstudianteProfesorTest con un método Main para realizar las siguientes acciones:

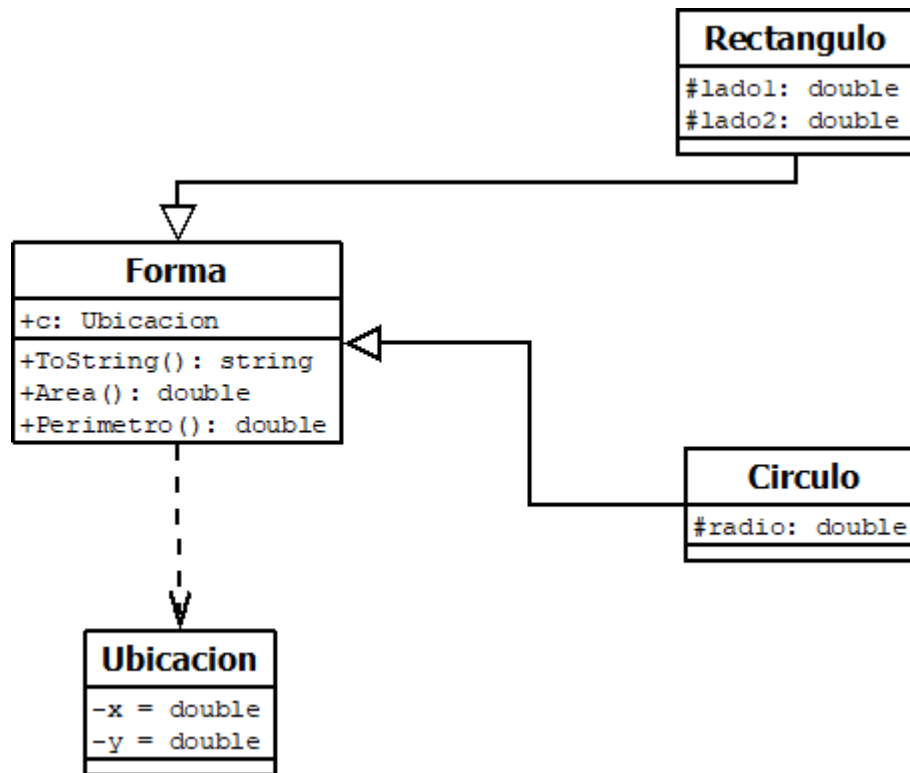
- Crear una nueva Persona y hacer que salude
- Crear una nuevo Estudiante, establecer una edad cualquiera, hacer que salude, mostrar su edad en pantalla y empezar a estudiar.
- Crear un nuevo Profesor, establecer una edad cualquiera, saludar y empezar la explicación.

2 - Crea un programa en C# que solicite tres nombres de personas al usuario y los almacene en un array de objetos de tipo Persona. Habrán dos personas de tipo Estudiante y una persona de tipo Profesor.

Para ello crea una clase Persona que tenga una propiedad Nombre de tipo string, un constructor que reciba el nombre como parámetro y sobrescriba el método ToString(). Después cree dos clases más que hereden de la clase Persona, se llamarán Estudiante y Profesor. La clase Estudiante tiene un método Estudiar que escribe por consola que el estudiante está estudiando. La clase Profesor tendrá un método Explicar que escribe en consola que el profesor está explicando. Recuerde crear además dos constructores en las clases hijas que llamen al constructor padre de la clase Persona.

Finalice el programa leyendo las personas (el profesor y los alumnos) y ejecute los métodos de Explicar y Estudiar.

3 - Cree un programa en C# que represente el siguiente diagrama de clases UML. En el diagrama se representan atributos públicos, privados y protegidos además de la dependencia de clases y la herencia.



4 - Crea un programa en C# que implemente una interfaz `IVehiculo` con dos métodos, uno para `Conducir` de tipo `void` y otro para `CargarNafta` de tipo `bool` que tenga un parámetro de tipo entero con la cantidad de gasolina a Cargar.

Después cree una clase `Coche` con un constructor que reciba un parámetro con la cantidad de gasolina inicial del coche y implemente los métodos de `Conducir` y `CargarNafta` el coche. El método `Conducir` imprimirá en pantalla que el coche está `Conduciendo`, si la gasolina es mayor a 0. El método `CargarNafta` aumentará la gasolina del coche y retornará `verdadero`. Para realizar las pruebas cree un objeto de tipo `Coche` con 0 de gasolina en el `Main` del programa y solicite al usuario una cantidad de gasolina para `CargarNafta`, por último ejecute el método `Conducir` del coche.

5 - Crea un programa en C# que implemente una clase abstracta `Animal` que tenga una propiedad `Nombre` de tipo texto y tres métodos `SetNombre(string nombre)`, `GetNombre` y `Comer`. El método `Comer` será un método abstracto de tipo `void`.

Además deberá crear una clase `Perro` que implemente la clase anterior `Animal` y el método `Comer` que diga que el perro está `Comiendo`.

Para probar el programa solicite un nombre de perro al usuario y cree un nuevo objeto de tipo `Perro` desde el `Main` del programa, asigne el nombre al objeto `Perro` y luego ejecute los métodos `GetNombre` y `Comer`.

6 - Dada la siguiente jerarquía de herencia, indica la visibilidad de los atributos de la clase `ProductoBancario` en las clases `CuentaJoven` e `Hipoteca`.

