Interfaz - nos obliga a definir los métodos establecidos

Las clases que se comprometen y cumplen con estos contratos tienen implementada esa funcionalidad y, por tanto, son capaces de llevarla a cabo.

Abstracta

Son aquellas que por sí mismas no se pueden identificar con algo 'concreto’, pero sí poseen determinadas características que son comunes en otras clases que pueden ser creadas a partir de ellas. (De ellas heredamos las características y los comportamientos pero no el modo de implementarlos.)

**De una interfaz si se crea una clase y de una abstracta no.**

Diagramas de clase: Visibilidad

.Nivel público: (public) símbolo (+), el atributo o método será visible

. Nivel protegido (protected) símbolo (#), atributo o método no será accesible

.Nivel privado (private) símbolo (-), solo acceden sus métodos

**Protegido que la clase puede acceder o no lo hereda**

**Una interfaz sirve para saber lo queremos hacer.**

**DEPENDENCIA**

**ASOCIACION**

ROL

MULTIPLICIDAD

NAVEGALIDAD

**AGREGACION**

**Asociación:** Varios estudiantes pueden asociarse con un solo maestro y un solo estudiante puede asociarse con varios maestros, pero no hay propiedad entre los objetos y ambos tienen su propio ciclo de vida. Ambos se pueden crear y eliminar de forma independiente.

**Agregación:** Un solo profesor no puede pertenecer a varios departamentos, pero si eliminamos el departamento, el objeto del profesor no se destruirá. Podemos pensar en ello como una relación \”tiene-a\”.

**Composición:** Una casa puede contener varias habitaciones: no hay vida independiente de la habitación y cualquier habitación no puede pertenecer a dos casas diferentes. Si borramos la casa – la habitación se borrará automáticamente.