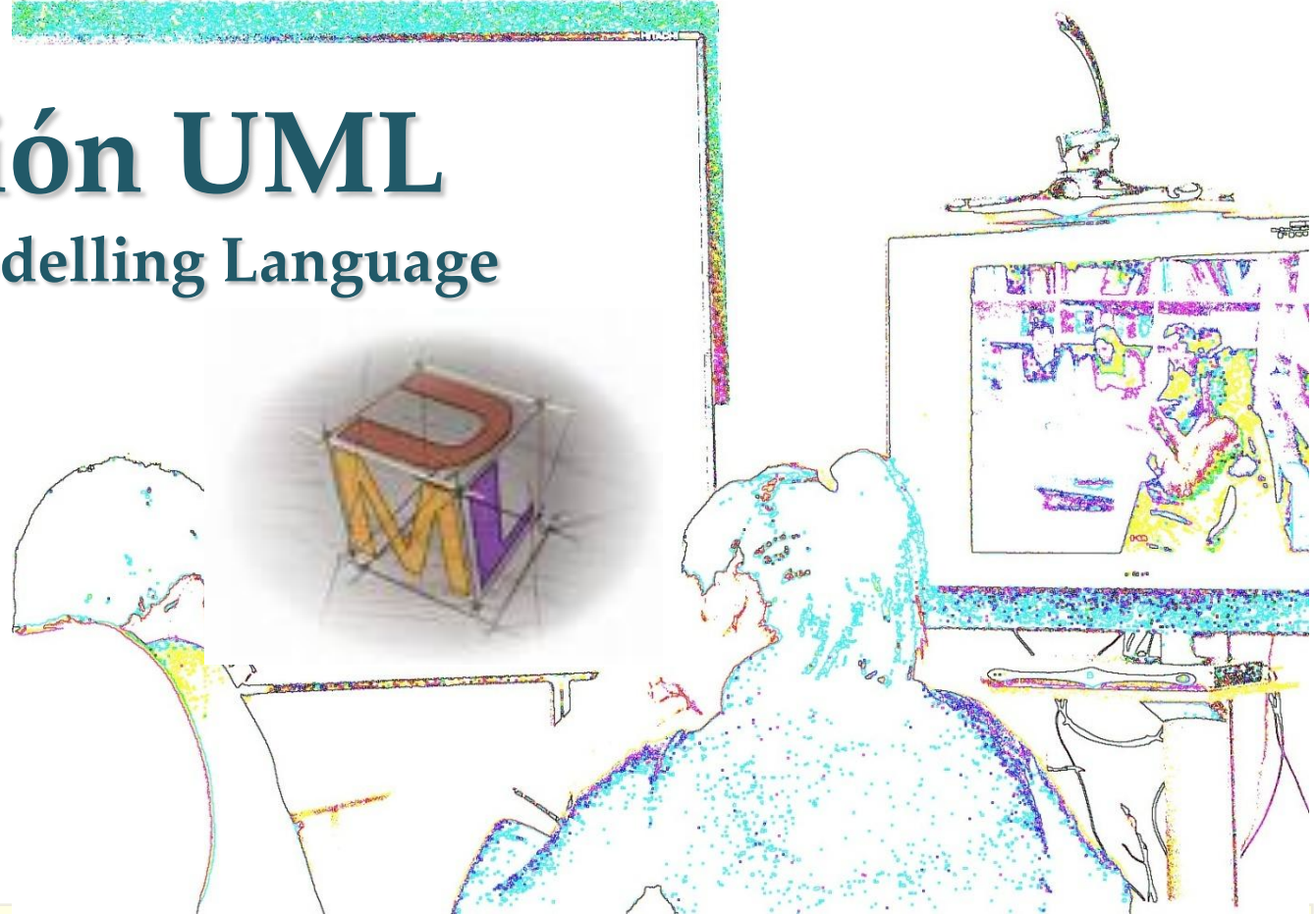


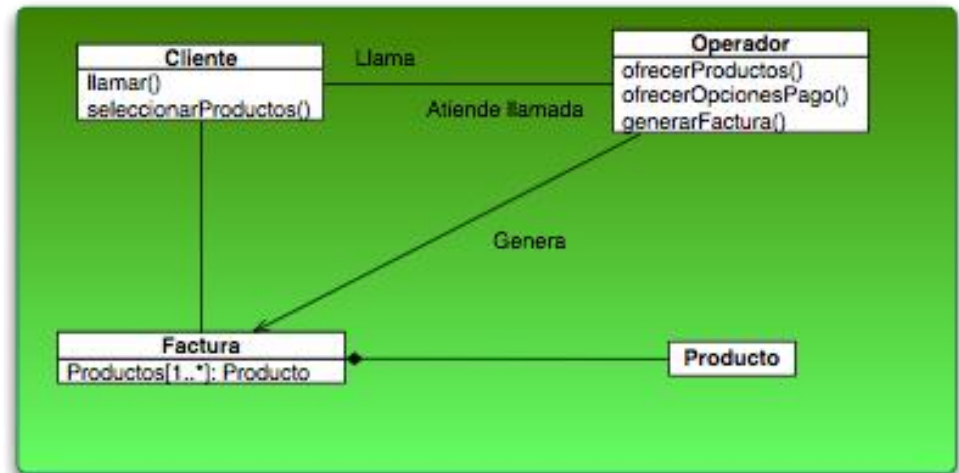
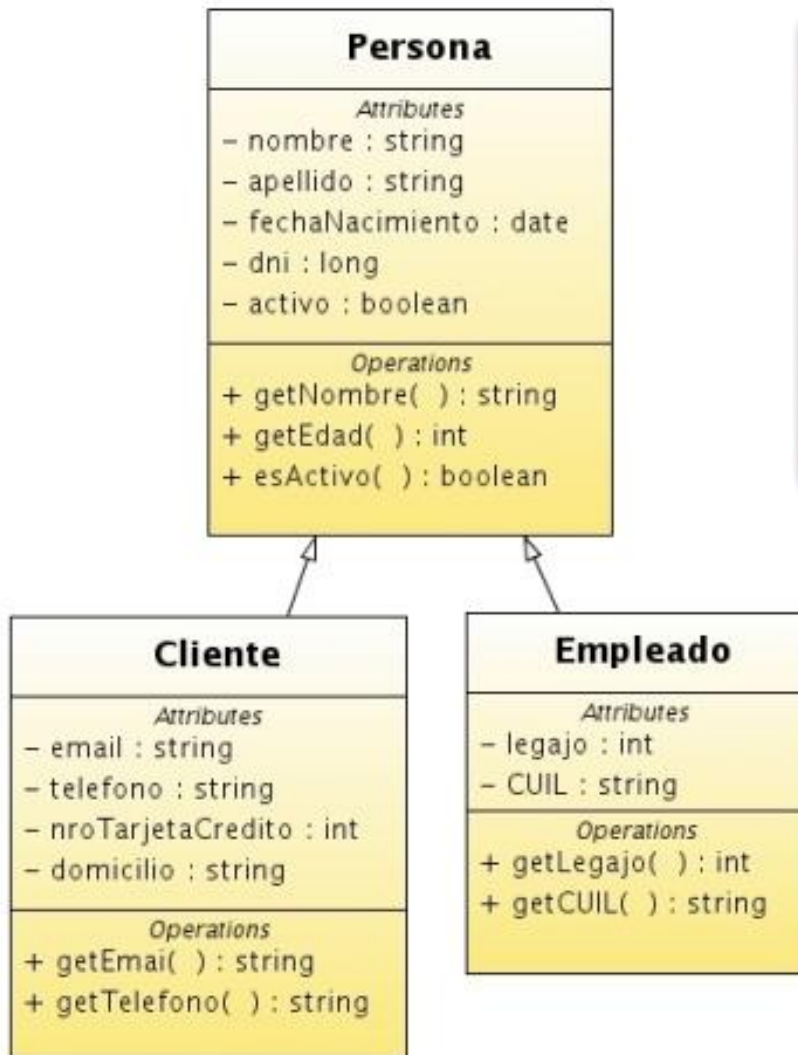
Notación UML

Unified Modelling Language



Covadonga Rodrigo San Juan y José Luis Delgado Leal
Profesores de Fundamentos de Informática

Notación UML



Representación de relaciones estáticas

Diagrama de clases

- Representación de entidades (clases) y sus relaciones estáticas.
- Muestra los componentes de cada clase

Cuenta
- saldo: float - titular: string - numeroCuenta: string
+ ingresar(cantidad:float):float + reintegrar(cantidad:float):float + consultarSaldo():float

Nombre

Atributos

Métodos/Procedimientos

Clases y Objetos

- Representación simple de clases: atributos y procedimientos
- Inicialización de valores en atributos

`Saldo:double=0.0`

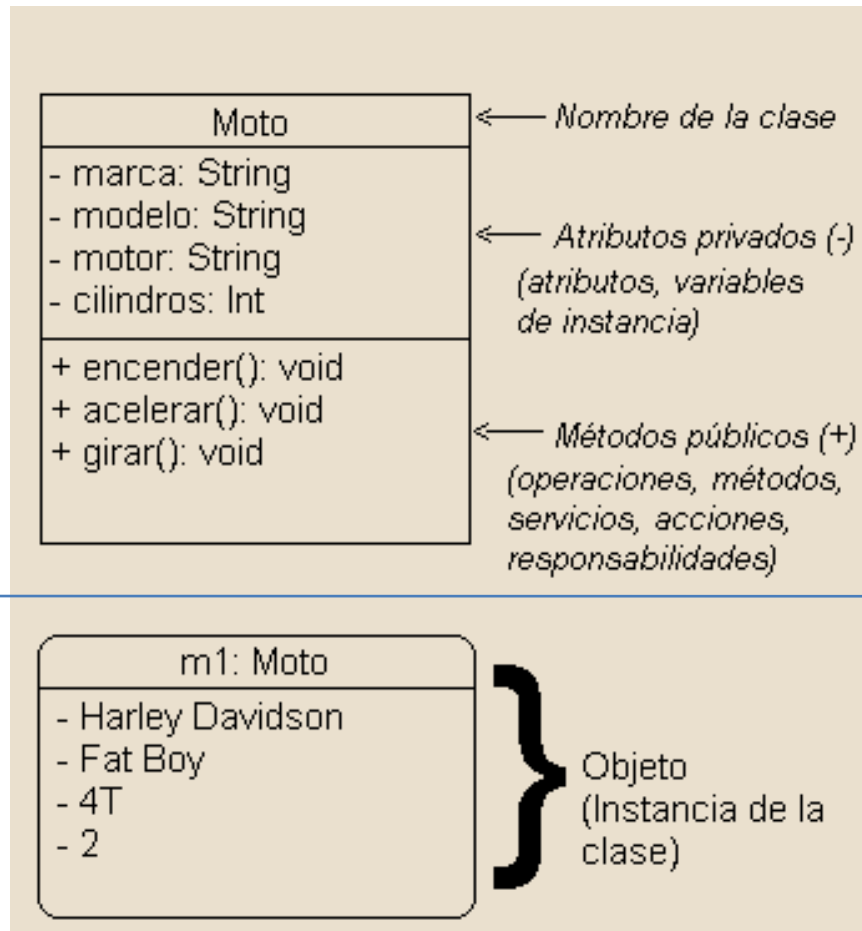
- En los procedimientos también se puede especificar el retorno

`ActualizarSaldo (valor:double): String`

- Representación de la visibilidad de los miembros de una clase: pública (+), protegida (#) y privada (-)

Diagrama de clases y objetos

- Representación simple de objetos



Clase

Objeto

Visibilidad de miembros

- Orden de los atributos: **public (+)**, **protected (#)** y **private (-)**
- Los atributos deben declararse **private (-)** excepto los que se quiera que sean accesibles por herencia que deberán ser **protected (#)**
- En los métodos los constructores se declaran en primer lugar. El resto por relación entre ellos.
- Hay una sola interfaz pública por fichero. Sus métodos son **public**, los de soporte privado (**private**)

Visibilidad en Java

Visibilidad	Java
+ Public	Un elemento siempre es visible en cualquier parte del programa y puede ser llamado y modificado por cualquier objeto del sistema.
- Private	Un elemento sólo puede ser usado por la clase que lo define.
#Protected	Un elemento puede ser usado por subclases y también por cualquier otra clase en el mismo Package como la clase propietaria. En Java, esto implica que la idea de visibilidad protegida es más público que Package.
Package	Un elemento sólo puede ser utilizado por otras clases que compartan el mismo Package.

Relaciones entre clases

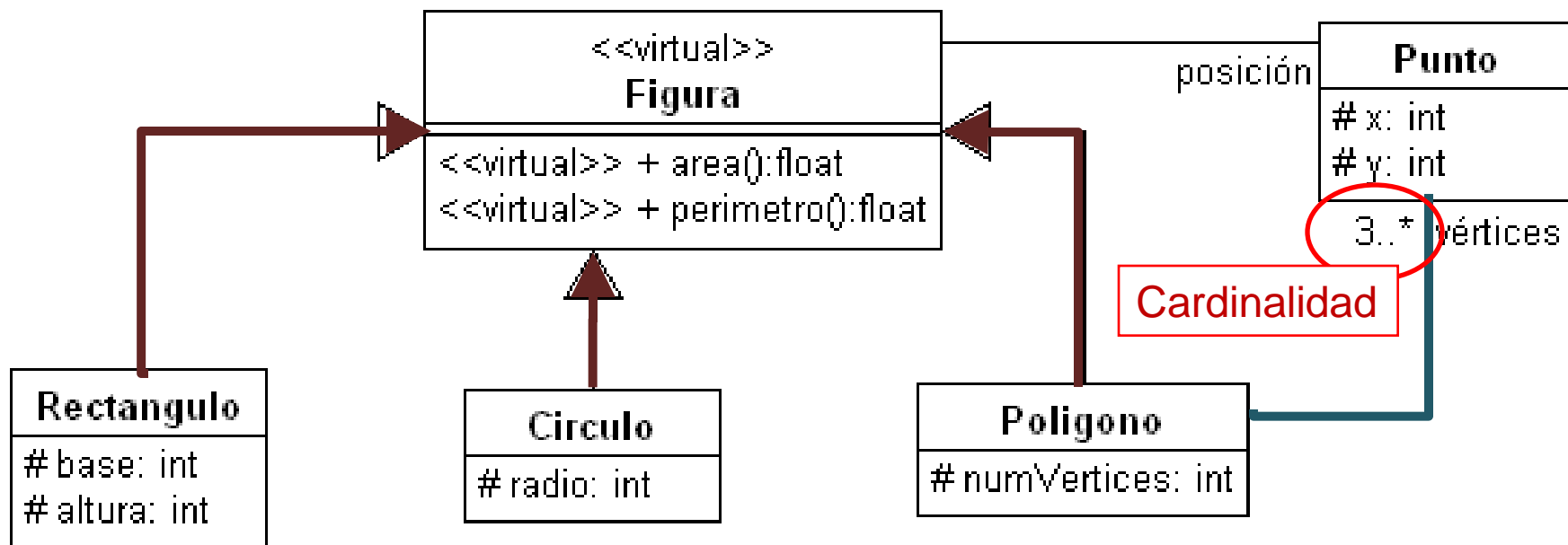
- HERENCIA: es una relación entre clases. Una representa el caso más general y las otras (hijas o descendientes) definen un subconjunto específico de ella.
- ASOCIACIÓN: relación entre ejemplares de la misma o diferentes clases. La multiplicidad o cardinalidad de la asociación indica número de ejemplares involucrados.

1..4: valor mínimo .. valor máximo

* = valor indeterminado

- DEPENDENCIA: relación de uso o utilización. Un elemento de una clase utiliza elementos de otra.

Diagrama estático de relaciones



Cardinalidad

1. Herencia o generalización ←
2. Asociación —