

# UNIFIED MODELING LANGUAGE

- object oriented analysis and design methods (appeared in late '80s and early '90s)
- graphical notation used to express designs

## Class Diagrams

★ DESCRIBES the TYPES of objects in the system and their kinds of static RELATIONSHIPS among them

"HAS A" → associations (a customer rents a number of videos)

"IS A" → subtypes (a mouse is a kind of person)

★ show ATTRIBUTES and OPERATIONS of a class

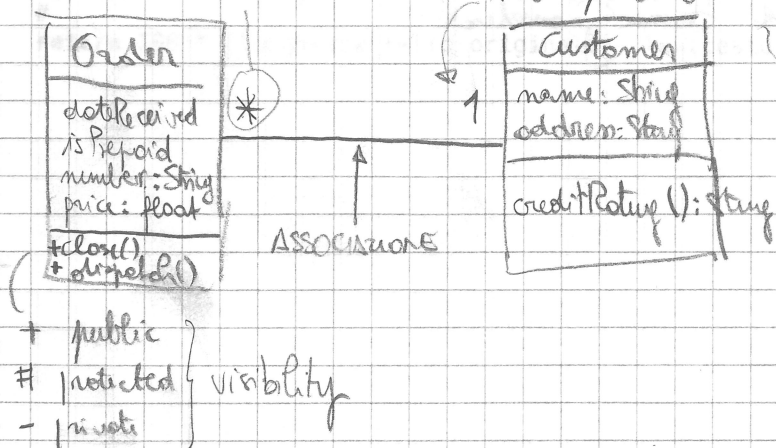
ATTRIBUTES → similar to associations (no conceptual difference)  
(think of them as small, simple classes, such as strings, dates, int and float)

OPERATIONS → processes that a class knows to carry out  
(methods of a class)

a Customer may have many Orders

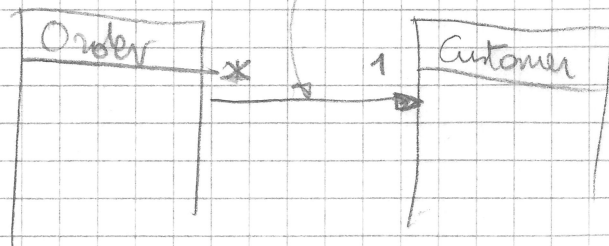
(methods of a class)

multiplicities: un ordine proviene un solo client



→ class

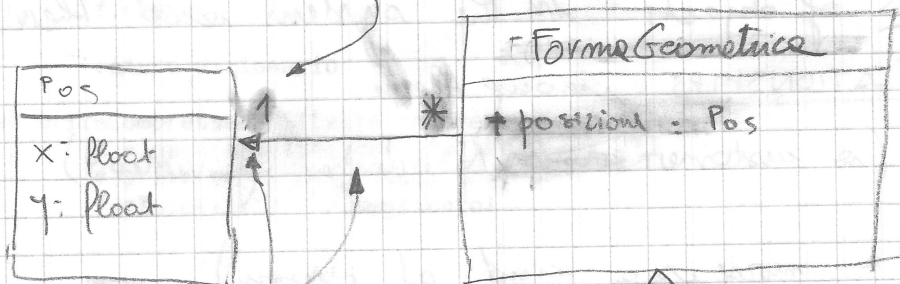
multiplicities



## Generalization (EREDITARIETÀ)

- similarities can be placed in a general class

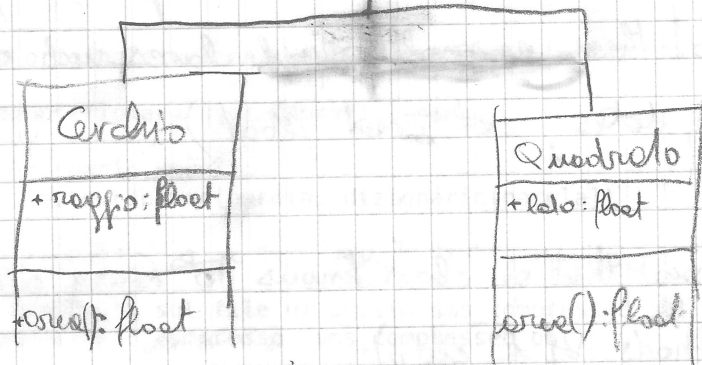
forme geom può avere una sola posit



HA una  
Posizione  
(coordinate)

responsabile di Forme geometriche di dire qual è la  
propria posizione, non viceversa

È una  
forme geom



Quadrato  
È UNA  
Forme Geometrica

↑  
Generalization  
(ereditarietà)

\* Cerchio è un TIPO SPECIALE di FormeGeometrica

INTERFACE : a class with no implementation

- Has operations declarations but no method bodies
- Has no fields

★ SUBCLASSING (ereditando) will provide the implementation

★ clients (utilization) will NEVER see the implementation, only the interface

⇒ corsivo per interface o metodo astratti

ABSTRACT CLASSES: may provide some implementation, but often they are used to declare an interface

