

Analisi e diagrammi di struttura: esercizi

Angelo Di Iorio

(in parte di: Gianpiero Favini e Sara Zuppiroli)

A.A. 2012-2013

Dove eravamo?

- Dopo l'analisi dei requisiti:
 - ▶ Conosciamo i casi d'uso di un sistema
 - ▶ Abbiamo steso una specifica dei requisiti
 - ▶ Abbiamo sequenze di eventi
 - ▶ Abbiamo un glossario di termini di progetto
- Prossimi passi:
 - ▶ **Analisi:** dare struttura ai requisiti, e decidere entità e relazioni tra entità
 - ▶ **Progettazione:** raffinare e decidere come implementare il sistema analizzato

Analisi vs. Progettazione

- L'analisi modella i concetti chiave del dominio del problema.
- La progettazione adatta il modello di analisi e lo completa affinché diventi implementabile.

In altre parole...

- L'analisi è più vicina al problema.
- La progettazione è più vicina all'implementazione.

Dal punto di vista di **UML**, si usano gli **stessi diagrammi con diversi livelli di dettaglio**: i diagrammi di analisi sono più 'astratti' di quelli di progettazione

Come procedere: analisi

- Estrarre un insieme di classi di analisi dalla specifica del problema (ne parleremo tra poco)
- Ragionare su queste **classi**: quali **attributi** e quali **operazioni** devono fornire?
- Stendere una mappa delle classi e delle loro **relazioni**.
- Modellare la dinamica delle classi con i *diagrammi di comportamento*.
- Procedere per **raffinamenti successivi** fino a quando il modello rappresenta efficacemente il dominio del problema.

Come procedere: progettazione

- Si parte dal modello di analisi che contiene classi abbastanza generiche, e lo si raffina.
- I costrutti più **astratti** di UML vengono trasformati in altri più **concreti** che possono essere implementati in un linguaggio di programmazione OO.
- Finalmente si considerano i **vincoli di piattaforma e linguaggio**, e i requisiti **non funzionali**.
- Le classi di analisi si trasformano in classi di progettazione (non c'è corrispondenza 1 a 1)
- Ancora una volta si procede per **raffinamenti successivi**.
- Il risultato è un modello pronto per l'implementazione.

Come estrarre le classi di analisi

- Una classe di analisi modella un concetto o entità del problema: se la specifica dei casi d'uso è buona i concetti basilari sono già in evidenza.
- I candidati più probabili sono nomi che compaiono nella specifica e nella documentazione.
- Una ragione in più per tenere un glossario di progetto: le parole nel glossario sono spesso candidati ideali per diventare classi di analisi.
- Le classi di analisi non sopravviveranno necessariamente alla progettazione.
- Due metodi molto diffusi per trovare le classi di analisi:
 - ▶ analisi nome-verbo
 - ▶ analisi CRC

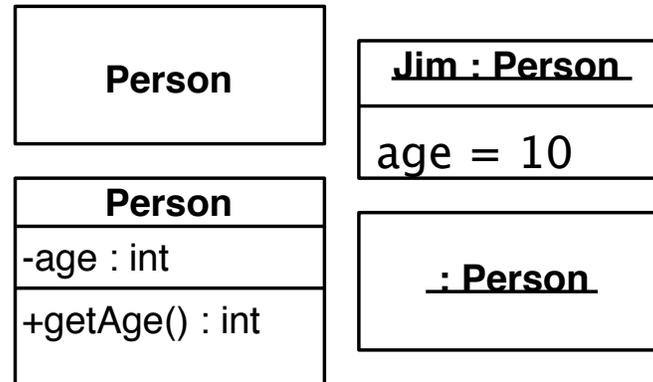
Analisi nome-verbo

- Si analizza tutta la documentazione disponibile, selezionando nomi e verbi.
 - ▶ I **nomi**: (es: conto corrente) sono i potenziali candidati per divenire classi o attributi.
 - ▶ I **predicati nominali**: (es: numero del conto corrente) sono i potenziali candidati per divenire classi o attributi.
 - ▶ I **verbi**: (es: aprire) sono potenziali candidati a divenire responsabilità di classe.
- Notate che ancora non parliamo di UML!

Analisi CRC

- Class-Responsibilities-Collaborators
- Si usano post-it divisi in tre sezioni chiamate proprio in questo modo.
- Si tratta di un metodo di brainstorming di gruppo che coinvolge sviluppatori, esperti, committenti.
- Si individuano i nomi delle classi, un insieme ristretto di responsabilità (cose che la classe sa/fa) e di classi collaboratori (alle quali viene richiesto comportamento/informazione).
- Le schede sono piazzate su un tavolo, la loro vicinanza fisica rispecchia quella logica.
- Si procede iterativamente.
- Usato in congiunzione con analisi nome-verbo.

Un po' di notazione UML



- Le classi possono avere fino a 3 slot:
 - ▶ uno per il nome (in UpperCamelCase) e l'eventuale stereotipo (slot obbligatorio)
 - ▶ uno per gli attributi (opzionale)
 - ▶ uno per le operazioni (opzionale)
- Gli oggetti (istanze) hanno una notazione molto simile
 - ▶ Il titolo degli oggetti è sottolineato e del tipo 'nome : classe', con nome opzionale.
 - ▶ Gli oggetti non hanno uno slot per le operazioni, possono definire valori per gli attributi.

Attributi e Operazioni

visibilità nome molteplicità:tipo=valoreIniziale

visibilità nome (nomeParam:tipoParam, ...): tipoRestituito

- Diversi livelli di dettaglio in fase di analisi e progettazione.
- Inizialmente solo gli attributi più importanti
- Tipi di visibilità:
 - ▶ + public
 - ▶ - private
 - ▶ # protected
 - ▶ ~ package

Relazioni tra classi

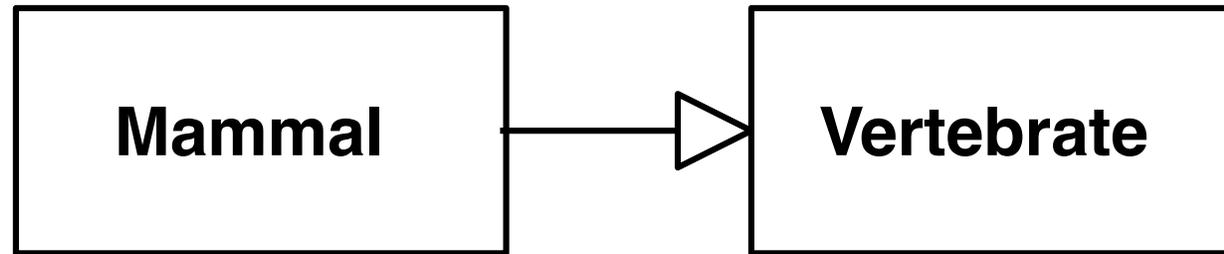
Ci sono alcune relazioni statiche tra classi particolarmente importanti in UML:

- Generalizzazione
- Associazione
- Aggregazione/Composizione

Vi sono poi altre due relazioni che possono legare le classi anche ad altri tipi di elementi (di cui per ora non parleremo):

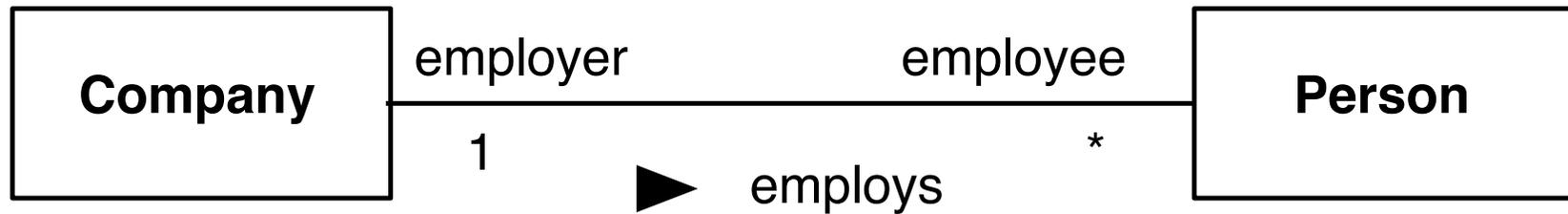
- Dipendenza
- Realizzazione

Generalizzazione



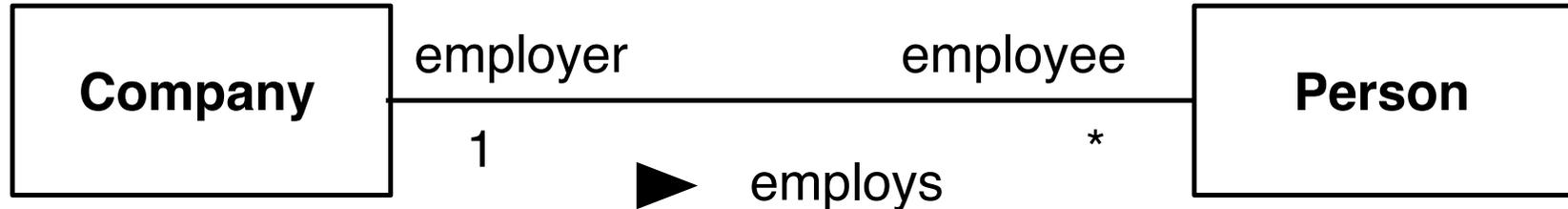
- Relazione tassonomica tra un elemento più generale e uno che lo specifica.
- La freccia parte dall'elemento specifico e punta verso quello più generale.
- Si tratta dell'ereditarietà in UML.
- Tra tutte le relazioni, questa è la più forte e vincolante.

Associazione



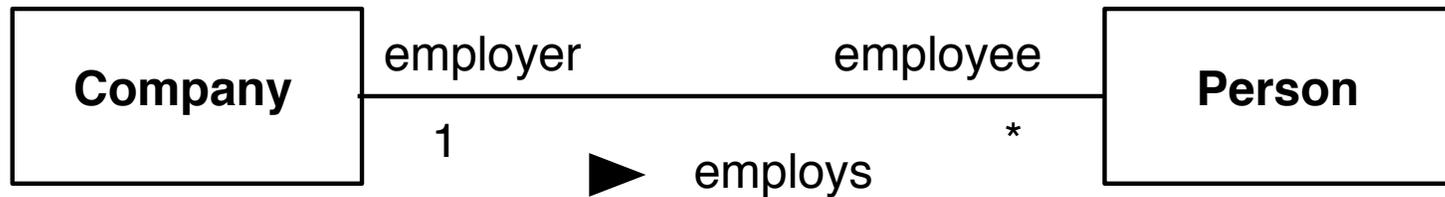
- Si tratta del tipo di relazione più generico: indica solo l'esistenza di collegamenti (link) tra le istanze delle classi.
- Rappresenta l'abilità di un'istanza di mandare messaggi a un'altra istanza.
- Può coinvolgere più di due classi e la stessa classe più di una volta.
- Tra le relazioni è anche la più flessibile e la meno vincolante.

Associazione: alcuni ornamenti



- Nome: opzionale.
- Triangolo direzionale: opzionale. Specifica la direzione in cui leggere l'associazione (aumenta la leggibilità).
- Ruoli: opzionali a ciascun estremo.
- Molteplicità: opzionale a ciascun estremo.

Molteplicità



- Indica quante istanze della classe in quell'estremo possono partecipare alla relazione.
- Può essere un numero o un intervallo min..max, con * che indica l'infinito.
- 1..3,7 significa 'da 1 a 3 oppure 7'.
- Molteplicità frequenti sono:
 - ▶ 1
 - ▶ 0..1
 - ▶ 1..*
 - ▶ *

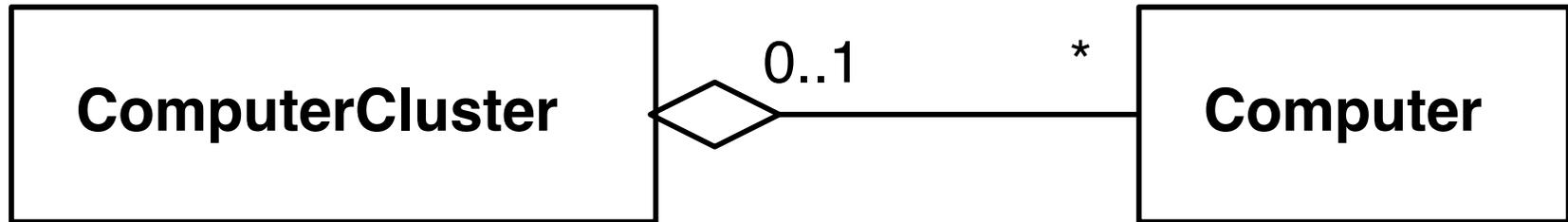
Aggregazione e composizione

Si tratta di particolari forme di associazione che rappresentano la relazione *whole-part* (tutto-parte) tra un aggregato e le sue parti.

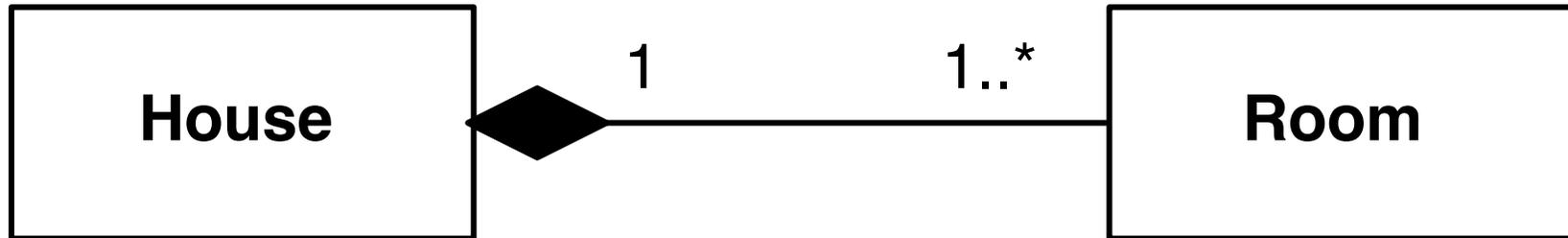
- **Aggregazione:** relazione poco forte, le parti esistono anche senza il tutto (es. i computer e il loro cluster).
 - ▶ È possibile che più aggregati condividano una stessa parte.
- **Composizione:** relazione molto forte, le parti dipendono dal tutto e non possono esistere al di fuori di esso (es. le stanze e la casa).
 - ▶ Se il composito viene distrutto, deve distruggere tutte le sue parti o cederne la responsabilità a qualche altro oggetto.

Aggregazione e composizione: notazione

Aggregazione



Composizione



Esercizi

De Montfort University (DMU)

Disegnare un diagramma delle classi di analisi per modellare il dominio:

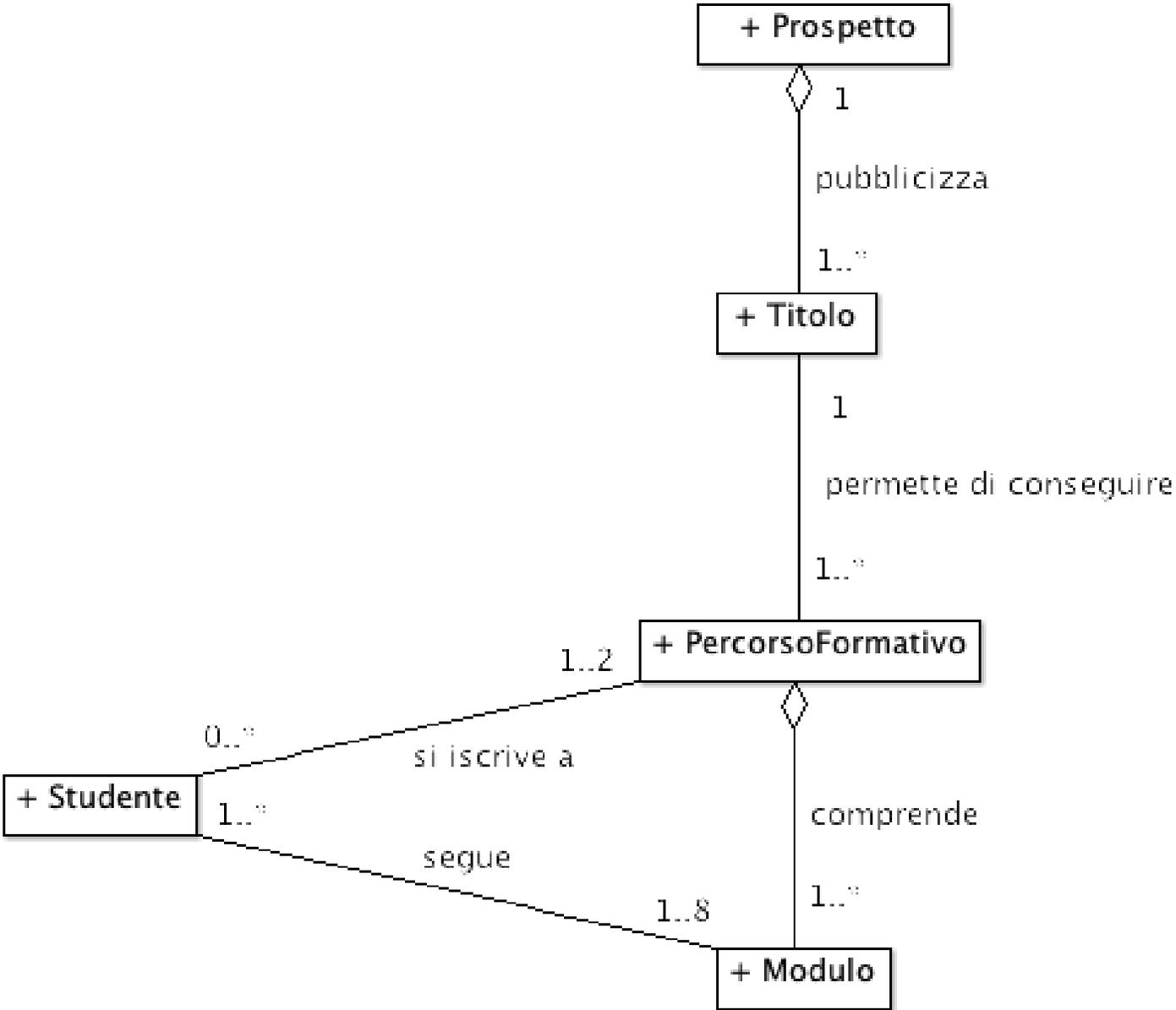
- De Montfort University (DMU) offre percorsi formativi ciascuno dei quali porta al conseguimento di un titolo di riconoscimento.
- Ogni titolo di riconoscimento è pubblicizzato nel prospetto informativo di DMU
- Ogni percorso comprende differenti moduli
- Gli studenti di un percorso seguono fino a 8 moduli all'anno
- Alcuni titoli sono 'congiunti', ad esempio uno studente può iscriversi a due differenti percorsi (come 'contabilità' e 'ragioneria')

De Montfort University (DMU)

Disegnare un diagramma delle classi di analisi per modellare il dominio:

- De Montfort University (DMU) offre **percorsi formativi** ciascuno dei quali porta al conseguimento di un **titolo** di riconoscimento.
- Ogni titolo di riconoscimento **è pubblicizzato** nel **prospetto** informativo di DMU
- Ogni percorso **comprende** differenti **moduli**
- Gli **studenti** di un percorso **seguono** fino a 8 moduli all'anno
- Alcuni titoli sono 'congiunti', ad esempio uno studente **può iscriversi** a due differenti percorsi (come 'contabilità' e 'ragioneria')

De Montfort University (DMU)



De Montfort University (DMU)

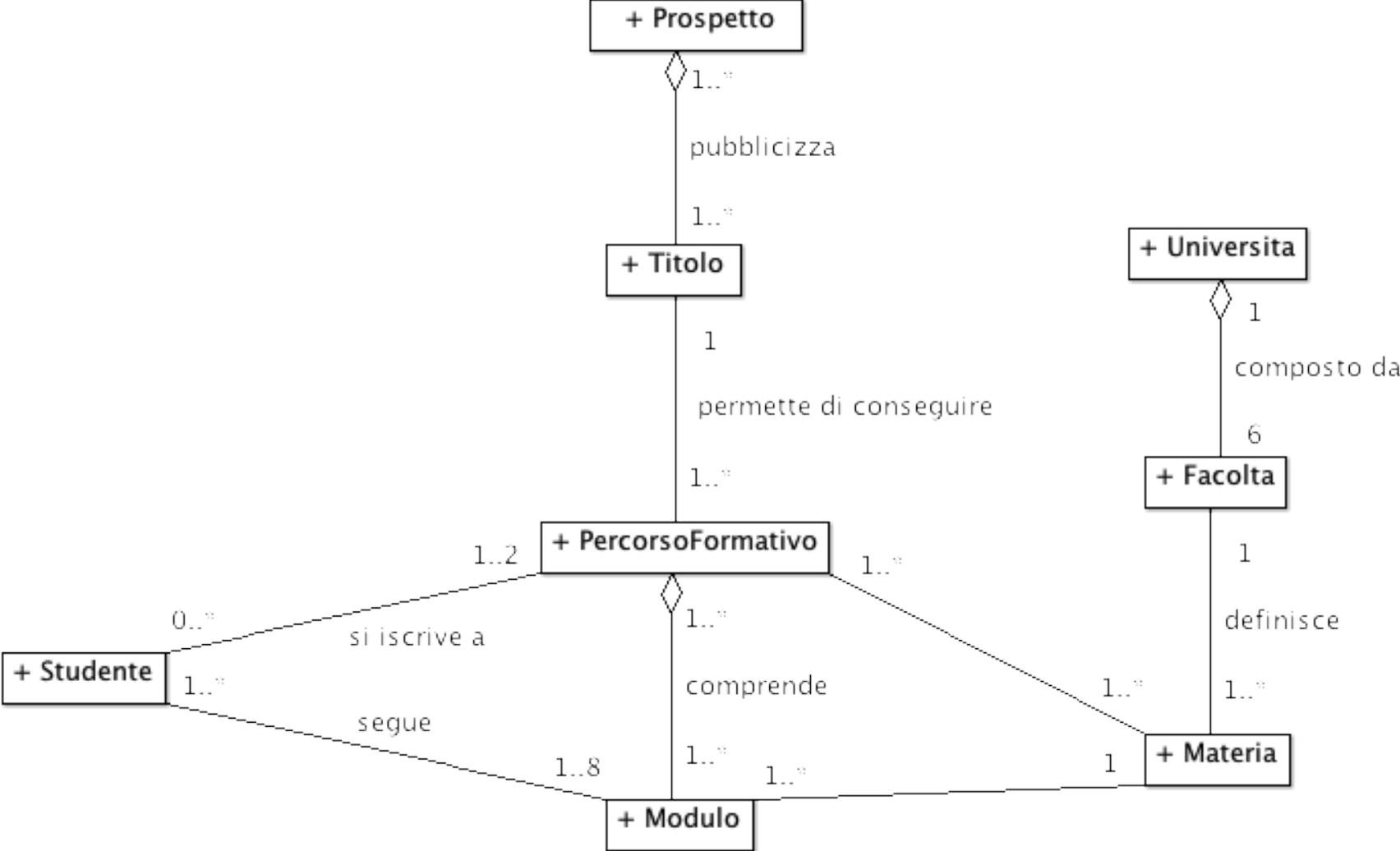
Completare il diagramma per rappresentare (parte 1):

- La DMU è composta di 6 Facoltà
- Ogni facoltà definisce un numero di materie ('contabilità', 'ragioneria', etc.) di cui si occupano moduli differenti (una materia per modulo) e che sono insegnate in percorsi differenti.

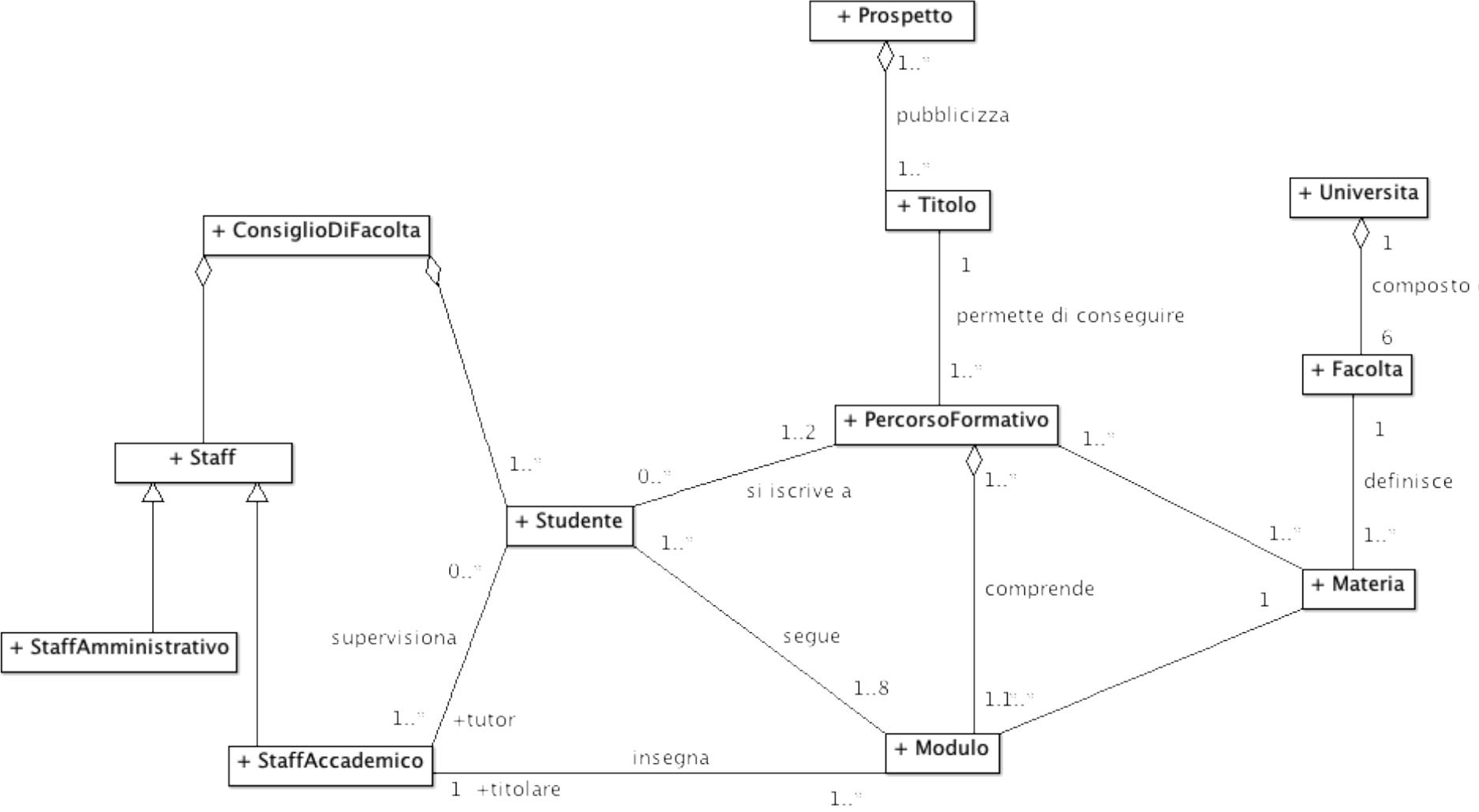
E successivamente (parte 2):

- Il consiglio di Facoltà è composto da studenti e da staff accademico o amministrativo
- Lo staff accademico insegna un numero arbitrario di moduli
- Lo staff accademico supervisiona diversi studenti, ciascuno dei quali segue un percorso formativo
- Alcuni rappresentanti dello staff amministrativo sono consiglieri ma non insegnano

De Montfort University (DMU)



De Montfort University (DMU)



Esercizio Birreria

- Compilare e discutere le schede CRC per rappresentare il seguente dominio:
- Una birreria è frequentata dai clienti e dallo staff. In particolare, lo staff raccoglie gli ordini e consegna le birre. Si paga alla cassa (e lo staff può dare il resto se necessario). Il gestore del pub si occupa, oltre che del servizio, anche di controllare la disponibilità di ogni birra in frigo e, se necessario, aggiungerne altre.

Esercizio Birreria: CRC staff

<u>Class: Staff</u>	
<u>Responsibility</u>	<u>Collaborators</u>
Prende gli ordini	
Prende le birre	
Serve le birre	
Prende i soldi	
Dà il resto	

Esercizio Birreria: CRC staff

<u>Class: Staff</u>	
<u>Responsibility</u>	<u>Collaborators</u>
Prende gli ordini	Ordine Cliente
Prende le birre	Birra Frigo
Serve le birre	Birra Cliente
Prende i soldi	Cliente Cassa
Dà il resto	Cassa Cliente

- Si può aggiungere qualche classe e collaborator?
- Qualche altra responsibility?

Esercizio Birreria: CRC gestore

<i><u>Class: Gestore</u></i> – <i><u>Superclass: Staff</u></i>	
<i><u>Responsibility</u></i>	<i><u>Collaborators</u></i>
Controlla birre in frigo	Frigo
Aggiunge una birra in frigo	Birra Frigo

Esercizio Birreria: CRC cliente

<u>Class:</u> Cliente	
<u>Responsibility</u>	<u>Collaborators</u>
Ordina la birra	Staff Ordine
Riceve la birra	Staff Birra
Paga	Staff
Riceve il resto	Staff

Esercizio Birreria: birra e frigo

<u>Class: Birra</u>	
<u>Responsibility</u>	<u>Collaborators</u>
Conosce il proprio prezzo	

<u>Class: Frigo</u>	
<u>Responsibility</u>	<u>Collaborators</u>
Conosce la disponibilità per ogni tipo di birra	
Permette di aggiungere una birra	

- C'è sufficiente coesione e disaccoppiamento?

Esercizio Elezioni

- Compilare e discutere le schede CRC per rappresentare il seguente dominio:
- Un sistema di voto da remoto prevede due modalità: touch-screen o via tastiera. Ogni sistema permette di esprimere il proprio voto per le elezioni comunali, regionali e nazionali. Gli elettori per votare inseriscono nel sistema un codice che gli è stato fornito in precedenza e, se il codice è corretto e non ancora usato, il sistema mostra a video le possibili scelte. E l'elettore esprime la sua preferenza.

Esercizio Elezioni

<u>Class: SistemaDiVoto</u> – <u>Subclasses: Touch-Screen, Tastiera</u>	
<u>Responsibility</u>	<u>Collaborators</u>
Conosce le elezioni in corso	
Mostra le elezioni in corso	
Valida il codice di voto	
Accetta un voto	

- Chi sono i collaborator?
- Qualche altra responsibility?

Esercizio Elezioni

<u>Class: SistemaDiVoto</u> – <u>Subclasses: Touch-Screen, Tastiera</u>	
<u>Responsibility</u>	<u>Collaborators</u>
Conosce le elezioni in corso	
Mostra le elezioni in corso	
Valida il codice di voto	
Accetta un voto	Voto
Permette di caricare le elezioni	Elezione

- Altri collaboratori?

Esercizio Elezioni

<u>Class: Voto</u>	
<u>Responsibility</u>	<u>Collaborators</u>
Sa per quale elezione è stato espresso	
Sa quale preferenza è stata data	

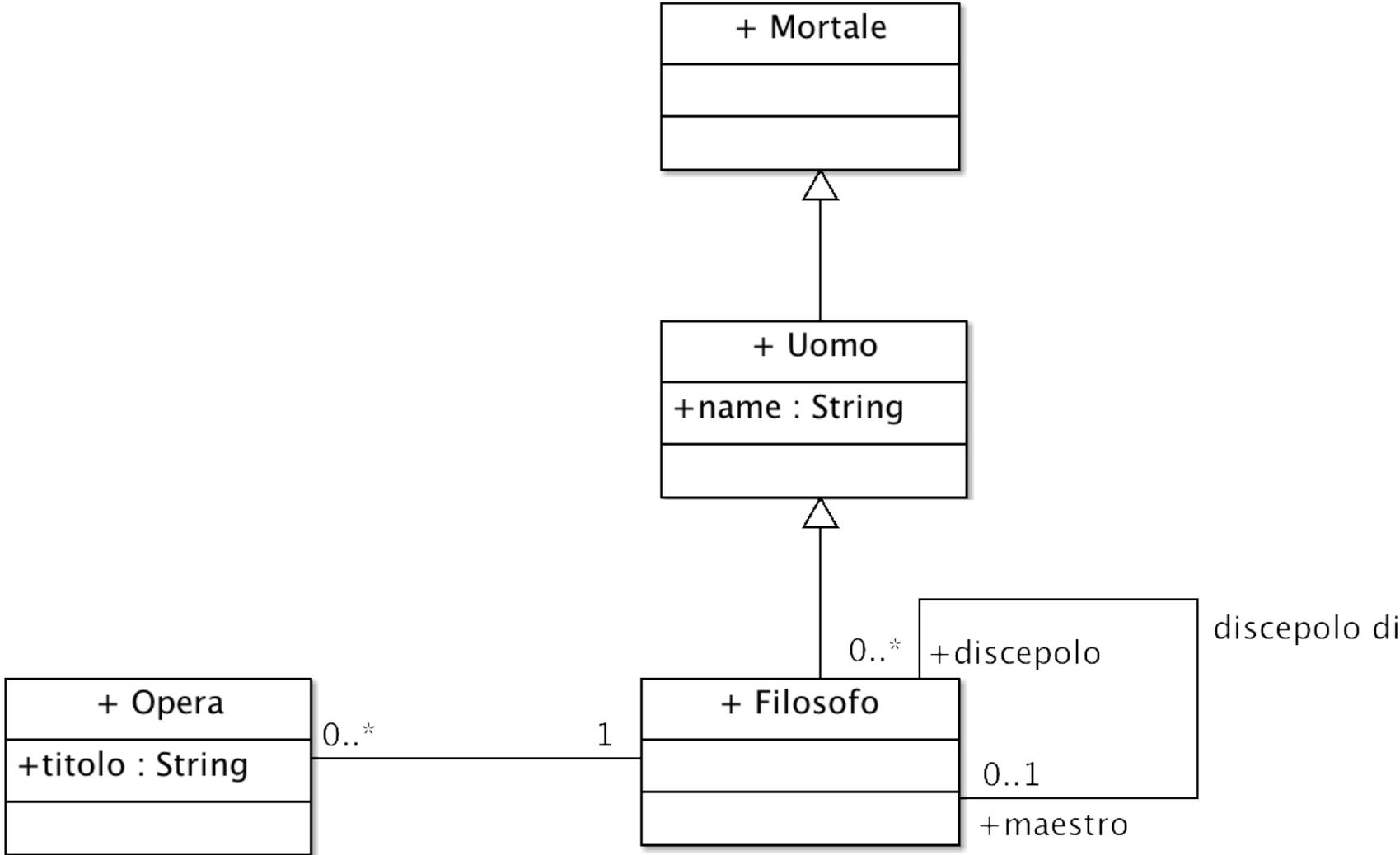
<u>Class: Elezione</u> – <u>Subclasses: Comunale, Regionali, Nazionale</u>	
<u>Responsibility</u>	<u>Collaborators</u>
Conosce i candidati	

- Altre classi?

Esercizio Filosofi

- Si usino un diagramma delle classi e uno degli oggetti per rappresentare:
- Tutti i filosofi sono uomini e tutti gli uomini sono mortali. Tutti gli uomini hanno un nome. Ogni filosofo è discepolo di al massimo un altro filosofo, e un filosofo può avere un qualunque numero di discepoli. Inoltre, un filosofo può produrre un qualunque numero di opere, ciascuna delle quali ha un titolo. Socrate, Platone e Aristotele sono filosofi; Platone è discepolo di Socrate e Aristotele è discepolo di Platone. Platone ha scritto 'La Repubblica'.

Esercizio Filosofi: classi



Esercizio Filosofi: oggetti

