

Università degli Studi di Trieste

Corso di ingegneria industriale

Esercitazioni di Fondamenti di Informatica

Giacomo Strangolino

mailto/chat: delleceste@gmail.com

Materiale didattico anche su:

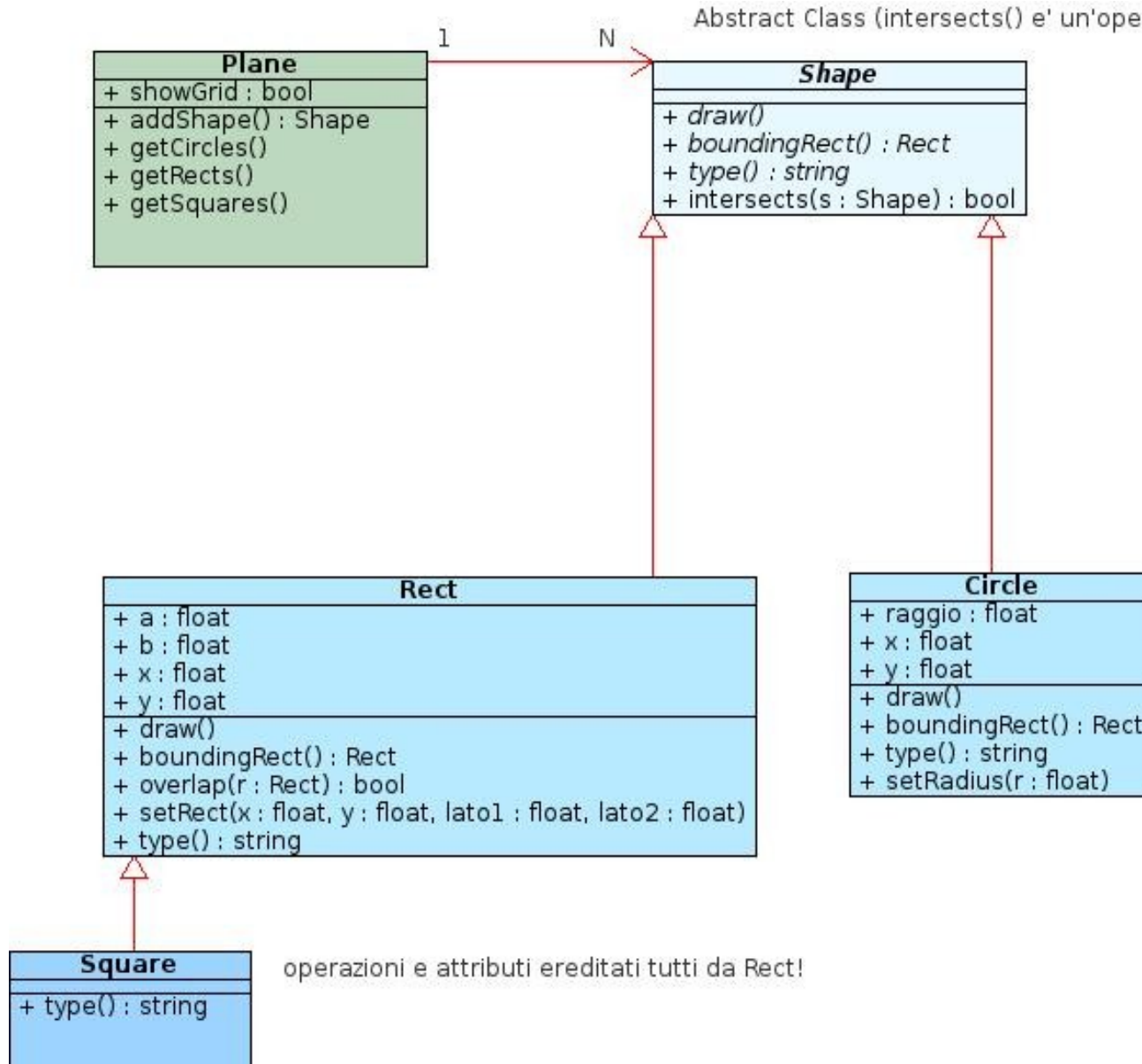
<http://www.giacomos.it>

(<http://www.giacomos.it/didattica/units/2012/>)

Lezione 8 (26/11/2012)

Classi e UML: Piano Cartesiano

Abstract Class (*intersects()* e' un'operazione che fornisce un'implementazione)



Classi e diagrammi UML

Piano Cartesiano

Da UML...



A java....
vedi file .java

Classi e diagrammi UML

Piano Cartesiano

Da UML...



A python....
vedi file cartesiano.py

Lezione 8

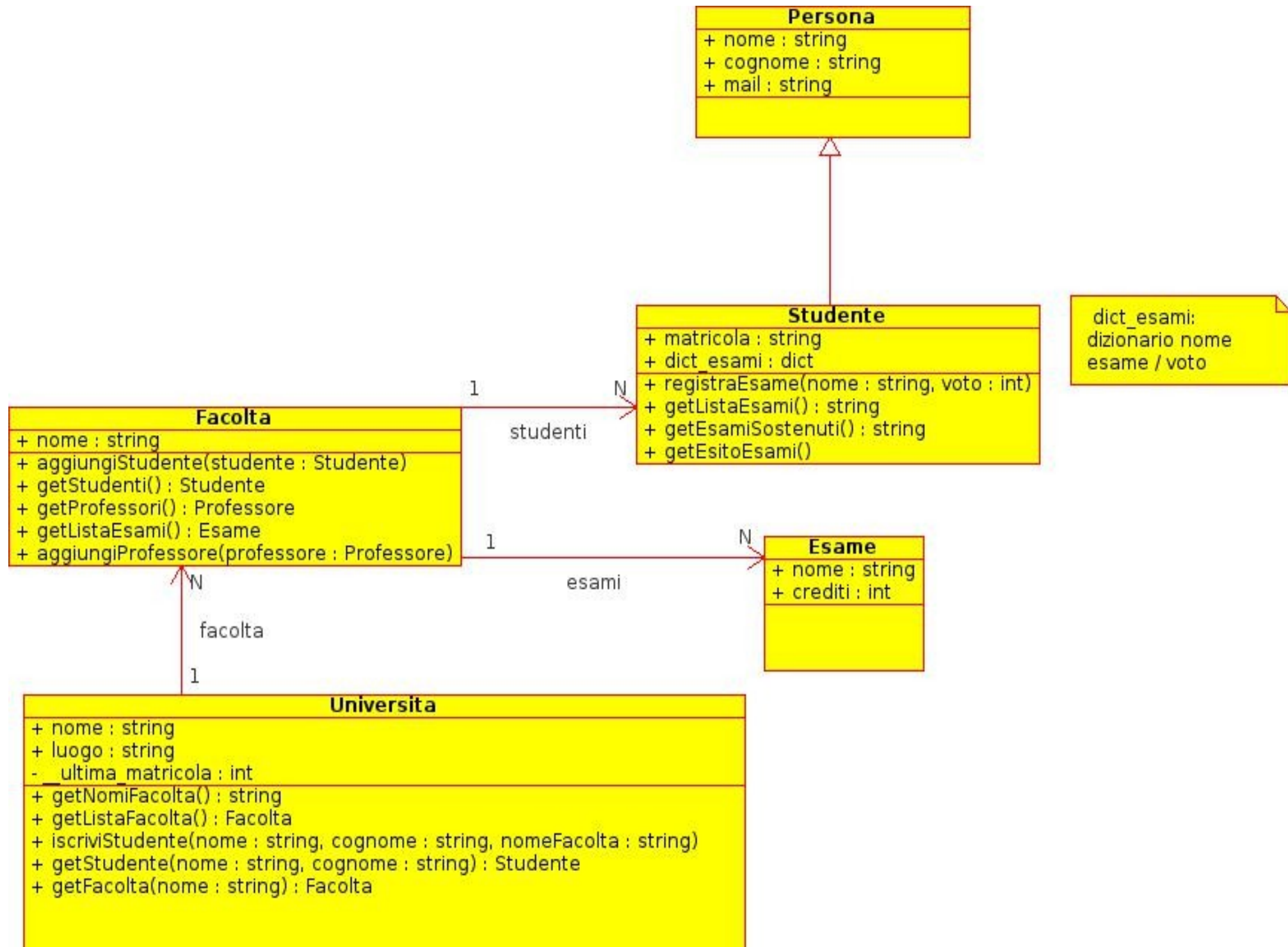
Classi e Programmazione ad Oggetti – Progetto Università

Si scrivano delle classi atte a modellare un'università con i propri *studenti*, *professori*, *esami* e *facoltà*. Deve essere possibile iscrivere studenti a una particolare facoltà, registrare esami, e chiedere alla classe università la lista delle facoltà e a ciascuna facoltà la lista degli studenti e di esami e quindi agli studenti i propri esami con il voto...

Lezione 8

Classi e Programmazione ad Oggetti – Progetto Università

- La classe **Persona** avrà un nome, un cognome e un indirizzo email
- La classe **Studente** è una Persona, in più ha una matricola, appartiene a una Facolta e avrà sostenuto una serie di Esami
- La classe **Facolta** è caratterizzata da un nome, da un certo numero di Esame e di Studente. Avrà dei metodi per aggiungere uno studente, ottenere la lista degli studenti, ottenere la lista degli esami.
- La classe **Esame** ha un nome e un numero di crediti.
- La classe **Universita** avrà un nome, un luogo e diverse Facolta.
- Deve essere possibile ottenere la lista delle Facolta, iscrivere uno studente, ottenere uno studente sapendo il suo nome e cognome e ottenere un riferimento ad una Facolta attraverso il suo nome.



Lezione 8

Homework

- Si completi il progetto *università* gestendo anche i professori.
Ogni professore insegnerà uno o più esami, apparterrà ad una ed una sola facoltà ed avrà uno o al più due assistenti, che saranno rappresentati da una nuova classe derivata da persona.
- Si modifichi il progetto università in modo che carichi gli studenti e gli esami da 2 file: il primo contiene righe del tipo:

Bruno, Resistenza, Ingegneria

Federico, Nice, Filosofia

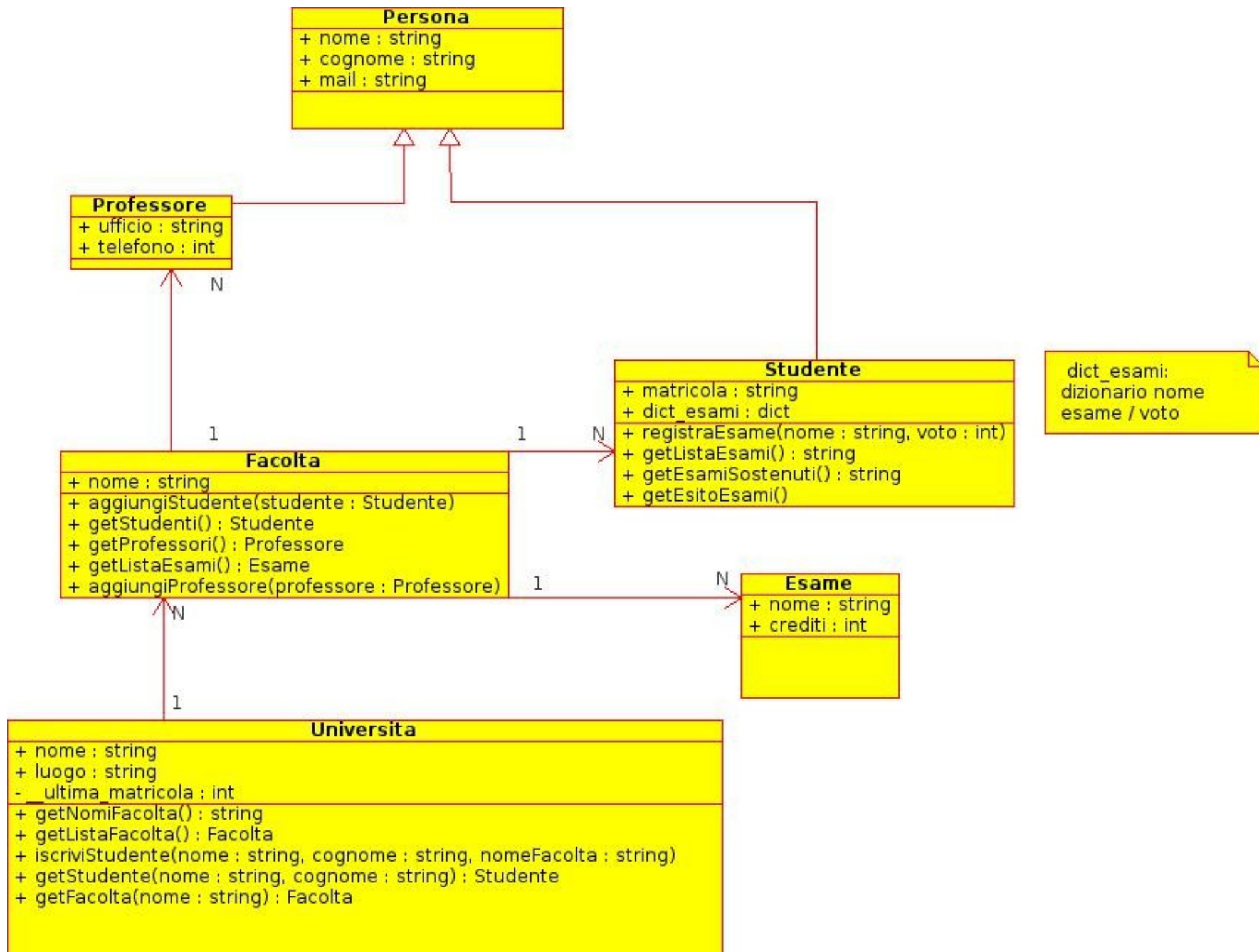
...

mentre il secondo:

Federico, Nice, Storia della Filosofia, 30

Federico, Nice, Filosofia Teoretica, 28

Tommaso, D'Aquino, Teologia, 30



Lezione 9

Homework

Si progetti un gioco per la battaglia navale.

- Vi saranno N navi, rappresentate ciascuna dall'oggetto *Nave*, con una posizione della prua, una dimensione e un orientamento (orizzontale o verticale) sulla scacchiera.

La classe *Nave* dovrà avere dei metodi che dicano se è *colpita(x, y)* o addirittura *affondata()*

- Un oggetto *Scacchiera* conterrà le N navi in una lista e dovrà fornire il metodo *colpisci(x, y)* che chiede ad ogni nave se (x,y) è una posizione che la danneggia o la affonda nel caso estremo.

La *Scacchiera* avrà altresì il metodo *disegna()* che ad ogni mossa eseguita dal giocatore deve disegnare la battaglia navale sotto forma di matrice contenente opportuni simboli adatti a descrivere la situazione corrente...

Lezione 9

Homework

Suggerimenti

- Si usi la funzione *randint(a,b)* del modulo random per generare un numero casuale tra 0 e N (ne serviranno 3, uno per la posizione x della prua, uno per la y e uno (0 oppure 1) per l'orientazione (orizzontale o verticale) di ogni nave).
- Se durante la generazione delle navi due navi si scontrano, bisogna rigenerare l'ultima nave finché non ci sono più intersezioni...
- La funzione che usa la classe Scacchiera dovrà eseguire un ciclo infinito finché o tutte le navi sono affondate o l'utente inserisce una stringa particolare, tipo “esci”, per terminare anticipatamente il gioco.