

# Programmazione Orientata agli Oggetti

## 8 febbraio 2023

### Cosa fare

Si progetti una o più gerarchie di tipo in Java in modo da supportare le operazioni indicate nel tema specificato sotto. Definire **classi e interfacce**, le signature dei **metodi**, e gli **stati astratti e concreti**, specificando il **ruolo** e **protocollo** delle classi/interfacce e i **contratti** dei metodi più importanti; **implementare lo stato concreto e i metodi**. Valuterò l'elaborato in base alla qualità del progetto e della sua implementazione (responsabilità, tipologie, contratti, ADT, parametrizzazione, qualità del body, pattern di design, astrazioni, incapsulamento). Non leggerò spiegazioni complesse, schemi e diagrammi.

Non consegnare il compito se non si sono implementati almeno i metodi per le operazioni principali e un Main che funga da client per questa API con esempi di chiamate di ciascuna delle operazioni principali.

La soluzione va scritta su dei file di testo – in particolare non binari - (ad es. usando IntelliJ, Eclipse o altro a piacere) che devono essere caricati sul sistema elearning.uniud.it entro il termine di scadenza (non ci sono deroghe se non per casi particolari concordati a priori e giustificati adeguatamente).

Scrivere i propri nome e matricola su ciascuno dei file con la soluzione.

## Esse4

L'azienda "Caneci" intende vendere a varie università italiane il software Esse4, successore di Esse3 attualmente in uso.

Con Esse4 si possono **gestire studenti** (coi loro dati, incluso la lista di corsi a cui sono iscritti, l'anno di corso, i CFU già ottenuti, gli esami già sostenuti con data ed esito) ,gli **appelli** previsti nell'anno accademico in corso (per un determinato corso in una determinata data e ora, durata e aula). Un **corso** ha un docente titolare, il numero di CFU, altri corsi come pre-requisiti, un anno di corso). Un appello può essere svolto di persona o in remoto, e in questo caso si basa su un quiz a domande chiuse (la cui correzione è automatica) e un "compito" dove lo studente deve caricare un elaborato che verrà corretto dal docente. Un appello può essere una prova finale, una prova parziale, e prevedere un eventuale orale. Uno studente può essere in-corso o fuori-corso, essere in regola o meno con le tasse.

Progettare e implementare un'API di supporto per questa applicazione, che copra queste operazioni:

- **iscrivi(studente, appello)** che aggiunge quello studente a un appello. Segnala ev. anomalie come studente che ha già sostenuto quell'esame, studente che non ha sostenuto qualche corso pre-requisito, studente non in regola con le tasse;
- **valuta(studente, appello, voto, giudizio)** che calcoli in punteggio del quiz se possibile, memorizzi il voto assegnato e il giudizio a tutto l'esame (se di persona) o del "compito" (se remoto), e calcoli l'eventuale voto complessivo;
- **calcolaVotoMedio(studente)** che calcoli il voto medio di tutti gli esami sostenuti dallo studente e il numero complessivo di CFU ottenuti.

**Obbligatorio:** scrivere un **main** che contenga chiamate a ciascuno di questi metodi.

**Nota:** non ho scritto da nessuna parte "stampare" né "visualizzare", e quindi non mi aspetto di vedere procedure che stampano alcunché.