

# INGEGNERIA DEL SOFTWARE mod. 2

## INDICAZIONI GENERALI SUL PROGETTO e SULLA VALUTAZIONE FINALE

### QUANDO CONSEGNARE:

Sia X la data dello scritto indicata sul SIFA, la consegna del progetto va effettuata entro la mezzanotte del giorno X+14 giorni.

### COME CONSEGNARE:

Il progetto va caricato nella apposita sezione di <https://upload.di.unimi.it>.

### COSA CONSEGNARE:

Va sottomessa una cartella (zippata) con nome **matricola\_cognome\_nome** [contenente](#):

- i sorgenti Java
- dump del database
- la relazione (in formato pdf)

## REQUISITI GENERALI

### REQUISITI DELLA DOCUMENTAZIONE ALLEGATA:

- La documentazione deve contenere la specifica dei requisiti considerati nell'implementazione del sistema richiesto
- La documentazione deve contenere i seguenti diagrammi UML:
  1. diagramma dei casi d'uso e descrizione degli scenari dei casi d'uso
  2. diagramma delle classi
  3. diagrammi di sequenza dei casi d'uso
  4. diagrammi delle attività
  5. macchine di stato
  6. diagramma delle componenti
  7. diagramma di deployment
- Vanno utilizzati (e giustificato il loro utilizzo) almeno i seguenti design pattern:
  1. Singleton
  2. MVC
  3. DAO
  4. Observer (i.e., Public-Subscribe)
  5. Uno a scelta tra quelli visti a lezione nel caso si individui la sua utilità
- Vanno giustificate in generale le scelte dell'architettura del sistema implementato in termini di alta coesione e lasco accoppiamento tra classi.
- Vanno specificati, dove occorre, un certo numero di vincoli OCL e deve venire specificato nella documentazione come siano stati mappati in JML. In caso si verifichino i vincoli utilizzando OpenJml (non obbligatorio visto i problemi di incompatibilità di OpenJML con versioni di Java  $\geq 8$ ), è opportuno darne evidenza nella relazione, riportando eventualmente violazioni di vincoli.
- Vanno documentati i test driver in JUnit e specificato il criterio di copertura utilizzato.
- Va descritta brevemente la GUI implementata, riportando alcune schermate a scopo esemplificativo ed eventualmente una *navigation map*.

## CHIARIMENTI SULLA DOCUMENTAZIONE:

- La documentazione deve includere anche tutte le spiegazioni necessarie a illustrare come è stata sviluppata la progettazione del sistema e modellata in termini di diagrammi UML (a volte non basta riportare solo il diagramma UML). Non basta presentare un codice funzionante per superare l'esame.
- È essenziale che ci sia tracciabilità tra requisiti, modelli e codice sviluppato. Questo vuol dire che ai requisiti descritti in linguaggio naturale devono corrispondere dei casi d'uso nel diagramma dei casi d'uso, che ci deve essere una descrizione degli scenari (almeno) per i casi d'uso principali, che ci deve essere un diagramma di sequenza per ogni caso d'uso di cui si sia descritto lo scenario, ecc. La tracciabilità con il codice viene invece ricercata nella corrispondenza tra le classi (attributi e metodi) del diagramma delle classi e il codice. Ovviamente, ci deve essere corrispondenza tra le classi (attributi e metodi) del diagramma delle classi e classi e metodi specificati nei diagrammi di sequenza, nei diagrammi di attività e nelle macchine di stato. Questo vuol dire che se durante il processo di sviluppo del progetto effettuate delle modifiche in un diagramma o nel codice, dovete modificare di conseguenza anche gli eventuali diagrammi coinvolti, in modo che venga mantenuta la coerenza.

## REQUISITI IMPLEMENTATIVI:

- il progetto va sviluppato in Java
- il progetto è individuale
- i design pattern devono essere stati implementati anche a livello di codice
- va sviluppata anche un'interfaccia grafica utilizzando Java Swing o JavaFX
- i dati persistenti devono venire memorizzati in un database.