

Note per lo svolgimento dell'esame di Ingegneria del Software – A.A. 20/21

Versione 17 Dicembre '21 – EV

1	Introduzione.....	1
2	Caratteristiche dell'elaborato auto-assegnato	2
2.1	Modo#1 – esercizio di micro-progettazione	2
2.2	Modo#2 - progetto e sviluppo di un modello di dominio.....	4
2.3	Modo#3 - progetto e sviluppo di un componente finalizzato	5
3	Preparazione all'esame.....	5
3.1	Iscrizione	5
3.2	nei giorni precedenti: predisposizione della postazione	6
3.3	Il giorno precedente: invio dell'elaborato e dichiarazione	6
3.4	Il giorno dell'esame: appello e colloqui	7

1 Introduzione

Il documento riporta indicazioni circa le modalità di svolgimento dell'esame di Ingegneria del Software nella fase avviata con il distanziamento sociale a partire da Febbraio 2020, consolidato nell'A.A. 20/21, e confermato per l'A.A. 21/22 sulla base di una positiva valutazione dell'efficacia del metodo nell'indirizzare la preparazione all'esame e nell'offrire una base per la valutazione della capacità dei candidati.

L'esame viene svolto attraverso un colloquio, nel quale assume un ruolo centrale la discussione di un elaborato di progettazione e produzione di software che lo studente può definire autonomamente entro modalità specificate o può anche concordare con il docente.

Il colloquio può svolgersi per via telematica, continuando la modalità praticata con buoni esiti fino da Febbraio 2020. Dove il candidato lo ritenga utile, il colloquio può svolgersi in presenza. In questo caso resta fondamentale che il candidato possa condividere lo schermo della macchina su cui ha installato il proprio ambiente di sviluppo. Nei casi residuali in cui il candidato intendesse svolgere l'esame in presenza ma non disponesse di un portatile con cui replicare il proprio ambiente di sviluppo presso la sede di esame verrà predisposta una modalità per permettere il colloquio.

Nel seguito di questo documento sono richiamati aspetti organizzativi e logistici per la preparazione e lo svolgimento dell'esame e sono poi riportate indicazioni circa la natura e l'intento dell'elaborato.

I concetti del disciplinare sono anche stati richiamati nella lezione del 17 dicembre 2021, di cui è disponibile la registrazione Webex.

2 Caratteristiche dell'elaborato auto-assegnato

La preparazione dell'elaborato è avviata a valle di una prima preparazione con la quale il candidato ha conseguito una generale padronanza dei contenuti del corso.

Per generale padronanza si intende che il candidato abbia ripercorso i contenuti delle slides e che sia in condizione di dare una propria interpretazione a quanto vi è scritto, che abbia ripetuto in autonomia qualcuno tra gli esercizi svolti a lezione o comunque sviluppati nei materiali condivisi, che abbia fatto esperienza nella produzione di codice Java e artefatti UML, anche solo replicando schemi UML inclusi nelle slides e esempi di codice Java distribuiti nei materiali condivisi. Per esperienza consolidata, questi passi costituiscono un percorso efficace attraverso cui acquisire conoscenze e capacità che il corso intende trasmettere.

L'elaborato costituisce a questo punto una occasione di approfondimento mirata ad alcuni intenti:

- creare un'occasione in cui lo studente si esercita nell'applicare metodi di ANALISI dei requisiti e PROGETTAZIONE Object Oriented;
- creare un'occasione in cui lo studente si esercita in un'attività che permette di acquisire fluidità nell'uso di strumenti di PRODUZIONE, tra cui in particolare uno strumento di produzione di UML (e.g. StarUML) e un ambiente di sviluppo integrato (e.g. Eclipse, IntelliJ);
- creare una base di discussione su cui potere interagire nel corso dell'esame.

Sono rilevanti ai fini della valutazione:

- la significatività e originalità del caso sviluppato;
- la qualità della metodologia seguita nello svolgimento;
- la qualità della presentazione, nella relazione scritta e nella discussione;
- la capacità di mostrare piena padronanza di quanto prodotto anche a fronte di richieste di modifica o discussione delle scelte.

Sono possibili diversi modi con cui scegliere il tema dell'elaborato, in relazione a quanto il candidato vuole avere occasione di apprendere circa la progettazione e lo sviluppo del software, all'interesse per la possibilità di integrarci un contenuto di problem solving legato a qualche problema specifico, e volendo anche in relazione alla possibilità di realizzare l'elaborato in aggregazione di più studenti.

In particolare, qui di seguito sono delineati 3 modi di definire il tema, che indicativamente corrispondono a livelli crescenti di complessità, che è bene siano commensurati alla base di partenza del candidato e agli obiettivi di apprendimento che intende consolidare:

- il Modo#2 è il metodo di riferimento;
- il Modo#1 è una semplificazione che può essere considerata da chi parte da una base più debole e ritiene di potere rinunciare a parte dei contenuti del corso;
- il Modo#3, viceversa, lascia spazio a una maggiore complessità per dare occasione di applicare i contenuti del corso alla soluzione di problemi di elaborazione dell'informazione.

2.1 Modo#1 – esercizio di micro-progettazione

È incentrato sulla sperimentazione di schemi progettuali e pratiche di sviluppo, dimostrati nel riferimento di un qualche contesto funzionale che serve da sfondo ma con rilevanza secondaria.

Permette di consolidare la capacità di sviluppare in Java, con applicazione di idiomi, schemi progettuali e pratiche di programmazione, anticipando e dominando la complessità attraverso modelli UML. Questo non valorizza a pieno i contenuti sviluppati nel corso, ma può essere un buon obiettivo laddove la sia più debole la base di partenza o l'interesse per la disciplina.

La relazione include:

- definizione di un problema
- progettazione e documentazione con artefatti UML
- implementazione delle classi Java
- un insieme di tests con argomentazione circa la loro progettazione

Nella definizione del problema vengono create condizioni per dispiegare in modo combinato una molteplicità di schemi progettuali o pratiche di programmazione in uno scenario di riferimento.

Questo tipo di elaborato è adatto ad essere svolto da un singolo studente, ed è conveniente che il tema sia definito in autonomia per evitare ritardi. L'identificazione di un tema dovrebbe potere essere effettuata senza eccessiva difficoltà a valle della preparazione dei contenuti del corso nelle parti "Java Basics and Idioms", "UML in implementation perspective", "Design Patterns". Esempi di come questo avviene sono già stati sviluppati nelle lezioni del corso (e.g. motivando e sviluppando una combinazione dei Patterns Composite, Observer e Adapter nello scenario del monitoraggio di un sistema a componenti soggetto a vincoli di legacy nel riuso di classi predefinite). Esempi di problemi possono essere identificati anche a partire dalla prima parte delle prove scritte assegnate negli esami anteriori a Febbraio 2021. Il valore dell'originalità è arricchito dall'aggiunta di schemi progettuali o pratiche che non siano già state trattate esplicitamente nel corso e dalla definizione di uno scenario di riferimento che non sia la diretta riproposizione di quanto già apparso in esami scritti assegnati o esercitazioni svolte a lezione.

Nella progettazione e documentazione è rilevante discutere il modo con cui schemi progettuali (Design Patterns) e pratiche di programmazione (copia difensiva, oggetti immutabili, ...) rispondono a specifici requisiti del problema. E' meno rilevante documentare le caratteristiche generali dello schema o della pratica, a meno che queste non siano originali rispetto a quanto già trattato nelle lezioni.

L'implementazione può essere documentata identificando le responsabilità delle classi, ad esempio attraverso il metodo delle schede CRC o anche riportando un commento. E' utile riportare e commentare frammenti di codice di particolare interesse per illustrare scelte progettuali o dettagli raffinati. Non serve includere tutto il codice nella relazione, semmai può essere lasciato in un'appendice. Può essere molto utile una descrizione di sintesi della complessità del codice, riportata in termini di numero di packages, classi, metodi, linee di codice.

Nella definizione dei tests è rilevante argomentare circa gli obiettivi di verifica con cui ciascun test è stato definito, che può corrispondere all'intento di usare un test per documentare un dettaglio della progettazione (e.g. mostrare che si innesca un qualche meccanismo di comportamento), verificare la correttezza coprendo elementi di vulnerabilità rispetto a possibili complessità nella struttura del codice (e.g. nell'installazione di dipendenze, nell'evoluzione di componenti dello stato), verificare il soddisfacimento di requisiti che caratterizzano il contesto funzionale). Nella realizzazione è raccomandato l'uso di JUnit.

2.2 Modo#2 - progetto e sviluppo di un modello di dominio

È incentrato sulla analisi dei requisiti di un contesto applicativo e sulla progettazione di un domain model che ne rappresenta le entità e permette l'implementazione dei casi d'uso identificati.

Permette di consolidare la capacità di mantenere visione complessiva sul processo di sviluppo software separando analisi dei requisiti progettazione e programmazione, con l'uso di modelli UML sia nell'analisi dei requisiti (class diagram concettuale, Use Case Diagram, templates, mockups) sia nella progettazione del modello di dominio dove possono trovare collocazione (se viene bene, senza doverlo fare necessariamente) gli stessi temi già identificati nella descrizione del Modo#1 (idiomi, schemi progettuali e pratiche di programmazione). La relazione include:

- definizione di un ambito di problema che identifichi concetti del dominio e requisiti funzionali
- identificazione di casi d'uso, di cui alcuni espansi in use case templates, di cui qualcuno accompagnato da mock-ups delle pagine su cui si immaginano realizzati i passi del caso d'uso
- progettazione di un modello di dominio capace di assumere le responsabilità sottese dai casi d'uso
- implementazione delle classi Java
- progettazione e sviluppo di test di unità che focalizzino specifiche complessità nella struttura della realizzazione e test funzionali che coprano particolari casi d'uso

Questo tipo di elaborato è adatto ad essere svolto da un'aggregazione di 2,3 studenti, ma non esclude lo sviluppo di un singolo individuo. Se è svolto da più studenti assieme è importante che ciascuno attraversi tutte le diverse fasi, evitando di separare compiti e competenze acquisite.

Per evitare ritardi, è conveniente che il tema sia identificato in autonomia facendo riferimento a un qualche dominio che gli studenti stessi conoscono per quanto è sufficiente a delineare requisiti e caratteristiche del problema. Esempi di problemi da cui trarre ispirazione possono essere reperiti nella seconda parte delle prove scritte assegnate negli esami anteriori a Febbraio 2021. Il valore dell'originalità è arricchito da approfondimenti, variazioni, e ancora più dalla definizione di uno scenario di riferimento che non sia la diretta riproposizione di quanto già apparso in esami scritti assegnati o esercitazioni svolte a lezione. Per chi ritiene che questo sia utile è possibile richiedere un riscontro a valle della definizione di uno statement iniziale nella forma di una pagina di testo che descrive il problema a grandi linee. Analogamente, può essere utile chiedere un riscontro nel momento in cui è stato identificato il modello concettuale e i casi d'uso ed essi sono stati documentati con diagrammi e un testo di accompagnamento, prima di avviare lo sviluppo delle classi Java; in generale, il riscontro che posso dare è proporzionato alla concretezza e qualità di quanto viene presentato.

Nello svolgimento dell'elaborato può essere utile raggiungere diversi livelli di completezza nell'analisi, la progettazione, l'implementazione e il test. In sostanza, mentre è utile identificare in modo ampio casi d'uso e entità del dominio, la progettazione può poi concentrarsi su un sottoinsieme dei casi d'uso, e solo alcuni di questi possono poi essere documentati attraverso use case templates, e solo alcuni use case templates essere espansi in mockups. Analogamente, i tests possono concentrarsi solo su alcuni punti di maggiore interesse. Complessivamente, ha molto più valore la varietà di strumenti e metodologie applicate rispetto alla completezza della realizzazione conseguita attraverso ripetizione.

Di norma, a meno che lo studente non abbia uno specifico interesse, la realizzazione non richiede lo sviluppo di un'interfaccia grafica. Essa può invece essere convenientemente sostituita

da un package che espone un insieme di metodi corrispondenti ai casi d'uso identificati nell'analisi dei requisiti, che potrebbero essere organizzati in classi distinte per i diversi ruoli dello use case diagram. Tali metodi vengono ad emulare quelli che sono i Controllers della Business Logic di una Web Application o di uno schema Model View Controller, e accedono direttamente ai metodi implementati nelle classi del modello di dominio. I test funzionali sono realizzati, con JUnit, esercitando i metodi corrispondenti ai casi d'uso.

Di norma, la realizzazione non include il livello di persistenza, che sia su database, su xml o che altro. Può essere realizzata, ma giusto se questi risponde ad uno specifico interesse degli studenti. In generale, il livello di persistenza può essere convenientemente surrogato attraverso una buona organizzazione dei test funzionali in JUnit.

2.3 Modo#3 - progetto e sviluppo di un componente finalizzato

E' finalizzato a rappresentare un problema su cui innestare una attività di problem solving e sviluppo di algoritmi di elaborazione dell'informazione rappresentata in un modello di dominio applicativo. E' da inteso per offrire spazio a chi ha interesse ad applicare progettazione e produzione software nella realizzazione di un componente interessante dal punto di vista di modellazione e sperimentazione.

Il problema può essere suggerito dal candidato sulla base di uno specifico interesse, più spesso è proposto dal docente in riferimento a progetti reali ma comunque qui riportati "a fatti stilizzati" e in isolamento.

Un ambito elettivo in cui sviluppare il tema può essere la realizzazione di un approccio di Model Driven Engineering nel quale un modello trattabile in modo automatico viene generato attraverso la visita dei dati rappresentati in un modello di dominio. A sua volta questo può applicarsi a molteplici finalità, tra cui ad esempio la costruzione di un componente analitico diagnostico o predittivo basato su un modello probabilistico, lo sviluppo di un componente di supporto alle decisioni attraverso la soluzione di un problema di ottimizzazione, lo sviluppo di un componente di analisi a supporto di processi di progettazione e verifica di sistemi software complessi.

Si presta bene ad essere sviluppato sia da un singolo che da un'aggregazione di 2, 3 studenti.

3 Preparazione all'esame

3.1 Iscrizione

Per partecipare all'esame è necessario essere iscritti ad un appello.

Chi ha interesse a svolgere l'esame in presenza lo evidenzia scrivendomi per posta elettronica in modo che possiamo concordare su orario e modalità. Nel subject del messaggio, ricordatevi di includere il termine [SWE] che uso per filtrare.

L'esame non necessariamente si svolgerà nella data dell'appello: lo studente potrà chiedere di rimandare la data, con l'intento di raggiungere una preparazione adeguata; io stesso sarò forzato a dilazionare le discussioni se queste si concentrano oltre il numero di 8-10 in una stessa mezza giornata.

3.2 nei giorni precedenti: predisposizione della postazione

Nel caso in cui il colloquio di esame viene effettuato in video conferenza, vi invierò un invito e voi vi collegherete tenendo (inizialmente) disattivo il canale audio. Durante gli esami, il candidato, e anche chi assiste deve tenere attivo il canale video. A meno di variazioni organizzative di livello superiore al nostro useremo Google Meet.

Nel corso dell'esame potrò chiedervi di produrre qualcosa usando il computer mentre mostrate lo schermo. Per questo dovete anche avere installati gli strumenti con cui siate abituati a produrre codice (e.g. Eclipse, IntelliJ, ...) e artefatti UML (e.g. StarUML, strumenti di produzione di Mockups, ...). La capacità di usare con fluidità gli strumenti di produzione è essa stessa parte della valutazione.

Occasionalmente può diventare necessario chiedere di produrre qualcosa sulla carta e poi mostrarmelo. Per questo dovete avere predisposto una postazione su un tavolo che vi permetta di posizionare la telecamera del PC in modo che io possa vedervi a metà busto, sia il volto che le mani, e possa eventualmente vedere un foglio posizionato sul tavolo davanti a voi, e voi possiate usare il computer quando ve lo chiedo. Dovete avere anche dei fogli bianchi e un pennarello a punta media con un tratto sufficiente a produrre uno scritto o disegno che risultino visibili.

Vi invito a sperimentare prima la vostra connessione, trovare una sistemazione idonea, e raggiungere una buona capacità nell'uso di Google Meet per evitare che l'attrito del mezzo finisca per penalizzare la presentazione. Purtroppo, non abbiamo il modo di effettuare l'esame nel caso in cui la connessione non sia sufficiente, in questo caso potremo recuperare attraverso l'uso di postazioni e ambienti resi disponibili dalla Scuola.

3.3 Il giorno precedente: invio dell'elaborato e dichiarazione

Entro le ore 18 del giorno precedente all'esame, dovete avermi inviato dalla vostra casella di posta elettronica unifi con un messaggio recante nel subject il termine [swe]

- il vostro elaborato auto-assegnato, in un file .zip o .rar che contenga il testo di report e il codice Java.
(dettagli sulla natura dell'elaborato sono riportati nel seguito, richiamando concetti già condivisi a lezione la cui osservanza diventa ora più rilevante ai fini della valutazione)
- una dichiarazione secondo il formato riportato sotto, nella quale accettate la modalità telematica per lo svolgimento dell'esame e dichiarate che durante la prova non utilizzerete strumenti di ausilio non autorizzati esplicitamente né avrete supporto da alcuno, fisicamente presente o in collegamento.
Mi è chiaro che difficilmente possiamo prevedere tutto, ma conto sulla vostra etica in un momento così complicato per il nostro Paese.

La dichiarazione può essere inclusa nel messaggio di posta elettronica, non è necessario che sia su un file allegato, ed è resa nel formato che segue:

Io sottoscritto ... NOME e COGNOME, numero di matricola ..., per l'esame di INGEGNERIA DEL SOFTWARE nel corso di Laurea ..., in relazione all'appello del ... e del colloquio previsto per il giorno ... dichiaro:

- di accettare la modalità telematica per lo svolgimento dell'esame;*
- che non utilizzerò fonti di informazione o strumenti salvo quelli che saranno esplicitamente autorizzati dal docente;*
- che non riceverò alcuno supporto da persone durante lo svolgimento dell'esame, presenti in forma fisica o in qualsiasi tipo di collegamento;*
- che rispetterò con lealtà tutte le restrizioni che sarebbero previste nel corso di un esame in presenza*

In fede

Nome e cognome

3.4 Il giorno dell'esame: appello e colloqui

Il giorno dell'appello, entro 30 minuti prima dell'appello, vi invierò per posta l'indirizzo di una video-conferenza, alla quale vi collegherete tenendo attivo il vostro canale video e disattivo il canale audio. All'ora dell'appello, identificati i presenti, vi comunicherò in modo indicativo l'orario del vostro colloquio.

Poi, procederemo con colloqui individuali. Chi è interessato potrà rimanere connesso, mantenendo attivo il canale video e disattivo il canale audio.