

ISTRUZIONI PROGETTI IS AA 2019-2020

LEGGERE ATTENTAMENTE TUTTO IL DOCUMENTO

Analisi, progettazione ed implementazione del progetto assegnato.

Oltre al progetto finale dovete consegnare una relazione sull'analisi e la progettazione in cui dovete inserire alcuni diagrammi UML:

- il diagramma dei Casi d'Uso
- diagramma delle classi a livello di dominio
- diagramma delle classi a livello di progettazione
- almeno un diagramma di sequenza (non SSD) e/o collaborazione
- diagramma dell'architettura software
- almeno un diagramma stati
- almeno un diagramma attività

Durante l'analisi e la progettazione farete tutti i diagrammi UML che vi saranno utili per la comprensione del progetto. I DIAGRAMMI UML NON DEVONO ESSERE FATTI A MANO e devono essere fatti con lo stesso strumento.

Dovete indicare nella relazione se avete deciso di applicare qualche **Design Pattern o Pattern Architetturale** sottolineandone la **motivazione**.

Inoltre indicate se avete seguito e applicato qualche specifico **design principle**, fra quelli di **Martin, SOLID, PHAME** (slide su Moodle: Progettazione dell'Architettura e Design Principle → Design Principle (Martin, Sharma))

Usate il tool Understand per capire se avete rilevato nel vostro progetto alcuni **antipattern strutturali** legati a problemi di dipendenza (Breakable, Butterfly, Hub,...) o altri problemi (slide su Moodle: Progettazione dell'Architettura e Design Principle → Antipattern Strutturali).

In caso di incomprensione di alcune parti del testo del progetto, proponete voi delle scelte o assunzioni che ritenete più logiche e opportune (non le più semplici...).

Inoltre sono richiesti due **diagrammi di Gantt** con Project che indicano le attività del progetto, i tempi relativi per lo svolgimento di tali attività, e le persone coinvolte. Il primo diagramma di Gantt rappresenta una stima delle attività, tempi e risorse (persone) coinvolte e deve essere inviato a claudia.raibulet@unimib.it entro una settimana dall'inizio del progetto. Il secondo diagramma di Gantt rappresenta le attività, i tempi e le persone effettivamente coinvolte e deve essere consegnato sempre a claudia.raibulet@unimib.it alla fine del progetto. Si consiglia di aggiornare sempre il diagramma di Gantt durante lo sviluppo del progetto per arrivare alla seconda versione.

Inoltre dovrete utilizzare

Git: verrà fornito a ogni gruppo un repository Git da usare per **gestire il codice del progetto**. La comunicazione tra i membri di ogni gruppo, per quello che riguarda il codice sviluppato dovrà passare dal repository. Verrà valutato il modo in cui viene usato Git per la gestione del codice, riguardo all'uso dei branch e al tipo di file caricati. Il contributo di ogni membro del gruppo sarà stimato anche in base al modo in cui ha usato il repository (non solo come quantità di operazioni, ma anche **la qualità**). La consegna del lavoro consisterà in un tag specifico del repository Git a voi affidato.

Accesso al server Git:

- Avrete l'accesso ad un repository github.
- Vi verranno mandate istruzioni di accesso.

SONAR: Usate SonarQube per mantenere controllato il livello di qualità del vostro lavoro. Nella consegna dovrete indicare tutte le impostazioni necessarie ad analizzare il vostro progetto con SonarQube, e **includere eventuali file di configurazione nel progetto** (pom.xml o sonar-project.properties). E' raccomandato consegnare un progetto che abbia **un rating livello A** nei vari indicatori di SonarQube, e senza violazioni critiche.

Installazione e configurazione di SONAR:

- alla pagina <http://www.sonarqube.org/downloads/> trovate i download per la piattaforma e gli analizzatori/plugin.
- Seguite le istruzioni di installazione:
 - <http://docs.sonarqube.org/display/SONAR/Installing>
 - <https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>
- Nota sull'uso del database: se non fate modifiche al file di configurazione, SONAR contiene un database H2 embedded che viene usato di default. Questo provoca un messaggio di avvertimento fisso nell'interfaccia web, ma ai nostri fini potrebbe essere sufficiente. Se invece pensate di impostare dei setup di gruppo di SONAR, è probabilmente meglio che lavoriate con un database esterno (es. MySQL).
- Come visto a lezione, un modo molto comodo di analizzare progetti Java è il plugin di Maven per SONAR:
<https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner+for+Maven>
- Anche in questo caso, se non applicate configurazioni custom, di default il plugin cerca un SONAR su localhost con database H2 nelle impostazioni di default. Per setup più complessi seguite la guida.

OPZIONE SONARCLOUD (consigliata): potete anche utilizzare la versione cloud di SonarQube, come visto a lezione. L'accesso è garantito tramite il vostro account github e come organizzazione potete utilizzare quella di laboratorio "[UnimibSoftEngCourse1920](https://unimibsoftengcourse1920.github.io/)".

- Link SonarCloud: <https://sonarcloud.io/about>

- Documentazione SonarCloud: <https://sonarcloud.io/documentation/integrations/github/>

TRAVIS: è consigliato, ma non obbligatorio, l'utilizzo di Travis per gestire la fase di build del vostro progetto. Se decidete di utilizzarlo, ricordatevi che potete automatizzare anche il lancio dell'analisi di SonarQube (tramite SonarCloud).

- Per maggiori informazioni riguardo Travis e SonarCloud, controllate la pagina Moodle nella sezione Laboratori.
- Link utile: <https://docs.travis-ci.com/user/sonarcloud/>

Il progetto consegnato dovrà essere funzionante e testabile da un utente tipo. Questo include che il sistema implementi i dovuti controlli dell'input fornito dall'utente. Questa condizione è necessaria per una valutazione sufficiente del progetto.

Per questo motivo, insieme al progetto è necessario consegnare una documentazione sufficiente a un utente per la sua installazione, esecuzione ed utilizzo. Questo implica che a partire dal contenuto del repository Git deve essere possibile in un modo documentato eseguire il build, il setup e il lancio dell'applicazione senza conoscenza pregressa del modo in cui essa è stata realizzata.

TUTTI DEVONO LAVORARE AL PROGETTO (IMPLEMENTAZIONE ED USARE GIT) E TUTTI DEVONO LAVORARE SUL DOCUMENTO DI ANALISI E PROGETTAZIONE

CONSEGNA PROGETTO

Chiarimenti sul modo in cui ci aspettiamo che venga eseguita la consegna dei progetti d'esame.

Per finalizzare la consegna, ci aspettiamo una comunicazione via email (ad Arcelli e Pigazzini) in cui un membro del gruppo notifica il termine del progetto, specificando:

- nome/numero del progetto e componenti del gruppo
- TAG Git da considerare per la consegna
- Relazione sul progetto: allegata alle email e inserita all'interno del repository Git; in questo caso, deve essere specificato il percorso del file da considerare.
- Informazioni relative alle modalità di installazione/esecuzione del progetto; possono essere incluse nella relazione oppure in un file a parte, ma la mail di consegna deve riportare un riferimento riguardante dove andare a trovare queste informazioni.

Anche se molte di queste informazioni sono intuibili dai nomi e contenuti dei file contenuti nei repository git, vi chiediamo precisione al riguardo, anche per evitare spiacevoli equivoci durante la correzione.

Per eventuali problemi relativi all'uso di Sonar, GIT o Travis potete contattare Ilaria Pigazzini: (ilaria.pigazzini@unimib.it)

NOTE

- Se troviamo progetti uguali, tutti e due i progetti verranno annullati.
- Se ci accorgiamo che un componente del gruppo non ha lavorato, dovrà sostenere un orale su tutto il programma del corso.
- I voti ai componenti del gruppo possono essere differenti.

Ognuno dovrà anche compilare a fine progetto alcuni questionari sulle vostre personali impressioni sull'utilizzo di Git, SonarQube, Project, Travis, Understand che avete utilizzato durante il progetto. Questi questionari li dovete mandare a claudia.raibulet@unimib.it

Riceverete i questionari quando consegnate il progetto.

ATTENZIONE!!!! NOTE PER L'ANALISI E PROGETTAZIONE IN UML

Fate particolare attenzione a:

- non confondere modello di dominio e modello di progettazione (i.e. oltre ai metodi, alcune classi di progetto non devono essere presenti nel modello di dominio)
- i diagrammi devono essere consistenti col codice che avete sviluppato: fra i diagrammi delle classi di progettazione dettagliati ci deve essere un mapping 1--1 col codice)
- diagramma di progettazione e diagramma di sequenza devono essere consistenti
- non confondete i diagrammi di stato e diagramma delle attività

NOI VALUTEREMO:

- Relazione del progetto con analisi e progettazione in UML: dovete seguire tutte le indicazioni elencate sopra.
- Progetto (Git, Sonarqube, codice,...)
- Orale
- Valutazione del Lab.
- Questionari: valutiamo se li compilate con attenzione!