Progetto di dettaglio e implementazione

In questa ultima fase dovete effettuare il progetto di dettaglio del software e implementarlo parzialmente. Considerate due operazioni di sistema, tra quelle che avete definito nelle fasi precedenti. Le operazioni di sistema devono far parte del caso d'uso (o dei casi d'uso) che descrivono la creazione di un test, e devono comprendere tanto un certo numero di operazioni di editing del test quanto un certo numero di operazioni di somministrazione di un test.
Nella documentazione della progettazione di dettaglio dovete produrre:

- Uno o più diagrammi delle classi che contengano le classi del software (classi Java);
- Un diagramma di sequenza (o di comunicazione) per ogni operazione di sistema, che parta da una invocazione dell'operazione di sistema ed indichi la successione di chiamate di metodo che viene effettuata per implementare tale operazione;
- L'indicazione dei pattern GRASP utilizzati per assegnare responsabilità alle classi, e degli eventuali pattern GoF applicati.

Riguardo allo strato Domain, raccogliete le operazioni di sistema utilizzando il pattern GRASP use case controller. Riguardo allo strato UI, considerate una semplice interfaccia utente testuale composta da una sola classe LinguisticTestsTUI. Riguardo allo strato dei servizi tecnologici, non è necessario che progettiate niente: assumete (temporaneamente) che l'applicazione sia tutta installata ed eseguita su una sola macchina, e che i dati creati non vengano salvati alla fine dell'esecuzione dell'applicazione.
Nel vostro progetto ponete particolare cura di queste cose (ogni deviazione sarà considerata un errore):

1. Il progetto di dettaglio deve essere coerente con l'architettura logica. Le classi devono fare parte di una partizione (può fare eccezione solo la classe LinguisticTestsTUI) in uno strato. Se non esiste nell'architettura logica una dipendenza tra una partizione ed un'altra, non devono esserci dipendenze tra le classi della prima partizione e le classi della seconda.
2. Rispettate i ruoli delle partizioni e degli strati: ad esempio, effettuare operazioni di input/output è permesso solo dalle classi nello strato UI (nel nostro caso, dalla sola classe LinguisticTestsTUI), e similmente implementare una operazione di sistema è permesso solo dalle classi nello strato Domain.
3. Nella indicazione di quali pattern GRASP o GoF usate specificate anche quali classi e quali operazioni sono coinvolte nel pattern.

Una volta completato il progetto di dettaglio implementatelo in Java, limitatamente allo strato Domain e allo strato UI. La classe LinguisticTestsTUI deve proporre un semplice menu testuale che permetta di creare un test, di fare le (poche) operazioni di editing che avete progettato, di marcare un test come definitivo, e di somministrarlo memorizzando le risposte. Ponete estrema attenzione a rendere il codice Java coerente con l'architettura logica e con il vostro progetto di dettaglio: ad esempio, gli attributi e i metodi nelle classi Java devono essere identici in nome e signature ad attributi e operazioni della classe UML corrispondente che avete messo nel progetto di dettaglio, e nel codice di un metodo le chiamate ad altri metodi devono rispettare i diagrammi di sequenza o comunicazione prodotti nel progetto di dettaglio. Ogni deviazione sarà considerata un errore.