

# Fondamenti di Ingegneria del Software a.a. 2019-20

## Laboratori 7-8:

### Esercizi relativo a persistenza, analisi statica del codice e metriche

Si consideri l'applicazione **LaTazza** utilizzata dalla segreteria dell'azienda ACME per gestire il business della vendita delle cialde di caffè, caffè arabica, thè, thè al limone, cioccolata, e camomilla al personale e clienti.

Di tale applicazione sono scaricabili da AW la documentazione (costituita da: problema, modello concettuale, Use Case Diagram e Use Case testuali), il design (modello UML disponibile come file Visual Paradigm), il codice Java e il JAR file corrispondente.

**E' richiesto di svolgere i seguenti task** che corrispondono ai laboratori 7 (persistenza) e 8 (analisi statica del codice e metriche):

1) Come passi preliminari:

- a. Scaricare i vari artefatti disponibili su AW, leggere con attenzione la documentazione (cartella Requisiti), aprire il file progettoLaTazza.vpp (cartella Design) in Visual Paradigm, importare il progetto LaTazza.zip in Eclipse. Se dopo l'import ci sono diversi errori del tipo: "Access restriction: The type 'Application' is not API" consultare e seguire una delle soluzioni proposte in: <https://stackoverflow.com/questions/25222811/access-restriction-the-type-application-is-not-api-restriction-on-required-l>
- b. Eseguire il codice di Latazza sia da Eclipse che considerando il file JAR cercando di capire le funzionalità offerte. Eseguire i casi di test JUnit presenti nel progetto LaTazza per capire il funzionamento del codice;

- 2) (**Laboratorio 7**) Sostituire nel progetto LaTazza il meccanismo di persistenza implementato con il file system (vedere il contenuto del file.txt - cartella Implementazione - usato come file per gestire la persistenza) con un DBMS. In particolare quello che dovrete fare sarà rifattorizzare il codice, e quindi preparare un nuovo progetto Eclipse, in modo da seguire il pattern architetturale DAO visto a lezione. Il **DAO** è un pattern che ha come scopo quello di separare le logiche di business da quelle di accesso ai dati. La logica di accesso ai dati dovrà essere implementata mediante tecnologia JDBC. Un sito Web che descrive come implementare la persistenza usando i DAO (chiamati nel sito DAL) e JDBC è il seguente: <https://dzone.com/articles/building-simple-data-access-layer-using-jdbc>. Un alternativa in italiano che spiega in maggior dettaglio il pattern è la seguente: <http://databasemaster.it/java-dao/>. Il DBMS che dovrà essere utilizzato è H2 (<https://www.h2database.com/html/main.html>) e dovrà essere utilizzato in modalità embedded. Per capire come scaricare, installare, utilizzare da Java il DBMS H2 fare riferimento al sito: [https://www.tutorialspoint.com/h2\\_database/index.htm](https://www.tutorialspoint.com/h2_database/index.htm). Per verificare che la nuova implementazione funzioni come la vecchia è possibile ri-eseguire la testsuite disponibile ed aggiungere ulteriori casi di test JUnit.

3) (**Laboratorio 8**) E' richiesto di analizzare il file LaTazza.jar (cartella Implementazione) usando il tool Stan4J (<http://stan4j.com/about/>) con due obbiettivi: verificare che il codice segua i principi di buona programmazione (ovvero che non ci siano violazioni delle metriche così come implementate in Stan4J, <http://stan4j.com/metrics/>) e verificare che il design fornito sia allineato con il codice. Per svolgere questo task è necessario utilizzare Stan4J come estensione (add-on) Eclipse (vedere <http://stan4j.com/download/ide/>). In particolare quello che dovrete fare è comprendere la logica delle metriche implementate nel tool, settare i parametri se necessario, calcolare le metriche e verificare se alcune sono state violate e perchè. Questo processo dovrà essere svolto ai vari livelli di: applicazione, package, classe e operazione. Per verificare l'allineamento design-codice dovrete invece ricavare le varie viste strutturali con Stan4J (<http://stan4j.com/dean/>) e controllare i diagrammi di design che vi sono stati forniti (disponibili nel file progettoLaTazza.vpp da visualizzare con VisualParadigm).

### **Cosa consegnare:**

Consegnare un file ZIP nominato: Labo1\_COGNOME1[\_COGNOMEn].zip che contiene:

- Eventuale file .txt o .doc contenente le informazioni del gruppo ed eventuali note su come eseguire il progetto ri-fattorizzato;
- Il progetto Eclipse contenente il nuovo codice rifattorizzato. Fare export come 'Archive file' usando la seguente procedura:  
<https://dzone.com/articles/exporting-and-importing>;
- Un file .doc contenente: le analisi svolte con Stan4J, le metriche che sono state violate nel progetto esaminato, una loro spiegazione e un possibile modo per rifattorizzare il codice al fine di risolvere le violazioni. Nello stesso file dovranno anche essere listati i disallineamenti trovati tra le viste prodotte da Stan4J e i documenti di Design.