**Programmazione ad oggetti**

Prof. Daniele Toti

***OBIETTIVO DEL CORSO E RISULTATI DI APPRENDIMENTO ATTESI***

L’insegnamento si propone di fornire agli studenti una panoramica della programmazione orientata agli oggetti, con specifico riferimento al linguaggio Java.

In quest’ottica, l’insegnamento tratta, con un certo livello di profondità, la maggior parte degli aspetti relativi alla programmazione di applicazioni orientate agli oggetti di tipo stand-alone o desktop attraverso la Standard Edition (SE) del linguaggio Java, con particolare riferimento alle ultime versioni del linguaggio che godono di supporto a lungo termine.

Risultato di apprendimento atteso è la capacità di realizzare applicazioni di media complessità dotate di interfaccia grafica, quali per esempio semplici videogame o applicazioni a finestre. Uno studente che segue con successo il corso avrà inoltre gli strumenti necessari per affrontare esami di certificazione relativi a Java SE, e avrà acquisito le competenze di base per svolgere il ruolo di sviluppatore software in contesti aziendali.

***PROGRAMMA DEL CORSO***

* Introduzione alla programmazione orientata agli oggetti e al linguaggio Java
* Sintassi di base, package, import, metodo main, I/O da console
* Variabili, assegnazioni e tipi primitivi
* Ambiente di programmazione e IDE: Eclipse
* Meccanismi fondamentali: commenti, istruzioni condizionali, cicli, operatori
* Costruttori, attributi, modificatori di accesso, metodi, cast
* Heap e Stack
* Stringhe, array, classi wrapper
* Ereditarietà, polimorfismo, classi astratte e interfacce
* Collezioni e generics
* Eccezioni
* File e gestione di elementi del file system: java.io, java.nio
* Utilizzo di librerie esterne
* Gestione delle dipendenze con Maven
* Enum
* Classi annidate
* Concorrenza e multi-threading
* Date, Locale, ResourceBundle
* Elementi di programmazione funzionale, espressioni Lambda e Stream
* Interfacce grafiche con Java Swing: aspetti fondamentali, elementi di grafica 2D, thread e worker
* Programmazione a eventi
* Introduzione alla progettazione orientata agli oggetti
* Pattern di design: GRASP, GoF
* Pattern architetturali: MVC
* Controllo di versione del software
* Aspetti relativi alla professione di sviluppatore software, con riferimenti a progetti reali e ad aziende del settore

***BIBLIOGRAFIA***

Slide delle lezioni e delle esercitazioni, appunti delle lezioni, soluzioni degli esercizi svolti, risorse online comunicate agli studenti.

Libri di testo consigliati:

* *Head First Java,* 2nd Edition
* *Think Java,* 2nd Edition
* *Thinking in Java*, 4th Edition
* *Applying UML and Patterns,* 3rd Edition
* *Design Patterns*

alcuni dei quali messi a disposizione degli studenti sulla piattaforma Blackboard assieme al resto del materiale. Il corso non seguirà alla lettera nessuno di questi testi, ma spazierà, integrerà e approfondirà i diversi argomenti, per cui la frequenza è caldamente consigliata.

***DIDATTICA DEL CORSO***

Lezioni di teoria, esercitazioni, assegnazione di esercizi individuali e di gruppo, sviluppo e discussione di videogiochi di esempio quali Battleships e Match-3.

***METODO E CRITERI DI VALUTAZIONE***

L'esame consiste in una prova orale che si compone di due parti:

1. la discussione dell’attività di progetto
2. la verifica della conoscenza degli argomenti trattati durante il corso.

L’attività di progetto corrisponde alla realizzazione di un’applicazione di media complessità in linguaggio Java, dotata di interfaccia grafica, da concordare con il docente, tipicamente un (più o meno semplice) videogioco. L’applicazione dovrà essere “mavenizzata”, ovvero con la gestione delle dipendenze affidata a Maven. Il progetto può essere svolto in gruppo (caldamente consigliato, massimo due persone) o individualmente. L’attività di progetto comprende inoltre la stesura di una breve relazione di documentazione del software in cui, tra le altre cose:

* si descrive il funzionamento dell’applicazione
* si presenta il modello dell’interfaccia grafica
* si elencano i comandi che l’utente può fornire in input
* si descrive l’architettura software adottata, anche attraverso il class diagram UML relativo alle classi principali
* si citano le eventuali librerie utilizzate
* si descrive chiaramente il contributo individuale degli eventuali membri del gruppo.

Il materiale relativo al progetto va consegnato almeno una settimana prima della data in cui uno dei membri del gruppo intende sostenere l'esame. La consegna avviene tramite modalità concordate con il docente, e comprende (almeno):

* i file sorgenti del programma, inseriti nel corrispondente progetto Eclipse basato su Maven
* il file JAR eseguibile del programma generato con Maven
* la relazione di accompagnamento in formato Word/PDF

La discussione del progetto in sede di esame riguarda le scelte progettuali adottate e la loro implementazione, e viene svolta contestualmente da tutti e due i membri del gruppo di lavoro; pertanto, è caldamente consigliato ai membri del gruppo di sostenere l’esame entrambi nello stesso appello.

La verifica della conoscenza degli argomenti del corso avviene attraverso domande di teoria, domande di ragionamento ed eventuale proposta di codice da scrivere e/o analizzare.

Una prova scritta intermedia a metà corso potrebbe essere erogata a discrezione del docente; in tal caso, il suo esito positivo produrrà punti bonus per l’esame, validi per la prima volta in cui si sostiene l’esame e azzerati subito dopo a prescindere dall’esito dell’esame.

***AVVERTENZE E PREREQUISITI***

Il corso non è adatto ad essere un primo corso di programmazione. Lo studente dovrà infatti possedere conoscenze di base di programmazione: nello specifico, si raccomanda che lo studente abbia seguito il corso di Laboratorio di Programmazione, e abbia acquisito familiarità con i concetti fondamentali della programmazione in generale e degli argomenti relativi alla programmazione orientata agli oggetti trattati in quel corso.

*Orario e luogo di orario di ricevimento*

Le modalità di ricevimento studenti sono consultabili alla pagina personale del docente.