

INGEGNERIA DELL'INFORMAZIONE (LB08)

(Lecce - Università degli Studi)

Insegnamento INGEGNERIA DEL SOFTWARE

GenCod A006527

Docente titolare LUCA MAINETTI

Insegnamento INGEGNERIA DEL SOFTWARE

Insegnamento in inglese software engineering

Settore disciplinare ING-INF/05

Corso di studi di riferimento INGEGNERIA DELL'INFORMAZIONE

Tipo corso di studi Laurea

Crediti 6.0

Ripartizione oraria Ore Attività frontale: 54.0

Per immatricolati nel 2022/2023

Erogato nel 2024/2025

Anno di corso 3

Lingua ITALIANO

Percorso CURRICULUM A

Sede Lecce

Periodo Primo Semestre

Tipo esame Orale

Valutazione Voto Finale

Orario dell'insegnamento

<https://easyroom.unisalento.it/Orario>

BREVE DESCRIZIONE DEL CORSO

Il corso intende fornire conoscenze di ingegneria del software e sviluppare le competenze necessarie per progettare e sviluppare programmi e servizi complessi. E' la naturale prosecuzione del corso di Programmazione Orientata agli Oggetti poiché richiede una buona padronanza del linguaggio Java, benché i contenuti teorici siano pienamente autonomi.

Nel concreto, il corso affronta in profondità le tecniche di analisi dei requisiti funzionali e non funzionali, progettazione con UML (Uniform Modeling Language) e sviluppo in Java di sistemi software organizzati secondo architetture standard (design patterns) e livelli di disaccoppiamento che ne facilitino l'evoluzione, con particolare attenzione ai servizi software. Ogni concetto esposto è sperimentato in modo pratico insieme agli studenti, utilizzando il personal computer e strumenti di sviluppo moderni e ampiamente diffusi nel mondo industriale.

PREREQUISITI

Elementi di teoria dell'informatica. Principi e basi per la programmazione. Esecuzione di operazioni seguendo una sequenza codificata di istruzioni elementari. Concetto di algoritmo, algoritmi di base e strutture dati. Concetti di programmazione orientata agli oggetti. Linguaggio Java. Principi di progettazione delle basi di dati.

Propedeuticità del corso: Fondamenti di Informatica e Programmazione Orientata agli Oggetti. Si richiedono conoscenze di Basi di Dati.

OBIETTIVI FORMATIVI

Conoscenze e comprensione. Al termine del corso gli studenti: (a) conosceranno i principi dell'ingegneria del software, in relazione alle caratteristiche e alle qualità del software; (b) conosceranno nel dettaglio le architetture software standard, i design patterns e i principi di riuso del software; (c) conosceranno e sperimenteranno i processi di sviluppo agili del software; (d) approfondiranno linguaggio Java in particolare per la costruzione di architetture software evolvibili; (e) comprenderanno le tecniche di condivisione del software e i sistemi di controllo delle versioni; (f) approfondiranno i principali ambienti di sviluppo Java anche in relazione alle necessità di sviluppo in coppia di applicazioni complesse.

Capacità di applicare conoscenze e comprensione. Gli studenti saranno in grado di applicare le conoscenze acquisite in diversi ambiti applicativi e, in generale, per la codifica al computer in linguaggio Java di applicazioni software con principi di riuso ed evolvibilità.

Autonomia di giudizio. Il corso favorisce l'autonomia di giudizio degli studenti attraverso l'analisi critica di problemi di modellazione del software da requisiti funzionali e non funzionali, per i quali trovare le soluzioni adeguate a risolverli in linguaggio Java con riferimento a noti design patterns. Diverse soluzioni proposte interattivamente dagli studenti saranno poste a confronto e valutate criticamente dagli studenti stessi.

Abilità comunicative. Gli studenti apprenderanno come comunicare adeguatamente e con il corretto livello di formalismo le scelte di design adottate e le strategie di implementazione scelte. Il metodo di insegnamento interattivo e teorico/pratico favorirà momenti di confronto in cui mettere in pratica tali abilità comunicative. Lo sviluppo di prototipi software in coppia richiederà tecniche di comunicazione allo stato dell'arte per l'ingegneria del software

Capacità di apprendimento. La materia in costante evoluzione (le tecniche di sviluppo orientate alle architetture software standard, i linguaggi che le implementano, gli strumenti di sviluppo) richiederà agli studenti la capacità di aggiornarsi e di ricercare materiale on-line, valutandone anche la qualità. Il metodo didattico favorirà l'approfondimento autonomo e in coppie da parte degli studenti, incuriosendoli su tecniche di sviluppo evolute che saranno oggetto del corso magistrale di Progettazione di Architetture di Servizi.

METODI DIDATTICI

Lezioni frontali, esercitazioni pratiche svolte con l'uso del personal computer, elaborazione di esercizi in linguaggio Java e specifici ambienti di sviluppo configurati durante il corso.

MODALITA' D'ESAME

L'esame prevede una prova orale per la verifica dell'apprendimento dei concetti teorici (verifica delle conoscenze) e della capacità di applicazione dei medesimi, in particolare per la codifica di servizi Java complessi (verifica delle competenze). Durante l'esame gli studenti dovranno usare preferibilmente il loro personal computer, configurato con gli ambienti di sviluppo illustrati e utilizzati durante il corso. L'esame orale è singolo e ad ogni candidato/a sarà richiesto di risolvere nel dettaglio un esercizio in linguaggio Java con uso design pattern, MySQL e Spring Boot.

APPELLI D'ESAME

Si veda la bacheca degli appelli su www.dii.unisalento.it

ALTRE INFORMAZIONI UTILI

www.unisalento.it/people/luca.mainetti

PROGRAMMA ESTESO

Presentazione dettagliata del corso e delle modalità d'esame (2 ore). Processi di sviluppo del software (2 ore). Processi agili (2 ore). UML: requisiti e casi d'uso (2 ore). UML: dai requisiti ai diagrammi delle classi, di sequenza e di stato (2 ore). Introduzione alle architetture software standard e ai design patterns (2 ore). Pattern Iterator e Observer (2 ore). Pattern Composite e Strategy (2 ore). Pattern Decorator, Abstract Factory e Factory Method (2 ore). Pattern Singleton e Bridge (2 ore). Pattern Command e Visitor (2 ore). MySQL e SQL: alcuni esempi pratici (2 ore). MySQL e Java (2 ore). Verifica del software (2 ore). Architetture a servizi (2 ore). Architetture a micro-servizi (2 ore). Design pattern per architetture a servizi: pattern Inversion of Control e API Gateway (2 ore). Design pattern per architetture a servizi: pattern Messaging semplice e Publish/Subscribe (2 ore). Design pattern per architetture a servizi: pattern Topic (2 ore). Creare servizi Java con Spring Boot: architettura di sistema e installazione dell'ambiente (2 ore). Dependency Injection e Inversion of Control in Spring Boot (2 ore). Controller e mapping richieste in Spring Web MVC (2 ore). Contenuti dinamici in Spring Web MVC (2 ore). Gestione delle form in Spring Web MVC (2 ore). Accesso ai dati con JDBC in Spring Web MVC (2 ore). Accesso ai dati con JPA in Spring Web MVC (2 ore). Creare e consumare servizi REST in Spring Boot (2 ore).

TESTI DI RIFERIMENTO

- Martin Fowler, "UML distilled. Guida rapida al linguaggio di modellazione standard", Quarta Edizione, Pearson, Maggioli Editore, 2010.
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns: Elementi per il riuso di software a oggetti", Prima Edizione Italiana, Pearson, 2002.
- Ian Sommerville, "Introduzione all'Ingegneria del Software Moderna", Pearson, 2021.
 - David Castelletti, "Creare applicazioni web con Spring Boot", Leanpub, 2021, <https://leanpub.com/creare-applicazioni-web-spring-boot>.