

Percorso Didattico del corso
DecHit Academy:

Full Stack Developer

Indice

OOP, UML, Java SE e Design Pattern	3
Obiettivi	3
Programma	3
Spring core, IoC e DI	4
Obiettivi	4
Programma	4
Database, jdbc, pattern DAO	5
Obiettivi	5
Programma	5
ORM e query mapping e Hibernate	6
Obiettivi	6
Programma	6
Architettura J EE, Web server, Tomcat, Web Application;	6
Obiettivi	7
Programma	7
Javascript, CSS 3, HTML5	8
Obiettivi	8
Programma	8
cenni EJB e WS	10
Obiettivi	10
Programma	10
Angular JS 1; Angular e Typescript	11
Obiettivi	11
Programma	11
Cenni sulle metodologie di sviluppo agile, ad esempio Scrum;	12
Obiettivi	12
Programma	12
Tecnologie Utilizzate	12

OOP, UML, Java SE e Design Pattern

Obiettivi

L'obiettivo in questo modulo è duplice.

Il primo scopo del modulo è ripassare il linguaggio java. Durante la presentazione degli argomenti e lo svolgimento di alcuni esercizi gli allievi hanno modo di provare che non è facile progettare il codice, quindi vengono via via presentate soluzioni ai problemi sempre più eleganti e più robuste utilizzando ad esempio i concetti del design pattern.

L'obiettivo è aumentare la motivazione degli allievi mostrando le migliori soluzioni possibili, per problemi che loro hanno cercato di risolvere precedentemente con poco successo.

Le presentazioni delle soluzioni migliori stimola negli allievi una riflessione che li porta a riconsiderare il loro mestiere come un lavoro di ingegno che richiede attività di progettazione di qualità.

In sintesi gli obiettivi sono

- trasmettere l'importanza di progettare codice di qualità
- migliorare le capacità di analisi
- migliorare la considerazione della attività di programmazione
- conoscere la sintassi Java
- conoscere i pattern principali e soprattutto saperli applicare
- migliorare l'autostima degli allievi e l'entusiasmo con l'obiettivo di migliorare l'approccio alla attività di sviluppo

Programma

- Sintassi del class diagram (UML)
 - classe
 - relazioni tra gli oggetti di tipo has-a e is-a, implementazione, ereditarietà, aggregazione
 - proprietà e funzionalità
 - modificatori di visibilità
- Sintassi di Java
 - classi, interfacce
 - metodi, metodi costruttori
 - gestione delle eccezioni
 - ereditarietà multipla, singola,
 - classe, classe astratta, interfacce
 - static e non static
 - concetto di contratto in merito all'uso delle interfacce
- Design Pattern
 - Singleton
 - Decorator
 - Composite
 - Strategy
 - Chain of Responsibility
 - Template Method
 - Proxy, Wrapper, Delegate
 - Abstract Factory
 - Builder

Spring core, IoC e DI

Obiettivi

In questo modulo si presenta il framework Spring, dando importanza soprattutto all'aspetto architetturale delle applicazioni, all'organizzazione del progetto e all'astrazione che ne deriva dall'utilizzo di Spring. Tutti gli esercizi del primo modulo attinenti il Design Pattern vengono riproposti utilizzando Spring con l'obiettivo di far apprezzare ancora di più i pattern, l'astrazione e la qualità del codice prodotto. Si rafforzano ancora di più alcuni degli obiettivi della prima parte.

Programma

- L'incapsulamento e il problema dell'accoppiamento tra le classi
- Inversion of Control
- Dipendence Injection
- Factory e container
- Progetti Spring e Spring Core
- Ripasso polimorfismo
- Esempio di applicazione con Spring
- creazione oggetti con costruttore, property e factory method
- init e destroy method
- lazy init
- scope dei bean singleton e prototype
- Rivisitazione dei Pattern già visti utilizzando Spring
- Aspect Oriented Programming
- AOP con Spring

Database, jdbc, pattern DAO

Obiettivi

Conoscere la struttura di un database relazionale e dei suoi componenti (record, tabelle, relazioni, chiavi indici) e i comandi SQL (DML, DDL) per poter operare con i database. Vengono spiegate le classi Java per l'uso di JDBC e la connessione con i database.

Tra gli obiettivi c'è anche quello della progettazione e realizzazione di un database di piccole dimensioni in autonomia. Prove di estrazione dati con JDBC.

Particolare attenzione viene posta alla organizzazione architettuale del progetto tramite pattern DAO e bean DTO.

Programma

- Database
 - modello E/R
 - cardinalità 1:1, 1:N, N:M
 - architettura RDBMS
 - SQL: istruzioni DML, DDL
 - DML: select, distinct, where condition, operatori, insert, update
 - clausola order by
 - group by e having
 - join: i vari tipi
 - subquery: single row e multiple row
 - subquery correlate
 - DDL: create, alter e drop; table, view
 - grant e revoke
 - transazioni: commit, rollback, savepoint
- JDBC
 - Connection
 - gestione della connessione
 - JDBC Driver
 - Lookup Dipendence
 - utilizzo di sql embedded in java
 - statement e prepared statement
 - result set
 - pattern architettuale DAO

ORM e query mapping e Hibernate

Obiettivi

Viene trattato il framework Hibernate come ORM per fornire strumenti per gestire la persistenza ad alto livello. Particolare attenzione viene posta alla flessibilità che offre l'utilizzo di hibernate e ad alcuni vantaggi architetturali che ne derivano. Ancora una volta gli allievi vengono invitati ad organizzare il progetto in modo elegante e il più robusto possibile.

Programma

- Utilizzo del database
- Strumenti ORM
- Framework Hibernate
- Configurazione di hibernate
- Mapping delle classi java
- Utilizzo dei Dialetti SQL
- Generator type id, primary key
- Tipi di Hibernate
- UserType
- Interceptors
- Criteria
- HSQL
- SQL Native
- Named Query
- cardinalità one-to-one, many-to-many, many-to-one
- politiche di caricamento, lazy e eager
- caching di primo e secondo livello

Architettura J EE, Web server, Tomcat, Web Application;

Obiettivi

Vengono introdotti l'importanza e i ruoli del Web Server e del Web Server Application. Vengono spiegate le classi Servlet e i Filter, il ciclo di vita, e il Pattern MVC per la realizzazione di applicazioni ben strutturate. Vengono inoltre trattate anche le tag library e la loro integrazione nelle pagine.

In questo modulo attraverso la realizzazione di alcune Servlet e di alcune JSP si realizza una applicazione web minimale che utilizza il database progettato nel modulo precedente. Gli allievi mettono insieme le parti del progetto sperimentate nei moduli precedenti costruendo una applicazione completa. Viene, in alcuni punti sottolineata, ancora una volta l'importanza di alcune scelte architetturali.

Programma

- Funzionamento di una applicazione web
- Web Server e Web Application
- Application Server: Utilizzo di Tomcat, Web Container
- Struttura di una applicazione web, deployment descriptor, parametri
- Container J EE
- Servlet (ciclo di vita, funzionamento, thread)
- Filters, esempi di applicazione
- Il pattern architetturale MVC
- JSP, direttive principali, java Beans, jsp expression
- Gestione della sessione utente
- Gestione della conversazione
- framework Spring MVC

Javascript, CSS 3, HTML5

Obiettivi

L'obiettivo di questa parte, più semplice rispetto al resto del corso, è quella di dare tutti gli strumenti utili per poter operare agevolmente in attività inerenti il front end di una applicazione web. L'obiettivo quindi non è quello di una trattazione esaustiva degli argomenti ma quello di fornire agli allievi un buon grado di autonomia. Vengono presentati tutti i tipi di selettori CSS, e la maggior parte dei tag HTML e la sintassi del linguaggio Javascript, particolare attenzione viene posta alle novità di CSS 3 e HTML 5.

Infine viene presentato il framework bootstrap 4 concentrandosi soprattutto su alcuni aspetti come il miglioramento grafico di alcune parti già svolte precedentemente, l'approccio responsive, e la cura dell'aspetto grafico delle applicazioni web.

Programma

- HTML, HTML5
 - sintassi html
 - problemi di compatibilità dei browser
 - utilizzo dei tag principali, div, table, img, p, span
 - sintassi, uso dei tag, utilizzo degli attributi
 - attributi di stile, attributo identificatore,
 - gestione colori,
 - links, immagini, tabelle, liste,
 - form, elementi delle form con tipi e attributi,
 - media, video, audio,
 - gestione degli eventi;
- CSS3
 - sintassi delle regole di stile
 - Css in line, file esterno
 - tipi di selettori (tag, elemento, attributo, classe, identificativo, pseudoclasse),
 - gestione colori,
 - background,
 - borders, margins, padding, height, width
 - box model
 - text, font, icons, links, list, tables,
 - position, align, opacity, tooltips, forms, buttons
 - rounded corners,
 - border images, Images
 - colors, gradients,
 - fonts,
 - animations, transition, transforms,
 - pagination
- Javascript
 - tipi predefiniti
 - prototype
 - gestione eventi

- gestione eventi temporali setInterval, setTimeout
- validazione form lato client
- Bootstrap 4
 - la griglia
 - l'approccio responsive
 - gli stili degli elementi più importanti: button, list, alert
 - tooltip, popover
 - carosello di immagini
 - utilizzo di jquery in bootstrap

Cenni EJB e WS

Obiettivi

Obiettivo di questo modulo è far comprendere come oggi si può organizzare l'architettura di una applicazione distribuita sia orientata ai servizi che non orientata ai servizi. Si inizia innanzitutto presentando degli scenari dove possono essere utili le soluzioni proposte. Come al solito particolare importanza viene posta alle scelte architettoniche e alle differenze progettuali tra le varie soluzioni descritte.

Programma

- Tipi di EJB e utilizzi tipici
 - Entity
 - Session
 - Statefull e Stateless
 - creazione di istanze, pool di ejb
- Web Service (Tipi e differenze)
 - Rest
 - SOAP

Angular JS 1; Angular e Typescript

Obiettivi

L'obiettivo è quello di far comprendere l'utilità dell'utilizzo del framework Angular JS oggi all'interno di una applicazione web. Viene presentata la sintassi e la nuova metodologia di sviluppo che il framework promuove. Per comprendere bene Angular viene presentato anche il linguaggio Typescript

Programma

- Configurazione
- MVC Architecture
- Direttive
- Expressions
- Controllers
- Filters
- Tables
- HTML Dom
- Modules
- Form
- includes
- View
- Scope
- Services
- Custom directive
- Sintassi di Typescript
 - oggetti
 - variabili
 - strutture di controllo
 - programmazione OOP in Typescript

Cenni sulle metodologie di sviluppo agile, ad esempio Scrum;

Obiettivi

Viene spiegato il concetto di ciclo di vita del software nelle varie fasi di Analisi, Progettazione, Implementazione, Test e Deploy. Cenni sulle problematiche di gestione del progetto in merito alla gestione dei rilasci.

Si introducono delle metodologie per riuscire a stimare le attività inerenti il proprio lavoro di programmazione e delle tecniche per ottimizzare la gestione delle attività di sviluppo e del tempo impiegato.

Programma

- Gestione e organizzazione progetti
- Metodologie di sviluppo e organizzazione delle attività
- Valutazione dei tempi di una attività
- Ciclo di vita del software e strumenti software a supporto
- Figure coinvolte in un progetto IT
- Gestione delle relazioni all'interno del gruppo di lavoro
- metodologia Scrum

Tecnologie Utilizzate

- Installazione Web Server e Web Container Apache-Tomcat
- ORM Hibernate
- Container Spring
- IDE: Eclipse, Visual Studio Code
- linguaggi: Java, XML, CSS, HTML, Javascript.
- installazione e configurazione Web Container: Apache Tomcat.
- installazione e configurazione db MySQL