



INGEGNERIA DEL SOFTWARE

AUTUNNO 2024

SINTESI AUTOMATICA DELLA LEZIONE 1, 19 SETTEMBRE 2024

BASE DI CONOSCENZA: <https://research.constructor.tech/platform/knowledge-models/44814c2adf2c42d9958c5de096336237/chat>

REPOSITORY GITHUB: <https://github.com/GiancarloSucci/UniBo.SE.A2024>

1 CAVEAT

Come largamente specificato in classe, questo riassunto è fatto in modo automatico e quindi potrebbe contenere errori, anche significativi, o addirittura essere fuorviante. In caso di dubbio gli studenti devono riferirsi al materiale spiegato in classe

2 INTRODUZIONE AL CORSO DI INGEGNERIA DEL SOFTWARE

Giancarlo Succi introduce il corso di ingegneria del software, spiegando che una parte delle lezioni sarà tenuta online e registrata, utilizzando varie piattaforme per la gestione del corso. Egli presenta una piattaforma sperimentale basata sull'intelligenza artificiale e spiega il metodo di calcolo del voto d'esame. Giancarlo fornisce istruzioni agli studenti su come accedere al materiale del corso attraverso un repository Git e sottolinea l'importanza di registrarsi sulla piattaforma per ottenere l'accesso.

3 SVILUPPO DI SISTEMI SOFTWARE E PRINCIPI DI PROGRAMMAZIONE

Giancarlo discute con gli studenti i principi fondamentali della programmazione, concentrando su concetti come modularità, organizzazione in sistemi e sottosistemi, e information hiding. La discussione si approfondisce sull'accessibilità e la visibilità del codice, con Daniele e altri studenti che partecipano attivamente condividendo le loro conoscenze su classi, interfacce e strutture dati in diversi linguaggi di programmazione. Giancarlo conclude riflettendo sulla possibilità di nascondere elementi nell'ambito dello sviluppo di un sistema software.

4 INFORMATION HIDING NEI LINGUAGGI DI PROGRAMMAZIONE

Giancarlo discute l'importanza dell'information hiding nella programmazione, spiegando come questo concetto sia stato implementato in linguaggi come Ada e Modula-2. Egli sottolinea che il codice stesso è la migliore forma di documentazione e che l'astrazione è un elemento chiave dell'information hiding. Giancarlo menziona anche l'importanza di commentare il codice in modo efficace e introduce brevemente il linguaggio di programmazione Smalltalk.

5 ASTRAZIONE DEI DATI

Giancarlo discute l'importanza dell'astrazione e del "data abstraction", chiedendo ai partecipanti di riassumere le loro conoscenze precedenti su questo argomento. Lorenzo fornisce una definizione di "data abstraction", e Giancarlo spiega l'origine latina della parola "abstraction".

6 ASTRAZIONE E ACCOPPIAMENTO

Giancarlo discute l'importanza dell'astrazione nell'ingegneria del software, fornendo esempi dalla vita quotidiana, e sottolinea il ruolo della documentazione e del codice ben scritto. Egli introduce il concetto di "coupling" (accoppiamento) tra moduli software, utilizzando analogie culinarie per illustrare diversi livelli di complessità, e spiega le due forme principali di accoppiamento: tra funzioni di moduli diversi e di dati. Giancarlo evidenzia che l'accoppiamento di dati è particolarmente critico, poiché le modifiche alle strutture dati interne possono richiedere cambiamenti alle strutture esterne, aumentando la complessità del codice.

7 GESTIONE DEL COUPLING ED EREDITARIETÀ

Giancarlo discute vari aspetti della programmazione, concentrandosi sui rischi del coupling e dell'ereditarietà. Egli sottolinea l'importanza di gestire attentamente le dipendenze nei progetti software, evidenziando i pericoli dell'ereditarietà multipla e delle assunzioni sul funzionamento interno del sistema operativo. Giancarlo illustra questi concetti con esempi pratici, tra cui un caso di incompatibilità tra architetture RISC e CISC, e conclude affermando che, sebbene sia impossibile eliminare completamente il coupling, è fondamentale saperlo gestire in modo efficace.

8 COESIONE

Giancarlo spiega il concetto di coesione nella programmazione, illustrando i diversi tipi come strutturale, logica, temporale, procedurale, comunicativa, sequenziale e funzionale, fornendo esempi pratici come l'uso di classi anonime in Java. La discussione si estende all'importanza della coesione nello sviluppo software, evidenziando come questi tipi si riferiscano a parti di codice con uno scopo comune o una sequenza logica di input e output. Giancarlo menziona anche l'importanza della depenalizzazione e la sua complessità in relazione al governo.

9 SEMPLICITÀ E PROBLEMI CON L'APPROCCIO "DESIGN FOR REUSE"

Giancarlo critica l'approccio "Design for Reuse" nello sviluppo software, evidenziando i problemi di spreco di tempo, aumento della complessità e costi di manutenzione. Enfatizza l'importanza della semplicità e della modificabilità nel codice, sottolineando la necessità di scrivere codice essenziale, effettuare regolarmente il refactoring e mantenere metodi brevi e interfacce pubbliche limitate. Lorenzo e Diego supportano la discussione, evidenziando gli svantaggi di sviluppare funzionalità non richieste.

10 INTRODUZIONE A UML

Giancarlo riassume i principali concetti di progettazione software, tra cui la suddivisione in sottosistemi, l'information hiding e la coesione, e introduce l'Unified Modeling Language (UML) come standard per la modellazione del software. Egli annuncia che il corso si concentrerà su una parte specifica dell'UML per comprendere la progettazione dei sistemi orientati agli oggetti. Giancarlo conclude informando gli studenti sulla prossima lezione e sulla disponibilità per domande.