

INTRODUZIONE AL CORSO

Ingegneria del Software
a.a. 2018/19

Orario del corso

	08.00	09:00	10:00	11:00	12:00	13:00	15:00
Lunedì	Fondamenti di Elettronica (Mosca)	Fondamenti di Elettronica (Mosca)	Basi di Dati e progettazione del SW (Seidita) (*)	Basi di Dati e progettazione del SW (Seidita) (*)	Basi di Dati e progettazione del SW (Seidita) (*)		Controlli Automatici (Fagiolini)
Martedì			Basi di Dati e progettazione del SW (Seidita) F230	Basi di Dati e progettazione del SW (Seidita)	Reti di Calcolatori e Internet (De Paola)		Controlli Automatici (Fagiolini)
Mercoledì		Basi di Dati e progettazione del SW (Seidita) (*)	Basi di Dati e progettazione del SW (Seidita) (*)	Controlli Automatici (Fagiolini)	Controlli Automatici (Fagiolini)		Reti di Calcolatori e Internet (De Paola)
Giovedì	Fondamenti di Elettronica (Mosca)	Fondamenti di Elettronica (Mosca)	Fondamenti di Elettronica (Mosca)	Reti di Calcolatori e Internet (De Paola)	Reti di Calcolatori e Internet (De Paola)		
Venerdì	Fondamenti di Elettronica (Mosca)	Basi di Dati e progettazione del SW (Seidita)	Basi di Dati e progettazione del SW (Seidita)	Basi di Dati e progettazione del SW (Seidita)			

Ricevimento e contatti

- Appuntamento solo per e-mail

valeria.seidita@unipa.it

- Giorno di ricevimento:
 - lunedì 9:00-10:00 (da scheda di trasparenza)
 - ufficio: (ex) Dipartimento di Ingegneria Informatica, ed. 6, tel. 091 238**62626**, stanza 4

Obiettivi del corso

- Acquisizione delle nozioni di base per la progettazione del software:
 - Metodologie di progettazione
 - Modellazione del software
 - Linguaggi di modellazione
 - Raccolta dei requisiti
 - Architettura del software
 - Testing
 - Project management

Argomenti del corso

- Progettazione di una base di dati
 - progetto concettuale
 - progetto logico
 - interfacciamento con Java
- Vedere scheda trasparenza

Argomenti del corso

ORE	Lezioni
2	Introduzione all'Ingegneria del Software. Concetti di: progetto, attività, risorsa, task, workproduct, sistema, modello, documento, obiettivi (goal), requisiti, vincoli, notazioni, metodi e metodologie
2	Le fasi principali dello sviluppo: raccolta requisiti, analisi dei requisiti, progetto di sistema, progetto esecutivo o degli oggetti, implementazione, gestione del progetto, testing, ciclo di vita del software.
6	Introduzione all'UML, diagrammi dei casi d'uso, diagrammi delle classi, diagrammi di sequenza e collaborazione. Diagrammi di stato diagrammi di attività, diagrammi di dislocazione, organizzazione dei diagrammi, estensione dei diagrammi, concetti di sistema, modello e vista. Modellazione object-oriented.
6	Introduzione alla raccolta dei requisiti. Concetti fondamentali: requisiti funzionali, requisiti non funzionali e pseudo-requisiti, livelli di descrizione. Principali attributi delle specifiche (correttezza, completezza, ...). Classificazione delle attività di raccolta dei requisiti.
6	Analisi dei requisiti: Identificazione degli attori, degli scenari, dei casi d'uso, dei casi d'uso, delle relazioni tra attori e casi d'uso. Identificazione degli oggetti d'analisi, identificazione dei requisiti non funzionali. Modelli di analisi: funzionale, degli oggetti, dinamico. Concetti di analisi: oggetti di tipo entità, di confine e di controllo. Passaggio dai casi d'uso agli oggetti, identificazione degli oggetti entità, di confine, di controllo, modellazione delle interazioni, identificazione delle associazioni e degli attributi, modellazione del comportamento degli oggetti.
5	Progettazione di sistema. Introduzione, concetti e attività principali. Architetture software: client-server peer to peer, pipe and filter. Attività della progettazione di sistema. Identificazione dei sottosistemi. La mappatura dei sottosistemi su processori e componenti. La definizione dei depositi di dati. La definizione dal controllo l'accesso. La progettazione del flusso di controllo: procedure-driven, event-driven, threads. La progettazione delle condizioni di confine.

Argomenti del corso

4	<p>Progettazione di dettaglio.</p> <p>Introduzione, concetti della progettazione del modello ad oggetti.</p> <p>Attività della progettazione del modello oggetti; oggetti d'analisi e oggetti della soluzione. Tipi, signature, visibilità.</p> <p>Contratti: invarianti, pre-condizioni, post-condizioni.</p> <p>Object Constraint Language.</p> <p>Identificazione degli attributi e delle operazioni mancanti; specifica di tipi, signature e visibilità.</p> <p>Specifica dei vincoli (constraint), delle eccezioni; identificazione e adattamento delle librerie di classi; realizzazione delle associazioni. Incrementare il riuso.</p>
3	<p>Testing.</p> <p>Introduzione al testing; tecniche di controllo della qualità; tecniche per evitare guasti; tecniche per la scoperta dei guasti tecniche per tollerare i guasti.</p> <p>Concetti di test: componente, guasto, errore, malfunzionamento, test case, test stub/driver, correzione.</p> <p>Attività di testing: ispezione dei componenti; unit testing; test di integrazione; test di sistema.</p> <p>- test di integrazione: strategie del test di integrazione (big bang, bottom-up, top-down, sandwich, modified sandwich).</p> <p>- test di sistema: test funzionale, test di performance, test pilota (alpha test, beta test), test di accettazione e test di installazione.</p> <p>Pianificazione del test; documentazione del test.</p>

ORE	Lezioni
3	<p>Project Management.</p> <p>Elementi fondamentali di project management</p> <p>Caratteristiche fondamentali del progetto</p> <p>Attività (ordinarie, di riepilogo, cardine). Struttura delle attività. Relazioni tra attività. Le risorse (il calendario, i costi). I costi fissi del progetto. Le relazioni di progetto.</p> <p>Il progetto iniziale.</p> <p>Il progetto con previsioni.</p> <p>Il progetto con variazioni.</p>

Testi consigliati

- B. Bruegge, A. H. Dutoit. Object-Oriented Software Engineering: Using UML, Patterns and Java. Third edition. International edition. ISBN: 0-13-606125-7. Prentice Hall
- Ian Sommerville Ingegneria del software 8/Ed. 2007 pp. 848 ISBN 9788871923543
- M. Fowler. UML Distilled. Pearson
- Jim Arlow, Ila Neustadt. UML2 e Unified Process - analisi e progettazione Object Oriented, Addison-Wesley

Come studiare

- Quando possibile le slide saranno fornite in anticipo tramite il sito del corso (portale studenti)
- PRENDETE APPUNTI (anche direttamente sulle slide)
- I libri di testo VANNO STUDIATI e integrati con gli appunti
- Le esercitazioni vanno seguite e ripetute in maniera individuale ed in gruppo

IMPORTANTE

- Modalità d'esame:
 - **tesina da concordare ed orale**
- Iscrizione corso tramite portale studenti
- Scheda di trasparenza
- Calendario didattico

... verso alcuni
concetti di base

Si scriva un programma in java che legga un insieme di dati in virgola mobile. Al termine, si stampi il conteggio dei valori, la media e la deviazione standard. La media di un insieme di valori x_1, \dots, x_n è data da

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n}$$

La deviazione standard è espressa da questa formula

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

Dichiarare una classe con due metodi (i nomi sono facoltativi). Il primo riceve in ingresso un numero intero e restituisce vero se il numero è pari e falso se il numero è dispari. Il secondo riceve in ingresso una stringa e restituisce un numero intero. Utilizzare i due metodi in una applicazione nella quale si richiede di digitare un numero, si usa il secondo metodo per leggerlo e si restituisce un messaggio che indica che il numero è pari o dispari.

Si scriva un programma in linguaggio java che sia in grado di gestire l'archivio ed i prestiti di una videoteca; le funzionalità previste sono le seguenti

- fornire all'utente, quando richiesto, l'elenco dei titoli disponibili con il relativo codice di riferimento
- gestire una richiesta di prestito inserita attraverso il codice o il titolo del film. Esempio: l'utente chiede la lista dei film, sceglie e seleziona l'opzione "richiesta", inserisce il titolo e/o il codice ed il sistema risponde in funzione della disponibilità del film stesso. Se il film è disponibile il sistema darà una risposta di conferma e il numero di film disponibili altrimenti darà un messaggio negativo ed inoltre se la data di consegna di almeno una copia è uguale a quella della richiesta se ne deve avvisare l'utente.

Si supponga che

1. le informazioni relative ai titoli dei film siano memorizzate in un array di stringhe di dimensione tre contenente codice, titolo e quantità
2. le informazioni relative ai prestiti siano memorizzate in un array di stringhe di dimensione due contenente il codice e la data di consegna
3. esiste un sistema di modifica della lista dei film da non implementare
4. esiste un metodo *time()* della classe *Calendar* che restituisce la data corrente sotto forma di stringa del tipo *aammgg* (lo stesso formato della data memorizzata nell'array dei prestiti).
5. il cliente non commette errori di inserimento dati

Si richiede la progettazione di un sistema di gestione di uno sportello bancomat.

Descrizione generale del sistema

Il sistema di cui è richiesta la progettazione deve fornire le seguenti funzionalità:

Presso una postazione bancomat ogni utente inserisce la sua carta ed accede ai servizi bancari tramite autenticazione (inserimento codice bancomat).

I servizi a disposizione dell'utente sono i seguenti:

- 1) Prelievo
- 2) Ricarica cellulare
- 3) Lista movimenti

L'utente che sceglie di effettuare un prelievo può selezionare uno tra sei possibili importi, dopo la selezione dell'importo viene richiesta la convalida o l'annullamento dell'operazione ed eventualmente viene erogato il denaro ed una ricevuta, quest'ultima su richiesta.

L'utente che effettua una ricarica può selezionare il gestore telefonico di cui è cliente e l'importo della ricarica (anche qui la scelta avviene tra sei possibili importi), confermare o annullare l'operazione, previa presa visione dei dati inseriti, e ricevere una ricevuta.

La stampa della lista movimenti avviene su uno scontrino nel quale vengono elencati gli ultimi dieci movimenti effettuati con le relative specifiche: importo, data, tipo di movimento (prelievo, assegno, pagamento bancomat, etc.)

Tesine

- Gestione documentazione esami universitari (15/16)
- Gestione portale conference management system (16/17)
- Software di gestione azienda autotrasporti cittadina (17/18)
- ??? (18/19)
 - tesina standard, tesina personalizzata in ambito robotico

Domanda	Risposta
Cos'è il software?	Programmi per computer e relativa documentazione. I prodotti software possono essere sviluppati per un particolare cliente o per il mercato in generale.
Cos'è l'ingegneria del software?	L'ingegneria del software è una disciplina ingegneristica che si occupa di tutti gli aspetti della produzione del software.
Qual è la differenza tra l'ingegneria del software e l'informatica?	L'informatica si occupa della teoria e dei fondamenti; l'ingegneria del software degli aspetti pratici che riguardano lo sviluppo di software di qualità.
Qual è la differenza tra l'ingegneria del software e l'ingegneria dei sistemi?	L'ingegneria dei sistemi ha come oggetto tutti gli aspetti dello sviluppo di sistemi informatici, inclusi quelli hardware, software e di processo. L'ingegneria del software è una parte di questo processo.
Cos'è il processo software?	È un insieme di attività il cui scopo è lo sviluppo o la modifica del software.
Cos'è un modello di processo software?	È una rappresentazione semplificata di un processo software visto da uno specifico punto di vista.
Quali sono i costi dell'ingegneria del software?	All'incirca il 60% dei costi è legato allo sviluppo, il 40% alle prove. Per il software personalizzato, i costi di evoluzione spesso superano quelli di sviluppo.
In cosa consistono i metodi di ingegneria del software?	Approcci strutturati di sviluppo software, che comprendono modelli di sistema, notazioni, regole, consigli di progettazione e guide al processo.
Cos'è CASE (<i>computer-aided software engineering</i> , ingegneria del software assistita dal computer)?	Sono sistemi software creati per fornire aiuti automatizzati alle attività di lavorazione del software. I sistemi CASE sono spesso utilizzati come ausilio ai metodi.
Quali sono le caratteristiche di un buon software?	Il software deve fornire le funzionalità e le prestazioni richieste dall'utente, e deve essere mantenibile, fidato e usabile.
Quali sono le sfide chiave che l'ingegneria del software si pone?	Affrontare la crescente diversità e sviluppare software affidabile in tempi sempre più ridotti.