

Modelli classici di sviluppo di sistemi informatici

Introduzione, ingegneria e ciclo di vita del software, modello a cascata.

Le fasi del ciclo di vita del software.

Modelli di sviluppo: modello a cascata, modello a prototipazione rapida, modello incrementale e a spirale. Sviluppo “agile” o iterativo incrementale del software, conclusioni.

Il modello di sviluppo OOP (Object Oriented Programming)

Perché tanti linguaggi di programmazione. Crisi, dimensioni e qualità del software. Leggi catastrofiche. Le cause della crisi del software. Astrazione, oggetti e classi, Dov'è la novità?

Conclusione: che cos'è la programmazione a oggetti.

Documentazione di un progetto, attori, casi d'uso e scenari

Documentare un progetto a OOP. Le schede CRC per le classi e le associazioni tra di esse.

Introduzione e descrizione dei casi d'uso. Relazioni tra casi d'uso. La documentazione dei casi d'uso. Esempi di diagrammi UML dei casi d'uso.

Tecniche di programmazione Concorrente in Linux

Ripasso sulla gestione dei files in C.

Programmi che ricevono parametri dalla Linea di Comando e processi in Linux.

Sincronizzazione dei processi e semafori software: sincronizzazione di due processi tramite una variabile intera condivisa, che funga da semaforo, inizializzata tramite la funzione mmap.

Gestione dei processi con la creazione di Thread in linguaggio C. Le funzioni della libreria pthread: pthread_create, pthread_join, pthread_exit.

Programmazione grafica in linguaggio JAVA, con l'uso di FORM, BUTTON, LABEL e la gestione degli eventi.

Laboratorio: In laboratorio sono state svolte le esercitazioni sul programma svolto in classe. I programmi sono stati sviluppati nel linguaggio C, JAVA con i comuni editor di testo offerti dal SO Linux e compilati da riga di comando.