

**PROGRAMMA DEL CORSO DI
METODI AVANZATI DI PROGRAMMAZIONE
Corso di Laurea in Informatica
A.A. 2007/2008**

Obiettivi. Il corso si propone di fornire le conoscenze di base della programmazione e della modellazione orientata agli oggetti. Assumendo che il discente conosca il paradigma di programmazione imperativo, si utilizza l'astrazione nella programmazione come chiave di volta per introdurre i concetti fondamentali del paradigma orientato a oggetti (oggetti e classi). Il passaggio dalla programmazione imperativa a quella basata su oggetti (object-based) è spiegato mediante la nozione di dato di prima classe. Ulteriori concetti fondamentali del paradigma orientato a oggetti (ereditarietà e polimorfismo) sono illustrati come estensioni dei corrispondenti concetti di sottotipo e di coercizione/overloading presenti nella programmazione imperativa. Nel corso si introdurrà anche la notazione UML per la modellazione grafica dei diversi concetti del paradigma object-oriented. Attraverso l'introduzione del linguaggio di programmazione Java e lo sviluppo guidato di un progetto completo si approfondisce la progettazione orientata a oggetti, la composizione di classi, l'uso di gerarchie di classi ed alcune strutture dati fondamentali, lo sviluppo di applicazioni client-server, lo sviluppo di interfacce grafiche.

Prerequisiti: conoscenze di programmazione imperativa, algoritmi e strutture dati.

Modalità d'esame: prova scritta e discussione orale del progetto di laboratorio. La documentazione del progetto va consegnata in anticipo rispetto alla data concordata per la prova orale .

Programma del corso a.a. 2007-2008

1. Introduzione ai paradigmi di programmazione.

I tre approcci alla programmazione: operazionale, definizionale e dimostrazionale.

2. L'astrazione nella programmazione

Fondamenti: Introduzione all'astrazione. Astrazione di funzione, di procedura, di controllo, e di selettore. Astrazione di tipo e tipi astratti di dato. Specifiche algebriche e assiomatiche per i tipi astratti di dato. I moduli per l'incapsulamento dell'informazione e l'information hiding. Oggetti e classi di oggetti. Astrazione di dati: Tipo astratto di dato vs. classe di oggetti. Astrazione generica.

Ambienti e linguaggi di programmazione.

I moduli in Modula-2, Turbo Pascal, C e Ada.

3. La programmazione orientata agli oggetti.

Fondamenti: oggetti, classi concrete, classi astratte, metaclassi, ereditarietà singola ed ereditarietà multipla, polimorfismo, gerarchia di classi e gerarchia di interfacce. Composizione di classi. Confronto tra ereditarietà e composizione nel riuso del software.

Ambienti e linguaggi di programmazione.

Java: caratteristiche generali del linguaggio; Java e Internet; Java vs. C++. Ambienti di sviluppo Java. Oggetti in Java: costruttori; distruttori; metodi, argomenti e valori di ritorno. Controllare il flusso di esecuzione: uso degli operatori Java; il controllo di esecuzione; l'inizializzazione. Nascondere le implementazioni: i package; i modificatori di accesso; le interfacce. Il riuso delle classi in Java: ereditarietà, derivazione protetta; polimorfismo. I contenitori: array; collezioni; le nuove collezioni. Approfondimenti su Java: il trattamento delle eccezioni; identificazione di tipo al run-time; programmazione generica in Java; il sistema I/O di Java. Connessione con le Basi di Dati: JDBC. Creazione di interfacce per applicazioni: il package SWING. Programmazione in rete: socket, invocazione di metodi remoti, applet. Il multithreading: creazione di classi attive; sincronizzazione nell'accesso dei metodi.

Laboratorio.

Esercitazioni guidate su: L'ambiente Eclipse. Progetto di applicazioni con singole classi; progetto di applicazioni con più classi organizzate gerarchicamente e in package; progetto di applicazioni con classi astratte e uso del polimorfismo; progetto di applicazioni con contenitori e trattamento delle eccezioni; progetto di applicazioni con I/O da file; progetto di connessione a database; progetto di applicazioni con GUI mediante con SWING; progetto di applicazioni client-server e multithreading, progetto di applet.

4. La complessità computazionale.

Misure statiche e dinamiche. Modelli di costo. Funzioni di complessità. Ordini di complessità. Relazioni di ricorrenza e analisi di algoritmi ricorsivi. Complessità nel caso ottimo, medio e pessimo. Classi di complessità. Tesi della computazione sequenziale. Analisi di complessità dei seguenti algoritmi di ordinamento: selectsort, bubblesort, insertsort, quicksort, mergesort. Analisi di complessità dell'algoritmo di ricerca binaria. Analisi di complessità dell'algoritmo per il calcolo dell'n-esimo numero di Fibonacci. Introduzione alla complessità di problemi. Complessità del problema dell'ordinamento. Algoritmi ottimi in ordine di grandezza.

Principali testi e articoli di riferimento

1. Introduzione ai paradigmi di programmazione.

A.L. Ambler, M.H. Burnett, & B.A. Zimmerman
Operational Versus Definitional: A Perspective on Programming Paradigms
IEEE Computer, 25(9): 28-43, September 1992.

2. Astrazione dati.

E. Lodi, & G. Pacini
Introduzione alle Strutture di Dati (cap. 3-4)
Bollati Boringhieri, 1990.

M. Shaw
Abstraction Techniques in Modern Programming Languages
IEEE Software, 10-26, October 1984.

D. A. Watt
Programming Language Concepts and Paradigms (cap. 5-6)
Prentice Hall, 1990.

3. La programmazione orientata agli oggetti.

G. Masini, A. Napoli, D. Colnet, D. Léonard, & K. Tombre
Linguaggi per la Programmazione a Oggetti (cap. 2-3, 6)
Gruppo Editoriale Jackson, 1989

Bruce Eckel
Thinking in Java, 4th Edition (cap. 1-11, 13-14, 16-17, 19-20, 23-24)
Prentice-Hall, 2006

4. La complessità computazionale.

A.A. Bertossi
Algoritmi e Strutture di Dati (Introduzione, cap. 6.1, 11.1, 11.2, 16.1)
UTET, 2000

Trasparenze del corso disponibili su:

<http://www.di.uniba.it/~malerba/courses/map/>