
Metodi Avanzati di Programmazione (II Anno)

a.a. 2011/2012

Docente: Annalisa Appice

Dipartimento di Informatica

Università degli Studi Bari Aldo Moro

tel.: 080 5443262

email: appice@di.uniba.it

<http://www.di.uniba.it/~appice/>

Informazioni generali

Sito del corso

<http://www.di.uniba.it/~appice/courses/1112/index.htm>

Password per l'apertura delle dispense: **map10**

Orario delle lezioni:

Martedì 15:00 – 18:00

Giovedì 15:00 – 18:00

Venerdì 9:00 – 11:00 (Esercitazioni guidate in laboratorio)

Obiettivo del corso

- Il corso mira ad approfondire il paradigma di programmazione orientato a oggetti.
- Si parte dal paradigma di programmazione imperativa studiata nei corsi di *Programmazione, Linguaggi di Programmazione, Architettura degli Elaboratori*,
- si approfondiscono i concetti di astrazione studiati in *Algoritmi e Strutture Dati*,
- per giungere a formalizzare i concetti di *oggetto* e *classe* nell'ambito del *paradigma imperativo*,
- quindi inquadrarli nel *paradigma basato su oggetti*,
- e infine studiarli nel *paradigma orientato a oggetti*, insieme ai concetti di *ereditarietà* e *polimorfismo*.

Obiettivo del corso

- Oltre ad illustrare i fondamenti teorici del paradigma orientato a oggetti, il corso mira a presentare alcuni strumenti operativi che supportano il paradigma di programmazione orientato a oggetti, ovvero i linguaggi di programmazione.
- Particolare enfasi sarà posta sul linguaggio Java. Le lezioni in aula saranno coordinate con le esercitazioni in laboratorio.

Obiettivo del corso

- ***Prerequisiti***: conoscenze di programmazione imperativa, algoritmi e strutture dati.
- ***Modalità d'esame***: prova scritta + caso di studio.
- ***Prove in itinere*** con valenza di esonero e validità per l'AA 2011-2012
- ***Progetto***: unico per tutto il corso. Può essere svolto in gruppi di al **più tre persone**.

Programma del corso

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.

Programma del corso

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.

Programma del corso

3. La programmazione orientata agli oggetti (cont.)

Ambienti e linguaggi di programmazione (cont.)

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; il sistema I/O di Java. Creazione di interfacce per applicazioni: il package SWING. Java Generics. Java Database Connectivity (JDBC). Programmazione di applicazioni client-server. Creazione di applet: il ciclo di vita di un applet; controllare il layout; la gestione degli eventi, multithreading.

Programma del corso

4. Esercitazioni guidate in laboratorio

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 applicazioni con GUI mediante SWING; progetto di applicazioni client-server, progetto di applet.

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.

Principali testi e articoli di riferimento

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.

Principali testi e articoli di riferimento

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, 2nd Edition (cap. 1-9, 13)

Prentice-Hall, 2000