PROGRAMMA DEL CORSO DI

PROGRAMMAZIONE II

Prof. Donato Malerba

Obiettivi. Il corso approfondisce alcune tecniche di programmazione utilizzabili nello sviluppo di applicazioni function-oriented, in cui la complessità prevalente del sistema riguarda le funzioni da realizzare. In particolare sono analizzate le tecniche di programmazione orientata agli oggetti e logica, legate a nuovi paradigmi di programmazione che si sono affermati in diversi contesti. L'obiettivo è duplice: da una parte, illustrare i fondamenti teorici dei paradigmi, per i benefici di rigore e sistematicità che ne derivano, e dall'altra presentare alcuni strumenti operativi che supportano i diversi stili di programmazione, ovvero i linguaggi di programmazione.

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

Programma del corso a.a. 2001-2002

1. Introduzione ai paradigmi di programmazione.

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

2. Il paradigma imperativo.

   Fondamenti: la manipolazione sequenziale di dati in memoria. La struttura dei programmi, i dati (dichiarazioni, tipi primitivi, composti e ricorsivi, equivalenza dei tipi), le espressioni, i comandi, la gestione del flusso di controllo, la composizione (blocchi e sottoprogrammi), legame statico e dinamico.

   Ambienti e linguaggi di programmazione.

   Linguaggio C: caratteristiche generali, operatori, operazioni di I/O, funzioni, tipi strutturati, puntatori. Linguaggio Ada: caratteristiche generali, istruzioni di controllo, operatori, procedure e funzioni, tipi di dati.

3. Astrazione.

   Fondamenti: L'astrazione nella programmazione. 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.

4. 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.

   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; il sistema I/O di Java. Creazione di interfacce per applicazioni: il package SWING. Creazione di applet: il ciclo di vita di un applet; controllare il layout; la gestione degli eventi. Gestione dei thread: il ciclo di vita di thread; il blocco di un thread; gestione delle priorità.

   Laboratorio.

   Esercitazioni guidate su: 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 SWING; progetto di applet; progetto di applicazioni/applet con gestione di thread.

5. La programmazione logica.
   Fondamenti: programmare per dimostrazioni, clausole e programmi definiti, le interrogazioni, semantica dei modelli e modello minimo di Herbrand, semantica del punto fisso e caratterizzazione di punto fisso del minimo modello di Herbrand, unificazione di termini, risoluzione binaria e proprietà, semantica operazionale e la risoluzione SLD. La negazione in programmazione logica: negazione per fallimento, completamento di un programma definito, correttezza della regola NF.
   Ambienti e linguaggi di programmazione logica.
   Prolog: dalla programmazione logica al Prolog, la ricerca depth-first, sintassi del Prolog, liste e operazioni su liste, il principio di invertibilità, gli operatori, la valutazione di espressioni aritmetiche, il cut, implementazione della negazione. Caso di studio: il problema della scimmia e della banana.

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. Il paradigma imperativo.

   H. E. Bal, D. Grune
   Programming Languages Essentials (cap. 1-2)
   Addison-Wesley, 1994

3. 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.

4. 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

5. La programmazione logica.

   U. Nilsson, & J. Maluszynski
   Logic, Programming and Prolog (cap. 1-5)
   Wiley, 1990

   I. Bratko
   Prolog Programming for Artificial Intelligence, 3rd edition
   Addison-Wesley, 2000.