

**INSEGNAMENTO DI INGEGNERIA DEL SOFTWARE B**  
**CORSO DI LAUREA SPECIALISTICA IN**  
**INGEGNERIA INFORMATICA**

Docente

**Marina Zanella**

Orario ricevimento: lunedì 10.30-11.30  
venerdì 9.30-10.30

Dip. Elettronica per l'Automazione

Studio 28, Tel. 030 3715.596

e-mail: [zanella@ing.unibs.it](mailto:zanella@ing.unibs.it)

## **Testi adottati**

E. GAMMA, R. HELM, R. JOHNSON, J. VLISSIDES, DESIGN PATTERNS: ELEMENTI PER IL RIUSO DI SOFTWARE A OGGETTI, ADDISON – WESLEY, PRIMA EDIZIONE ITALIANA, 2002 (ANCHE SU CD IN VERSIONE INGLESE DEL 1998)

## **Testi per la consultazione**

A. BINATO, A. FUGGETTA, L. SFARDINI, INGEGNERIA DEL SOFTWARE – CREATIVITÀ E METODO, PEARSON EDUCATION ITALIA, 2006  
(WebBook disponibile al sito <http://webbook.cefriel.it>)

B. BRUEGGE, A. H. DUTOIT, OBJECT-ORIENTED SOFTWARE ENGINEERING USING UML, PATTERNS, AND JAVA, 2/E, PEARSON EDUCATION, 2004

C. GHEZZI, M. JAZAYERI, D. MANDRIOLI, INGEGNERIA DEL SOFTWARE – FONDAMENTI E PRINCIPI, 2/E, PEARSON EDUCATION ITALIA, 2004  
(Copie dei lucidi disponibili al sito <http://www.prenhall.com/ghezzi>)

C. S. HORSTMANN, PROGETTAZIONE DEL SOFTWARE E DESIGN PATTERN IN JAVA, APOGEO, 2004

C. LARMAN, APPLICARE UML E I PATTERN, 3/E, PEARSON EDUCATION ITALIA, 2005

L. A. MACIASZEK, B. L. LIONG, PRACTICAL SOFTWARE ENGINEERING: A CASE STUDY APPROACH, ADDISON-WESLEY, 2005

S. J. METSKER, DESIGN PATTERN IN JAVA, PEARSON EDUCATION ITALIA, 2003

R. S. PRESSMAN, PRINCIPI DI INGEGNERIA DEL SOFTWARE, 4/E, MCGRAW-HILL, 2004

I. SOMMERVILLE, INGEGNERIA DEL SOFTWARE, 7/E, PEARSON EDUCATION ITALIA, 2005

### **Altre fonti**

Copie dei **lucidi** del docente disponibili al sito <http://eracle.ing.unibs.it/seb>

## ESAME

- Prova **scritta** (da superare necessariamente con votazione  $\geq 18/30$ )
- Presentazione e discussione di un **progetto** proposto dal docente

Le due prove sono sostenibili in qualsiasi ordine relativo

Linguaggio di programmazione di riferimento: Java

Il progetto assegnabile può essere

1) Tecnico: manutenzione (correttiva e adattiva, se è il caso) + refactoring + testing (N.B. tutte e tre le attività sono obbligatorie e devono essere documentate) di un programma prodotto nell'ambito degli insegnamenti di *Ingegneria del Software A* o *Algoritmi e Strutture Dati*; è realizzato, individualmente o in gruppo (massimo quattro persone), dagli autori del programma

oppure

2) Didattico + Tecnico: è realizzato da un singolo studente;

Parte didattica: creazione/aggiornamento di supporti didattici relativi a un *pattern di progettazione*; presentazione in aula del pattern stesso

Pattern di riferimento: Observer, Composite, Decorator, Abstract Factory, Model-View-Controller

Parte tecnica: testing (documentato) di un programma prodotto dallo studente nell'ambito degli insegnamenti di *Ingegneria del Software A* o *Algoritmi e Strutture Dati*

