

Ingegneria del Software T

Corso di Laurea Triennale in
Ingegneria Informatica
III anno – A.A. 2011/2012

Premessa

Una domanda fondamentale

- ✱ Che cosa significa scrivere del buon software?

- ✱ Risposta del programmatore C:

“Scrivere del buon software significa ottimizzare ogni istruzione, in modo da ottenere il codice più compatto ed efficiente possibile”

- ✱ Risposta del programmatore Visual Basic:

“Scrivere del buon software significa fornire le funzionalità richieste dall’utente nel minor tempo possibile e col minor costo possibile, indipendentemente da come si arriva al risultato”

Risposta dell'ingegnere del software

Scrivere del buon software significa

- ✱ Trovare il miglior equilibrio fra diversi fattori:
 - ✱ La soddisfazione dell'utente
 - ✱ La facilità di estensione dell'applicazione
 - ✱ La comprensibilità delle soluzioni adottate
- ✱ Adottare tecniche adeguate a gestire la crescente complessità delle applicazioni

Risposta dell'ingegnere del software

Scrivere del buon software significa

- ✱ Utilizzare al meglio l'investimento, spesso ingente, necessario per produrre un'applicazione, garantendo in particolare:
 - ✱ Il maggior tempo di vita possibile
 - ✱ Il riutilizzo in altri progetti di parte del codice prodotto

Ingegnere del Software

Compiti principali

- ✱ Affrontare in modo sistematico e misurabile:
 - ✱ il progetto
 - ✱ la realizzazione
 - ✱ l'utilizzo
 - ✱ la manutenzionedei prodotti software
- ✱ Studiare le strategie per realizzare il punto precedente

Ingegneria del Software

Di cosa si occupa

- ✓ Gestione del processo di sviluppo del software
- ✓ Attività di analisi
- ✓ Attività di progettazione
- ✓ Attività di codifica
- ✓ Attività di verifica e convalida (testing)
- ✱ Attività di tipo gestionale
 - ✱ Stime dei costi (e dei tempi)
 - ✱ Gestione dei progetti (delle persone, pianificazione)
 - ✱ Gestione dei rischi
 - ✱ Gestione della qualità
- ✱ Metriche

Obiettivi del corso

- ✱ Fornire i concetti di base dell'ingegneria del software
- ✱ Esaminare e utilizzare tecniche *object-oriented* per l'analisi, il progetto e la realizzazione di applicazioni software

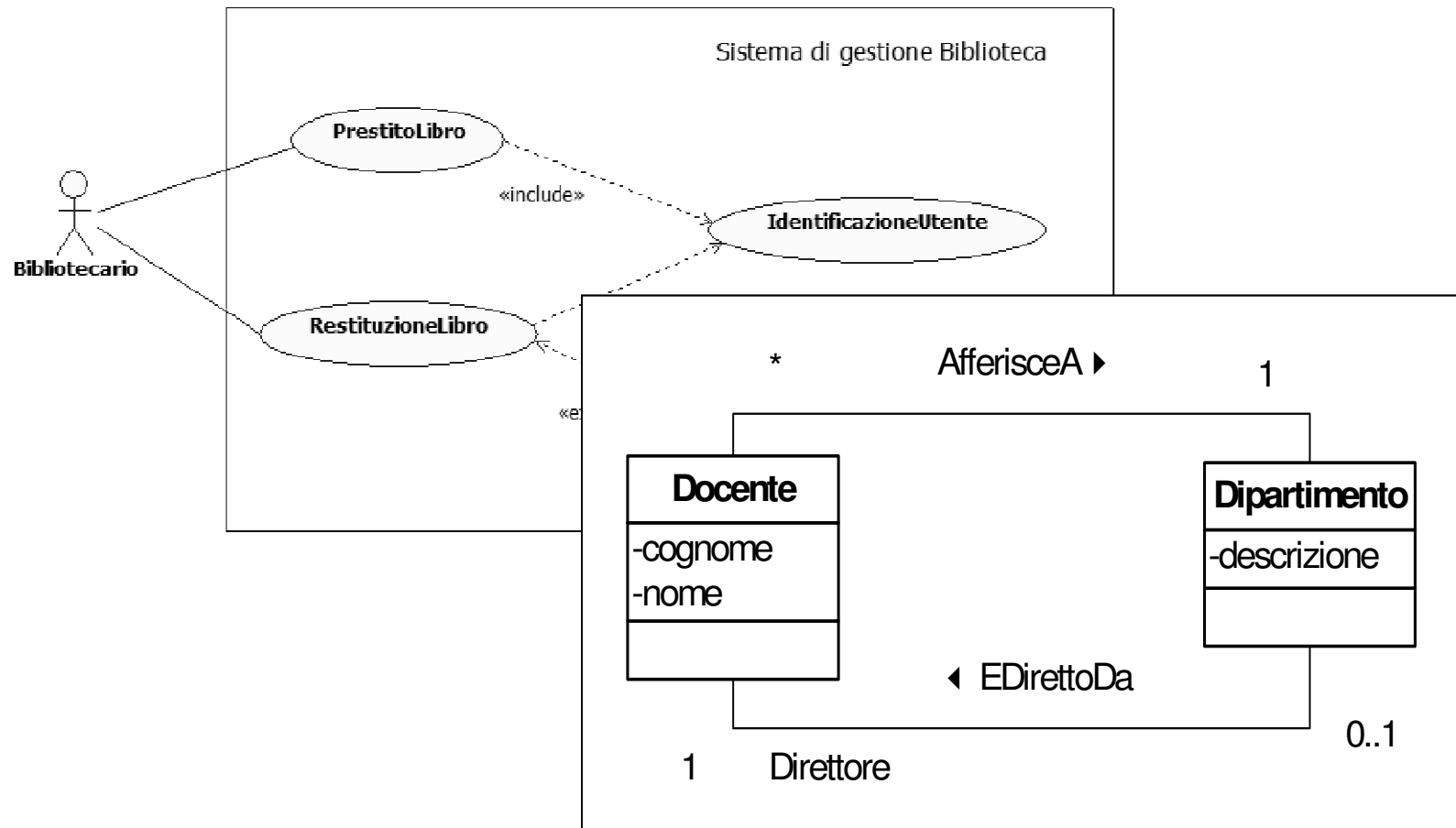
Obiettivi del corso

- ✱ Fornire nozioni avanzate sulla tecnologia .NET
 - ✱ *Framework .NET*
 - ✱ Linguaggio C#
 - ✱ Delegati, eventi
 - ✱ Attributi, introspezione (*reflection*)
 - ✱ *Garbage collector*
 - ✱ Interfacce utente (cenni)
 - ✱ Accesso ai dati (cenni)
 - ✱ ADO .NET
 - ✱ ORM (*Object-Relational Mapping*)

Linguaggio di modellazione

- ✱ Durante il processo di sviluppo del software è indispensabile poter utilizzare un linguaggio per costruire i modelli da discutere con il cliente e gli altri sviluppatori
- ✱ Il linguaggio dovrebbe
 - ✱ essere visuale (una figura è meglio di mille parole)
 - ✱ possedere capacità dichiarative
- ✱ UML - *Unified Modeling Language*

Linguaggio di modellazione



Design Pattern

- ✱ Durante la progettazione è indispensabile conoscere e utilizzare i design pattern, al fine di:
 - ✱ risolvere problemi progettuali specifici
 - ✱ rendere i progetti *object-oriented* più flessibili e riutilizzabili
- ✱ Ogni design pattern
 - ✱ cattura e formalizza l'esperienza acquisita nell'affrontare e risolvere uno specifico problema progettuale
 - ✱ permette di riutilizzare tale esperienza in altri casi simili

Testi consigliati

- ✿ Slide viste a lezione
- ✿ *C. Larman*, **Applicare UML e i pattern**
Analisi e progettazione orientata agli oggetti
(terza edizione), Prentice Hall, 2005
- ✿ *E. Gamma, R. Helm, R. Johnson, J. Vlissides*,
Design Patterns – Elements of Reusable Object-
Oriented Software, Addison Wesley, 1998

Bibliografia

- ✱ *M. Fowler*, **UML Distilled** (4a edizione italiana), Addison Wesley, 2010
- ✱ *S. Bennett, J. Skelton, K. Lunn*, **Introduzione a UML**, McGraw-Hill, 2002
- ✱ *W. Zuser, S. Biffl, T. Grechenig, M. Köhle*, **Ingegneria del software con UML e Unified Process**, McGraw-Hill, 2004
- ✱ *J. Arlow, I. Neustadt*, **UML e Unified Process – Analisi e progettazione object-oriented**, McGraw-Hill, 2003

Bibliografia

- ✿ *I. Sommerville*, **Ingegneria del software** (8a edizione), Addison Wesley, 2008
- ✿ *R. S. Pressman*, **Principi di Ingegneria del Software** (quinta edizione), McGraw-Hill, 2008

Software

★ MSDN Academic Alliance



Utilizzabile per la modellazione UML



Microsoft Visio 2010



Microsoft Visual Studio 2010

Esiste anche una versione Express più 'leggera' – scaricabile gratuitamente dal sito MS

Orario delle lezioni

Martedì 14-16	14.15 – 15.45
Giovedì 14-16	14.15 – 15.45
Venerdì 14-17	14.15 – 15.45 16.00 – 16.45

Esame

- ✱ Ingegneria del Software T (9 CFU)
 - ✱ Orale (2/3) + Progetto (1/3)
 - ✱ Quiz (2/3) + Progetto (1/3)

- ✱ Ingegneria del Software L-A (6 CFU)
 - ✱ Orale (1/2) + Progetto (1/2)
 - ✱ Quiz (programma 2011/12)