

Lezione 0

Introduzione al corso

Vittorio Scarano
Corso di Programmazione Distribuita (2003-2004)
Laurea di I livello in Informatica
Università degli Studi di Salerno

Organizzazione della lezione

- Struttura del corso
- Alcune informazioni di carattere pratico
- Alcuni consigli

Programmazione Distribuita (2003-2004), Vittorio Scarano

2

Programmazione Distribuita

- Struttura del corso:
 - una veloce introduzione al linguaggio Java
 - i sistemi distribuiti e le problematiche connesse alla loro programmazione
 - un metodo per poter programmare oggetti distribuiti in Java
 - *Remote Method Invocation*
 - cenni su alcuni framework per la programmazione distribuita in Java (come Jini, JXTA, JavaSpace, etc.)

Programmazione Distribuita (2003-2004), Vittorio Scarano

3

Informazioni di carattere pratico

- Lezioni:
 - il mercoledì ed il venerdì (ore 9-11) aula C-38
- Esame:
 - scritto ed orale
 - prova in itinere a metà corso
 - esonera sullo scritto della prima metà del corso
- Esercitazioni (libere)
 - laboratorio Reti (richiesta account in classe)
- Infoline:
 - <http://www.dia.unisa.it/vitsca.dir/PD>
 - vitsca@unisa.it

Programmazione Distribuita (2003-2004), Vittorio Scarano

4

Libri di testo (1)

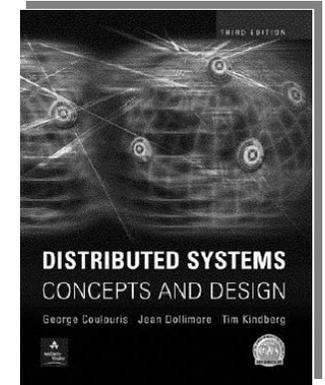
- Java:
 - Qualsiasi libro su Java (manuale) va bene
 - Riferimento a "*Java 2: i fondamentali*" di Cay Horstmann e Gary Cornell. (Seconda Edizione), Mc Graw Hill.
 - Ulteriore documentazione disponibile sul sito
 - Strumenti di sviluppo per Java:
 - editor e linea di comando (ok)
 - BlueJ: sistema integrato, Pure-Java, con editor, debugger, e sistema run-time (<http://www.bluej.org>)
 - Eclipse: sistema professionale di sviluppo, open-source, editor, debugger completo e sistema run-time (<http://www.eclipse.org>)



5

Libri di testo (2)

- Sistemi Distribuiti:
 - "*Distributed Systems: Concepts and Design*" di George Coulouris, Jean Dollimore, Tim Kindberg.
 - Ed. Addison-Wesley
 - primi 5 capitoli
- Remote Method Invocation
 - documentazione SUN disponibile sul sito
- Altri sistemi di sviluppo:
 - documentazione SUN disponibile sul sito



6

Dal Syllabus: conoscenze pregresse

- Lo studente deve avere:
 - una buona conoscenza di programmazione ad oggetti
 - una conoscenza operativa di un sistema operativo tra Windows Microsoft e Linux (o Unix) per poter sviluppare il codice richiesto.
- Una conoscenza pregressa del linguaggio Java può essere di aiuto anche se è prevista, comunque, una breve introduzione al linguaggio.
- Lo studente dovrà seguire contemporaneamente il corso di Reti (che si tiene nello stesso semestre) in quanto, dopo il primo terzo del corso si useranno conoscenze su protocolli di rete, TCP/IP, indirizzi IP, socket TCP e UDP, etc..

7

Dal Syllabus: obiettivi e risultati attesi

- Principi di programmazione distribuita con oggetti distribuiti
 - Applicazione tramite Remote Method Invocation di Java
- Principi progettuali del linguaggio Java e sulle sue caratteristiche per applicazioni distribuite.
- Architetture dei sistemi distribuiti e sulla nomenclatura generale utilizzata per denotare le componenti.
- Problematiche tipiche di sistemi distribuiti a oggetti (marshalling, serializzazione, naming e localizzazione).
- Paradigmi client-server e peer-to-peer.

8

Dal Syllabus: struttura del corso (1)

- **Introduzione al linguaggio Java (16 ore)**
 - Le strutture fondamentali del linguaggio
 - Programmazione OO ed Ereditarietà in Java
 - Namespace
 - Eccezioni e Thread
- **Java nei Sistemi Distribuiti (8 ore)**
 - Concorrenza e distribuzione in Java: Server e handlers
 - Socket in Java (Socket, stream e filtri)

9

Dal Syllabus: struttura del corso (2)

- **Introduzione ai Sistemi Distribuiti (8 ore)**
 - Architetture dei sistemi distribuiti
 - Da Remote Procedure Call a Remote Invocation Method
- **Java Remote Invocation (16 ore)**
 - Architettura di Java RMI
 - Il Registry di RMI
 - Garbage Collection distribuita, logging dei metodi e factory di oggetti
 - Oggetti attivabili
 - Architetture distribuite client-server e peer-to-peer con RMI
 - Progettazione di applicazioni distribuite di esempio

10

Dal Syllabus: modalità di esame

- Per studenti che seguono il corso
 - 2 prove intercorso svolte al termine di ciascuno dei due periodi di lezioni, ed un colloquio finale.
 - I prova intercorso: in test a risposta multipla ed alcune domande di programmazione (scrittura di semplici programmi in Java)
 - si intende superata se si ottiene almeno una votazione di 14/30.
 - II prova intercorso è principalmente costituita da una applicazione distribuita in Java RMI.
 - richiesta in aula la progettazione ed implementazione, anche se in maniera parziale.
 - Colloquio orale
 - Discussione di tutti gli argomenti della seconda parte del corso
 - Programma scritto durante la prova scritta, funzionante, sia in formato cartaceo (listato, diagrammi, etc) che digitale (floppy).
 - quanto più simile a quello consegnato alla prova scritta.

11

Dal Syllabus: modalità di esame

- Per studenti che **non** seguono il corso
 - 1 prova scritta:
 - in test a risposta multipla
 - alcune domande di programmazione (scrittura di semplici programmi in Java)
 - applicazione distribuita in Java RMI.
 - Colloquio orale
 - Discussione approfondita di tutti gli argomenti del corso
 - Necessario portare il programma scritto durante la prova scritta, funzionante, sia in formato cartaceo (listato, diagrammi, etc) che digitale (floppy).
 - quanto più simile a quello consegnato alla prova scritta.

12

Dal Syllabus: responsabilità dello studente

- Gli studenti sono responsabili di studiare ed approfondire gli argomenti spiegati in classe utilizzando il materiale didattico indicato dal docente.
- Lo svolgimento delle esercitazioni e la frequenza del corso sono fortemente consigliate.
- Gli studenti devono essere preparati a trascorrere una buona quantità di tempo nello studio al di fuori delle lezioni, essendo necessaria una notevole attività di programmazione.
- Una preparazione soddisfacente richiede in media due ore di studio per ciascuna ora trascorsa in aula

13

Alcuni consigli utili....

5 Maniere diverse per farsi bocciare!

- 5. Non studiare “lezione per lezione”
– “*Tanto in un paio di giorni recupero...*”
- 4. Studiare solo da appunti e slide, *mai* dai libri di testo!
– “*Altrimenti perché il professore li ha messi sul sito?*”
- 3. Non programmare mai, MAI, **MAI!**
– al massimo, compilare 1 volta “*HelloWorld.java*”
- 2. Non venire a lezione,
– ma se proprio dovete... allora veniteci a dormire
- 1. Non studiare

14