

Ingegneria Dei Sistemi Distribuiti



- Materiale, link utili, avvisi
- <http://www.dmi.unict.it/tramonta/sdm>

1

Prof. Tramontana - Ottobre 2020

Programma

- Tecniche per sviluppare un sistema software distribuito o parallelo, di grandi dimensioni, che deve andare in produzione
- Buone norme di progettazione di tali sistemi
 - Design Pattern [Gamma]: Proxy
 - Design Pattern [Buschmann]: Remote Proxy, Forward-Receiver, Broker

3



Lezioni

- In Aula 23
 - Martedì 11:00–13:00
 - Giovedì 11:00–13:00
- Coprono tutto il programma del corso
- Partecipazione fortemente consigliata: si impara di più, e meglio, è possibile fare domande ed ottenere risposte!

2

Prof. Tramontana - Ottobre 2020

Programma

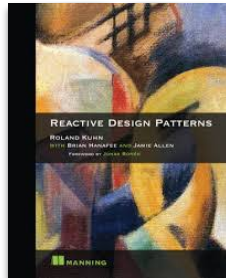
- Design Pattern [Fowler]: Remote Facade, Data Transfer Object, Session State, Serialized LOB, Gateway, Registry
- Design Pattern per la sicurezza [Schumacher]: Access Control (Reference Monitor), Authenticator, Firewall

4



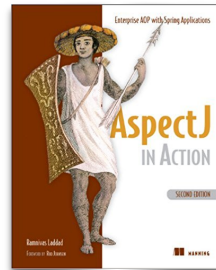
Programma

- Aspect-Oriented Programming (AOP)
- Time-Out, Latenze e Design Pattern Circuit Breaker e Bulkhead



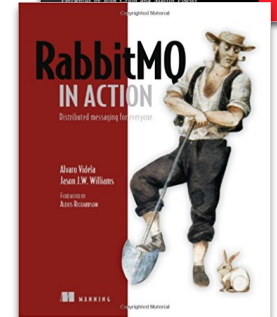
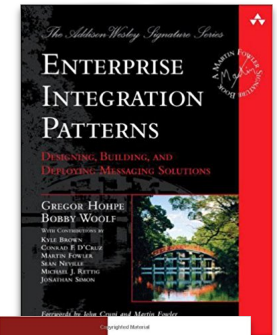
5

Prof. Tramontana - Ottobre 2020



Programma

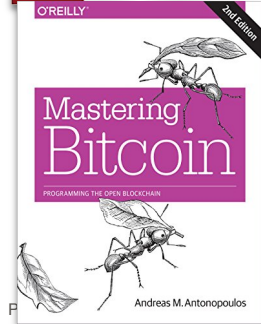
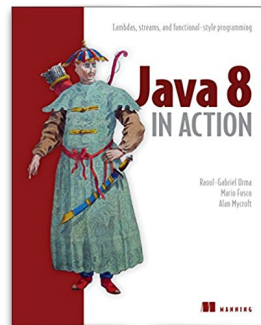
- Middleware orientato al Message-Passing, RabbitMQ
- Design pattern: event-driven consumer, competing consumers, publish-subscribe, message-filter,



6

Programma

- Java CompletableFuture per programmazione asincrona e timeout
- Architettura distribuita di una blockchain per il consenso e per la registrazione di transazioni



7

Libri Consigliati

- Le slide sono di supporto
- Gli appunti presi a lezione, insieme alle slide potrebbero bastare
- Libri
 - Buschmann, et al. Pattern-Oriented Software Architecture. A System of Patterns. Volume 1. Wiley, 1996
 - Fowler. Patterns of Enterprise Application Architecture. Addison-Wesley, 2003
 - Gamma, et al. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994
 - Hohpe, et al. Enterprise Integration Patterns. Addison-Wesley, 2003
 - Kuhn, et al. Reactive Design Patterns. Manning, 2017
 - Laddad. AspectJ in Action. Manning, 2nd ed, 2010
 - Schumacher, et al. Security Patterns. Wiley, 2006
 - Urma, et al. Java 8 in Action: Lambdas, streams, and functional-style programming. Manning, 2015
 - Videla, et al. RabbitMQ in Action. Manning, 2012

8

Prof. Tramontana - Ottobre 2020

Modalità Esami

- Progetti individuali o per coppie + Orali
- I Progetti saranno concordati, a partire da circa metà del corso
 - Si dovrà produrre
 - Relazione con: descrizione requisiti, diagrammi UML, dettagli di progettazione e implementazione
 - Codice applicazione e codice test
- Valutazione
 - Chiarezza della documentazione
 - Modularità del codice (alta coesione, basso accoppiamento), uso di design pattern