Laboratorio Reti di Calcolatori

Laurea Triennale in Comunicazione Digitale

Anno Accademico 2013/2014

Account Provvisorio

User: RETI

Psw: 2013

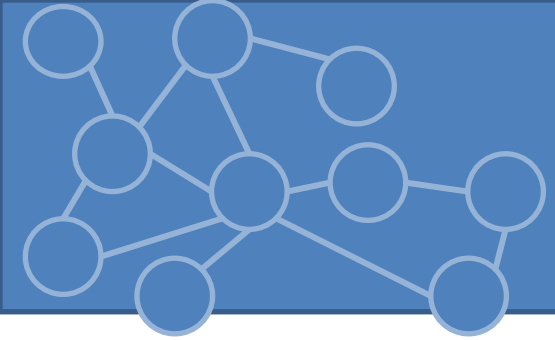
Accedi a : CSD

172.16.19.24/registrati

Aspetti Organizzativi

- Orario Laboratorio
  - Mercoledì 13:45 - 16:45 (Aula 309 – Settore Didattico) A -L
  - Giovedì 13:45 - 16:45 (Aula 309 – Settore Didattico) M-Z

- Contatto: matteo.zignani@unimi.it
  reti@di.unimi.it
- Ricevimento: appuntamento via email presso NPTLab (S230 Comelico)



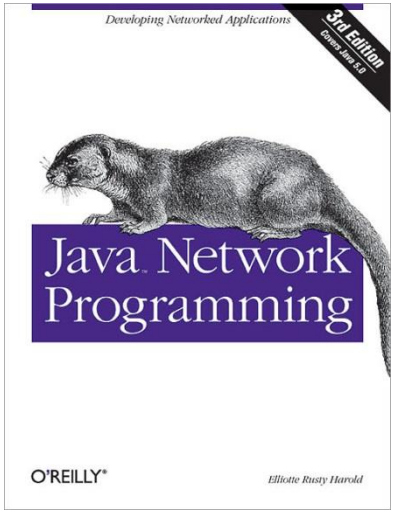
Programma

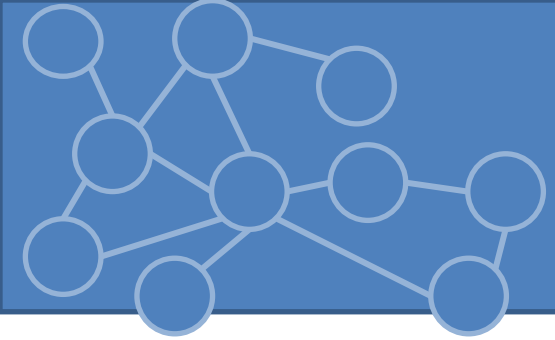
1. Introduzione ad Eclipse
2. Collection e Streams
3. Threads
4. Programmazione con socket TCP
    1. Server iterativo
    2. Server multithread
5. URL e HTTP Connection
6. Protocol e Content Handler
7. XML, JSON
8. API di alcune social networks (Twitter, Facebook)
    1. OAuth
9. Grafi e libreria Graphstream

Libri consigliati

- Dario Maggiorini, ‘Introduzione alla programmazione client-server’, Pearson Editore 2009

- Elliotte Rusty Harold, ‘Java Networking Programming’, O’Reilly Media 2004





Riferimenti Utili

- Documentazione Java SE 6
  - http://download.oracle.com/javase/6/docs/api/
- Slide del corso
  - http://reti.di.unimi.it
  - Caricate prima della lezione in cui vengono utilizzate
  - Servono come riferimenti per preparare l'esame, NON sono un libro di testo !!

Prerequisiti

1. Conoscenza base del linguaggio di programmazione Java.
2. Compilazione ed esecuzione bytecode
3. Consultazione documentazione Java => conoscenza inglese
4. Prerequisito morale: il laboratorio serve per apprendere e sviluppare le abilità richieste dal programma, non serve per sviluppare abilità sociali e ludiche.

Tools

- In laboratorio
  - Eclipse Indigo
  - WinXP
- A casa
  - Si consiglia ultima versione di Eclipse, ma qualsiasi altro tool è ammesso

Modalità lezione

- Mercoledì pomeriggio: spiegazione + serie di esercizi con soluzione dopo # minuti.
- Giovedì pomeriggio: lo stesso

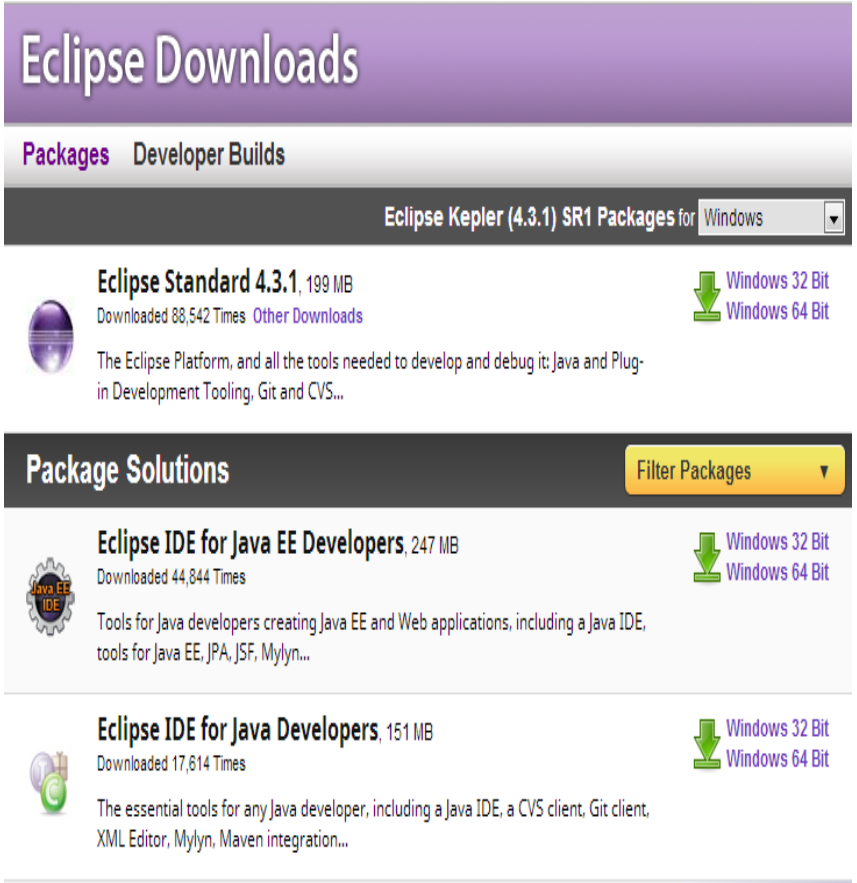
Eclipse

- Eclipse: IDE per sviluppo software con supporto di diversi linguaggi Java, C, C++, Python, Ruby, mediante plugin

Download:
http://www.eclipse.org/downloads/

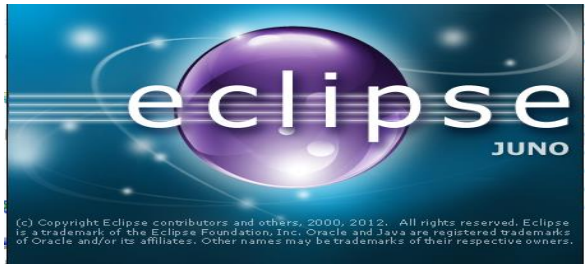
- Eclipse Classic
- Eclipse IDE for Java Developers

Ultima versione: Kepler (4.3.1)

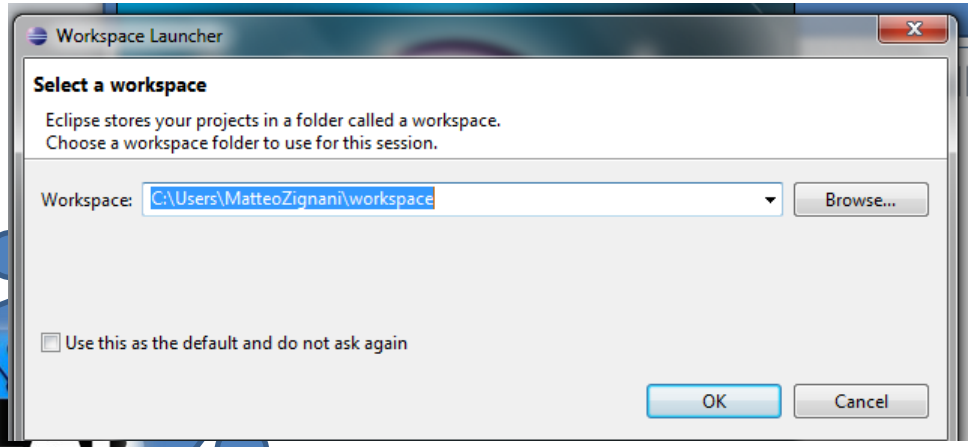


La prima volta in Eclipse

- Supponiamo che JRE e JDK siano già installate
- Unzip della cartella contenente Eclipse



- Settaggio del workspace



Esercizi guidati

- Si crei un progetto in Eclipse denominato ‘SocialBook’. Si aggiungano le librerie in formato .jar scaricabili dal sito del corso.
- Si aggiunga il package org.reti.element al progetto ‘SocialBook’.
- Si aggiunga il package org.reti.factory al progetto ‘SocialBook’.
- Si aggiunga il package org.reti.social al progetto ‘SocialBook’.

Esercizi guidati 2

- Nel package org.reti.element si crei la classe *Libro.*
- Si inseriscano i seguenti campi
  - Titolo: String
  - Insieme di autori: ArrayList<*Autore*>
  - ISBN: String
  - Anno di pubblicazione: String
- Si crei il costruttore della classe *Libro* senza argomenti e con tutti gli argomenti.
- Si crei il costruttore della classe *Libro* che accetta un numero variabile di oggetti *Autore*.

Esercizi guidati 3

- Si creino i metodi di get e set per tutti i campi della classe *Libro.*
- Si creino i metodi *toString()* e *equals()* della classe Libro.
- Si crei la classe Autore definita dai campi
  - nome, secondo nome e cognome: String.
- Si creino i costruttori, i metodi di set e get e i metodi *toString()* e *equals()*.
- Si crei un metodo main nella classe Autore per testare il funzionamento delle classi appena create.

Esercizi guidati 4

- Si crei la classe *MagazzinoLibri* con i campi
  - magazzino: Hashmap<String,Libro> =>chiave isbn e valore un oggetto Libro
  - nome: String
- Si implementino il costruttore e i metodi
  - Libro *getLibro(String isbn)* : restituisce un libro se presente nel magazzino, altrimenti solleva l'eccezione *LibroNotFoundException*
  - void *addLibro(Libro l)* : aggiunge un libro al magazzino
  - void *deleteLibro(String isbn)* : elimina un libro dal magazzino

Esercizi guidati 5

- Si crei una classe *Libreria* sempre nel package org.reti.element caratterizzata dai campi
  - miaLibreria: TreeSet<Libro>
  - Nome: String
- Si implementino il costruttore e i metodi
  - *void addLibro(Libro l)*: aggiunge un libro alla libreria
  - *void deleteLibro(Libro l)*: rimuove un libro dalla libreria

Esercizi guidati 6

- Si crei la classe *User* nel packae org.reti.social caratterizzata dai campi
  - username, password, nome, cognome: String
  - Amici: ArrayList<User>
  - libriMiei : Libreria
- Si definiscano il costruttore, i metodi di set e get solo sugli attributi di tipo String e Libreria e i metodi
  - *void addAmico(User)*: aggiunge un utente come amico
  - *void deleteAmico(User)*: rimuove un mio amico

Esercizi guidati 7

- Si crei la classe *SocialBook* nel package org.reti.social caratterizzata dai campi
  - Utenti: HashMap<String,User>
- Si implementi i costruttori e i metodi
  - addUser(User): aggiunge utente
  - deleteUser(User): rimuove utente
- Si definisca un metodo main in *SocialBook* per testare le diverse funzionalità dell'applicazione.