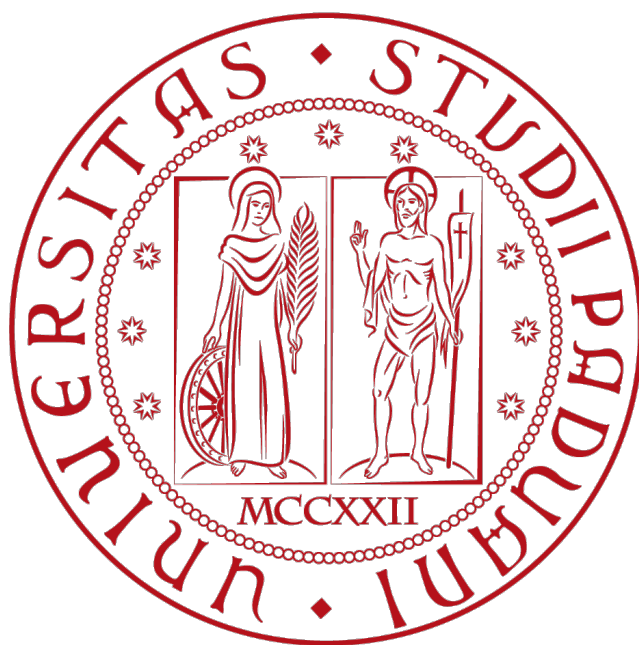


Programmazione Concorrente e Distribuita

Terza Parte

Suierica Bogdan Ionut 1008089

June 10, 2015



1 Cambiamenti

In questa sezione verranno descritti i cambiamenti apportati al codice per permettere al programma di essere distribuito tra un server e un client.

Sono stati individuati 3 package:

- **Server:** il package contiene le classi:
 - **PuzzleToSolve:** la classe definisce l'oggetto remoto implementando l'interfaccia *SolverAlgorithm* con il suo metodo *solve* (che si occupa di risolvere il puzzle) ed estendendo la classe *UnicastRemoteObject*. La classe contiene un costruttore con corpo vuoto che può sollevare un'eccezione di tipo *RemoteException*;
 - **PuzzleSolverServer:** è la classe che definisce il metodo *main*. La classe crea l'istanza dell'oggetto remoto *PuzzleToSolve* e gli associa un nome. Attraverso il metodo *rebind* lo registra nell'RMI registry.
- **Client:** il package contiene le classi:
 - **IOReader:** è la classe che si occupa della gestione dei file in input;
 - **IOWriter:** è la classe che si occupa della gestione dei file in output;
 - **PuzzleSolverClient:** è la classe che contiene il metodo *main*. Il programma client interroga il registro RMI utilizzando il metodo statico *lookup* della classe *Naming*, il quale restituisce un riferimento di tipo *Remote* all'oggetto cercato. Infine invoca il metodo remoto *solve* facendo un downcast al tipo *SolverAlgorithm*.
- **Shared:** il package contiene le classi in comune del server e del client. Le classi *Tile* e *Puzzle* estendono l'interfaccia *Serializable* in quanto gli oggetti vengono passati come parametri e come valori restituiti dal metodo *solve*:
 - **Tile:** rappresenta il singolo pezzo del puzzle;
 - **Puzzle:** rappresenta il puzzle;
 - **SolverAlgorithm:** rappresenta l'interfaccia remota. L'interfaccia estende l'interfaccia *Remote*. È presente il metodo *solve*, che ha il compito di risolvere il puzzle. Il metodo può sollevare un'eccezione di tipo *RemoteException*.