Programmazione Concorrente e Distribuita 2014-2015. Prima parte.

Andrea Giacomo Baldan 579117

August 25, 2015

Contents

1 Organizzazione e scelte implementative

Pur trattandosi della prima parte del progetto e quindi di un programma non particolarmente complesso, ho comunque deciso di implementare il codice con un minimo di organizzazione e modularità seguendo un design pattern che potesse facilitare l'estensione futura del software per le parti successive. Anche in assenza di un interfaccia grafica, ho scelto di seguire il design pattern MVC per la modularità e la separazione logica - output che offre, inoltre nel caso ipotetico in cui venisse implementata in futuro una GUI, il design pattern scelto agevolerebbe notevolmente il lavoro.
La directory src contenente i sorgenti, è dunque suddivisa nelle tre classiche sotto-directory models, views, controllers, dove rispettivamente:

- models contiene gli oggetti che rappresentano il puzzle e supporti I/O per l'interazione con i file di input e output
- views contiene le classi adibite alla rappresentazione output dei risultati
- controllers contiene le classi che si occupano della logica da applicare ai models

1.1 Organizzazione delle classi

Le classi sono raggruppate in un package, **puzzlesolver** con all'esterno la classe **PuzzleSolver** che contiene il `main` e le chiamate esecutive del programma.

1.1.1 Principi di OOP

1. Information hiding I campi dati delle classi sono stati dichiarati tutti `private`, accedibili mediante classici metodi `getters` e `setters`.

2 Algoritmo di ricostruzione

L'algoritmo di risoluzione implementato nella classe SortAlgSeq, derivata dalla classe base astratta SortAlg, è appunto un algoritmo di risoluzione sequenziale, e si può riassumere in 3 passi:

1. Localizzazione del primo pezzo del puzzle, che corrisponde al pezzo avente "VUOTO" a nord e ad ovest.
2. Inizio ciclo: ordina la riga a partire dal primo pezzo
3. Ricerca del pezzo a sud del primo pezzo della nuova riga ora ordinata, se il pezzo a sud non esiste e troviamo dunque "VUOTO", il ciclo si ferma in quanto tutte le righe sono quindi già state ordinate, altrimenti si ripete (1) utilizzando il "nuovo" primo pezzo.

3 Note

Il progetto è stato sviluppato in ambiente linux, utilizzando la JVM versione 1.7.0.