Mattia Sorgato, matricola 1004404

# Relazione di Progetto di Programmazione Concorrente e Distribuita

## Anno didattico 2014/2015

### Principi di Programmazione ad Oggetti

All'interno del progetto didattico da me svolto, ho prestato molta attenzione a far sì che ogni campo utilizzato all'interno della classe rimanesse non accessibile (soprattutto non modificabile) dall'esterno della classe stessa. Il principio dell'incapsulamento è visibile maggiormente nella classe "Piece", che rappresenta un pezzo di un Puzzle.

In questa classe, i campi riguardanti gli identificativi proprio e dei pezzi adiacenti, sono nascosti all'esterno e sono marchiati **final**, cosicché non possano venir modificati da azioni esterne alla classe. Sono comunque presenti dei metodi di confronto tra due **Piece** che permettono di effettuare controlli sugli **id**, nascondendo però l'architettura sottostante.

Il Parser che effettua il controllo dell'input e la classe che rappresenta l'algoritmo di ordinamento sono entrambe composte da un'interfaccia che espone i metodi base di ognuna e una classe che implementa l'interfaccia. Ciò è motivato da cause di modifiche future o di riadattamenti successivi, in quanto il progetto sarà modificato nelle prossime consegne, e avendo un'interfaccia comune, bisognerà solamente ridefinire gli **override** dei metodi usati per chiamate polimorfe.

Questa scelta è stata presa anche in un'ottica di gestione della concorrenza. Infatti, con le prossime specifiche e concorrenza, l'algoritmo di ordinamento è già stato pensato per ordinare il puzzle in maniera parallela, su più linee, così da rendere il programma più concorrente possibile.

La suddivisione in diversi *package,* invece, è dovuta ad una maggior divisione dei vari aspetti del progetto, così da aumentare l'indipendenza delle varie funzionalità del progetto.

### Organizzazione

#### Package "puzzlesolver"

La dicitura *"puzzlesolver"* identifica il package generale del progetto.

#### Package "puzzlesolver.parser"

Questo package contiene l'inerfaccia **IParser** e la sua implementazione **PuzzleParser**.

così come la classe SequentSort (che implementa l'algoritmo di risoluzione del Puzzle sequenzialmente) e la sua interfaccia si trovano nel package **sort**. Così anche i test sono nel package **tests**. L'unica classe che è posta nel package **puzzlesolver**, il package generale del progetto, è la classe contenente il main, ovvero **PuzzleSolver**.

### Algoritmo di ordinamento

L'idea alla base dell'algoritmo sequenziale è molto semplice. Per la sua ideazione e codifica, mi sono ispirato al metodo comune di risoluzione di un puzzle, ovvero partendo dai bordi e successivamente riempiendo l'interno. La strategia utilizzata per l'ordinamento si divide in diversi passi:

1. Trovare il pezzo che identifica il bordo in alto a sinistra;
2. Grazie a questo pezzo, completare la prima linea del puzzle cercando ogni volta il pezzo che ha come proprio ID quello che sta ad EST di quello precedente, fino ad arrivare al bordo EST;
3. Si passa all'ordinamento della riga successiva cercando il primo elemento, il quale avrà l'ID corrispondente al SUD del primo elemento della riga precedente;
4. Ripetere il punto 3 fino ad arrivare all'ultima riga del Puzzle.

### Note

motivi del java 7

opzioni di lancio dei comandi

utilizzo del bash