

# Progetto Programmazione Concorrente e Distribuita Prima Parte

Suierica Bogdan Ionut 1008089

December 21, 2014

## 1 Introduzione

### 1.1 Scopo del documento

Lo scopo del documento è quello di presentare le principali scelte architettoniche del progetto.

## 2 Descrizione Progetto

Il progetto realizzato permette di ordinare un puzzle di caratteri. Il file in input contiene il testo rappresentato dal puzzle disordinato. Il file in output contiene il testo rappresentato dal puzzle disordinato organizzato in una riga, il puzzle ordinato in forma tabellare, e la dimensione della tabella.

## 3 Componenti e Classi

### 3.1 Package puzzle

Per questa prima parte del progetto, puzzle rappresenta il package globale del progetto. Si potevano utilizzare due package differenti, uno per le operazioni di input-output e uno per le funzionalità di ordinamento del puzzle, ma ho preferito tenere tutto insieme.

### 3.2 Interface SolverAlgorithm

Interfaccia che espone la funzionalità di ordinare un Puzzle. Il suo unico metodo **solve** viene realizzato nella classe **SolvedPuzzle**.

### 3.3 Classe **IOReader**

Questa classe rappresenta la lettura da file del puzzle disordinato in forma tabellare. Quando avviene la lettura , calcola la dimensione della tabella.

### 3.4 Classe **IOWriter**

Questa classe rappresenta la scrittura su file.

### 3.5 Classe **Tile**

Questa classe rappresenta il singolo pezzo del puzzle. Ogni pezzo ha un id univoco e gli id dei pezzi che li stanno vicino nelle quattro direzioni.

### 3.6 Classe **Puzzle**

Questa classe rappresenta l'insieme di pezzi del puzzle. E' caratterizzata da un ArrayList di Tile , due interi che indicano la dimensione della puzzle, una stringa che indica il path a cui si trova il file in input e un riferimento all'oggetto **IOReader** .

Inoltre alla costruzione dell'oggetto Puzzle avviene la lettura da file e l'ArrayList di Tile viene popolato.

### 3.7 Classe **SolvedPuzzle**

La classe **SolvedPuzzle** implementa l'interfaccia **SolverAlgorithm**. Ha il compito di salvarsi il puzzle disordinato e restituirlo ordinato attraverso il metodo realizzato solve, fornito dal interfaccia **SolverAlgorithm**.