

# 03FYZ TECNICHE DI PROGRAMMAZIONE

## Esercitazione di Laboratorio 05 – es. 01 – 13 Aprile 2016

Obiettivi dell'esercitazione:

- Apprendere il meccanismo della ricorsione

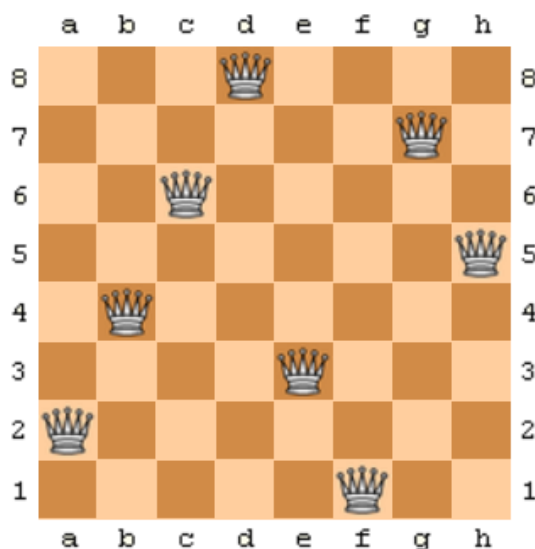
### ESERCIZIO 1

Scopo dell'esercitazione: Realizzare in linguaggio Java un'applicazione priva di interfaccia grafica per risolvere il "Rompicapo delle 8 regine" utilizzando un algoritmo ricorsivo.

Descrizione del problema: Il "Rompicapo delle 8 regine" consiste nel trovare il modo di posizionare otto regine su una scacchiera 8x8 tali che nessuna di esse possa catturarne un'altra. Una soluzione prevede che nessuna regina abbia una colonna, riga o diagonale in comune con un'altra regina. In *Fig. 1* un esempio di soluzione.

Riferimento: [https://it.wikipedia.org/wiki/Rompicapo\\_delle\\_otto\\_regine](https://it.wikipedia.org/wiki/Rompicapo_delle_otto_regine)

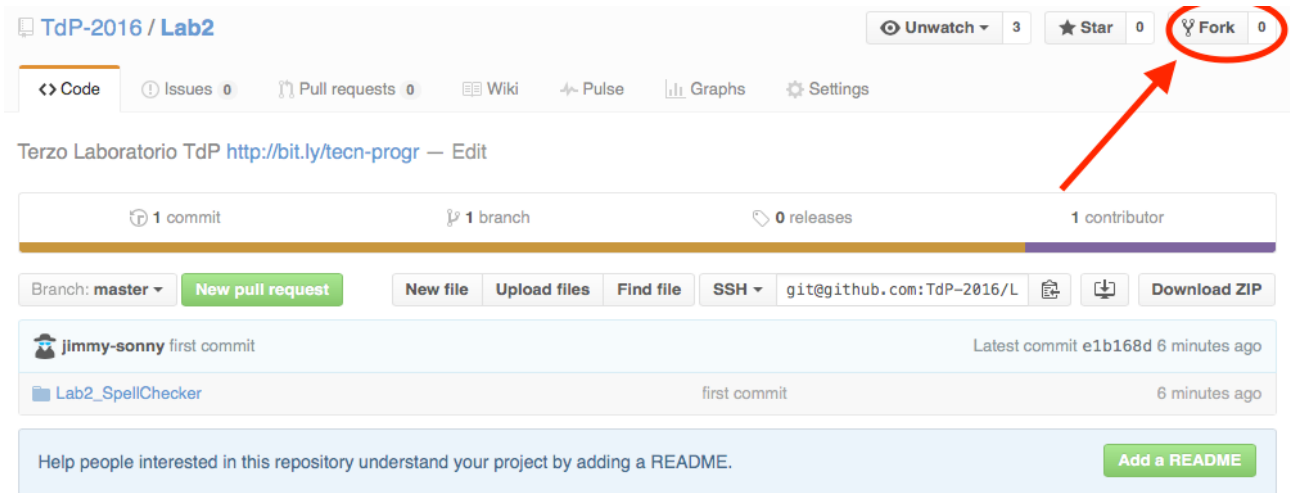
*Fig. 1*



# 03FYZ TECNICHE DI PROGRAMMAZIONE

## Istruzioni per effettuare il fork di un repository GitHub

- Effettuare il login su GitHub utilizzando il proprio username e password.
- Aprire il repository su GitHub relativo al sesto laboratorio:  
<https://github.com/TdP-2016/Lab5>
- Utilizzare il pulsante *Fork* in alto a destra per creare una propria copia del progetto.



L'azione di Fork crea un nuovo repository nel proprio account GitHub con una copia dei file necessari per l'esecuzione del laboratorio.

- Aprire Eclipse, andare su *File -> Import*. Digitare *Git* e selezionare *Projects from Git -> Next -> Clone URI -> Next*.
- Utilizzare la URL del **proprio** repository che si vuole clonare (**non** quello in TdP-2016!), ad esempio:  
<https://github.com/my-github-username/Lab5>
- Fare click su *Next*. Selezionare il branch (*master* è quello di default) fare click su *Next*.
- Selezionare la cartella di destinazione (quella proposta va bene), fare click su *Next*.
- Selezionare *Import existing Eclipse projects*, fare click su *Next* e successivamente su *Finish*.
- Il nuovo progetto Eclipse è stato clonato ed è possibile iniziare a lavorare.
- A fine lavoro ricordarsi di effettuare Git commit e push, utilizzando il menù *Team in Eclipse*.

**ATTENZIONE:** solo se si effettua Git **commit** e successivamente Git **push** le modifiche locali saranno propagate sui server GitHub e saranno quindi accessibili da altri PC e dagli utenti che ne hanno visibilità.

## 03FYZ TECNICHE DI PROGRAMMAZIONE

### Esercitazione di Laboratorio 05 – es. 02 – 13 Aprile 2016

Obiettivi dell'esercitazione:

- Utilizzo del pattern MVC e DAO
- Utilizzo di JDBC
- Apprendere il meccanismo della ricorsione

### ESERCIZIO 2

Scopo dell'esercitazione: Realizzare in linguaggio Java un'applicazione dotata di interfaccia grafica per risolvere con un algoritmo ricorsivo il famoso gioco "Ruzzle".

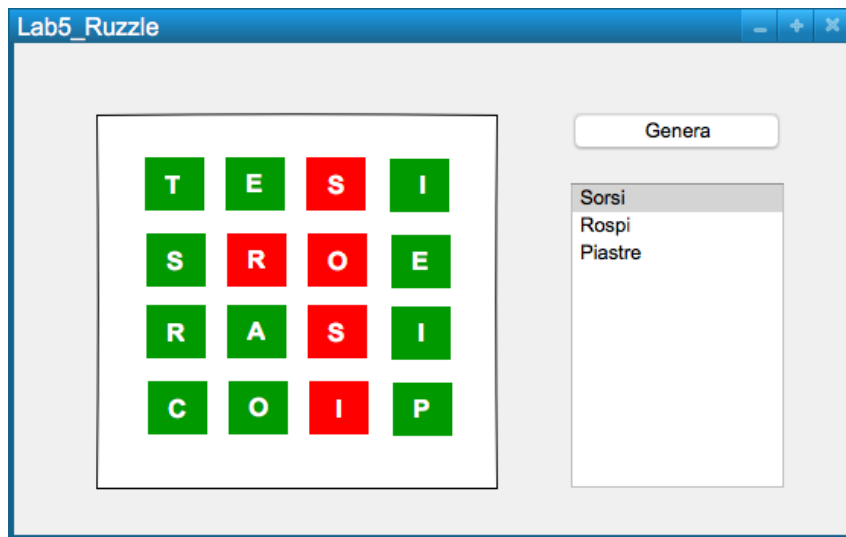
Descrizione del gioco: Il giocatore deve formare il maggior numero di parole di senso compiuto con le sedici lettere a disposizione nella griglia 4×4 sullo schermo (vedi Fig. 2). Le parole devono essere di almeno 2 lettere e devono essere formate unendo lettere adiacenti fra loro in orizzontale, verticale o diagonale. Non è possibile inserire la stessa casella-lettera più volte all'interno della stessa parola.

Funzionamento previsto dell'applicazione: L'utente utilizza il pulsante *Genera* per generare una griglia 4x4 di 16 lettere in modo random. Utilizzando un algoritmo ricorsivo, il programma inserisce tutte le parole italiane trovate in una *ListView* di JavaFx. Se l'utente fa click su una parola della *ListView*, le lettere che formano la parola si colorano di rosso.

Fig. 2



Fig. 3



- Esercizio 2.1** Realizzare un'interfaccia grafica con *JavaFx* simile a quella mostrata in Fig. 3.
- Esercizio 2.2** Importare il database *dizionario.sql*.
- Per importare il database, avviare il database locale *XAMPP*; successivamente lanciare il programma *HeidiSQL* e selezionare l'opzione *File -> Carica file SQL*. Selezionare il file *dizionario.sql* (scaricato dal sito del corso, sezione Materiale, Data-Sets), eseguire la query direttamente (senza caricare nell'editor i dati) per importarlo e cliccare sul tasto *Aggiorna*, per permettere la corretta visualizzazione all'interno dell'elenco del database appena caricato.
- Fare uso dei pattern MVC e DAO, come spiegato a lezione.
- Esercizio 2.3** Implementare la generazione automatica di una griglia 4x4 di 16 lettere in modo random. Tutte le lettere A-Z hanno la stessa probabilità di essere selezionate.
- Esercizio 2.4** Implementare l'algoritmo ricorsivo per cercare le parole italiane. (Suggerimento: per ogni sequenza di lettere generata fare una Query SQL per vedere se esiste nel dizionario italiano. Se non esiste, conviene interrompere la ricerca per quella sequenza.)
- Esercizio 2.5** Inserire le parole italiane trovate in una *ListView* di JavaFx in ordine crescente per lunghezza.
- Esercizio 2.6** Selezionando una parola dalla *ListView*, le lettere che la compongono vengono colorate di rosso.