

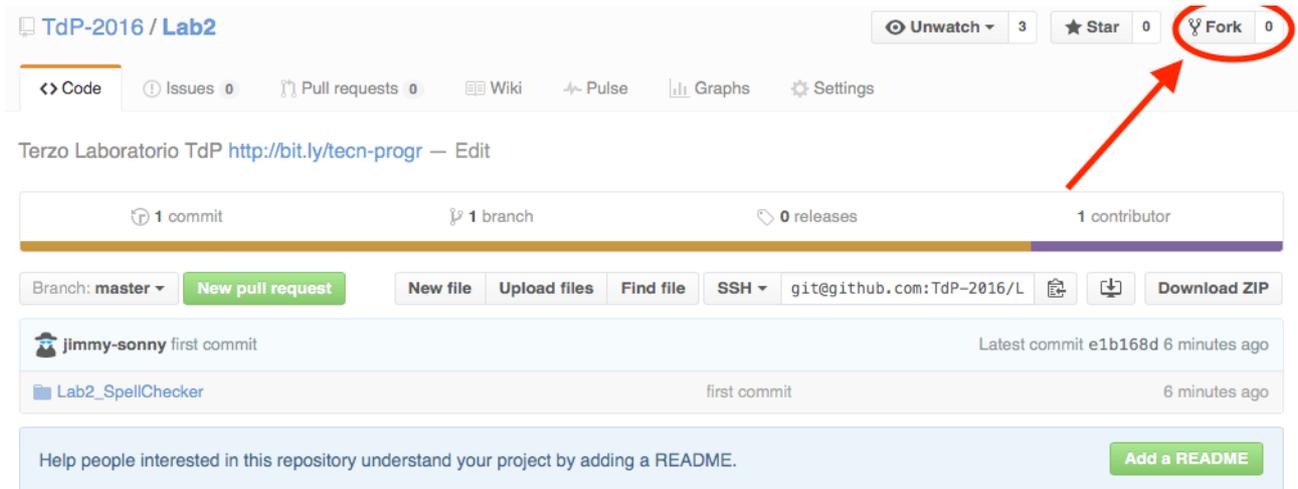
03FYZ TECNICHE DI PROGRAMMAZIONE

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 nono laboratorio:

<https://github.com/TdP-2017/Lab09>

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-2017!), ad esempio:

<https://github.com/my-github-username/Lab09>

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 08 – 3 maggio 2017

Obiettivi dell'esercitazione:

- Utilizzo della libreria JGraphT
- Grafi pesati e cammini minimi

ESERCIZIO 1

Dopo aver effettuato il fork del progetto relativo al nono laboratorio (Lab09), sviluppare un'applicazione JavaFX che permetta di calcolare il tempo di percorrenza tra due stazioni della rete metropolitana di Parigi.

Realizzare un'interfaccia grafica con *JavaFx* simile a quella mostrata in *Fig. 1*. Partendo dalle informazioni riportate nel database *metroparis.sql*, costruire un grafo non orientato e pesato per rappresentare il sistema metropolitano di Parigi. I vertici del grafo sono rappresentati dalle stazioni, mentre gli archi collegano due stazioni unite da una tratta (utilizzare la tabella *Connessione*). Il peso dell'arco è dato dal tempo di percorrenza tra due stazioni. Infine popolare i menù a tendina dell'interfaccia grafica con tutte le fermate disponibili.

L'utente seleziona dai menù a tendina una stazione di partenza ed una di arrivo. Facendo click sul pulsante "*Calcola percorso*", il programma deve calcolare il cammino minimo, ignorando i possibili cambi di linea, tra la stazione di partenza e di arrivo selezionate. Visualizzare tutte le fermate intermedie ed il tempo di percorrenza totale (assumendo una sosta di 30 secondi per ogni fermata).

Strutturare il programma secondo i pattern MVC e DAO come spiegato a lezione.

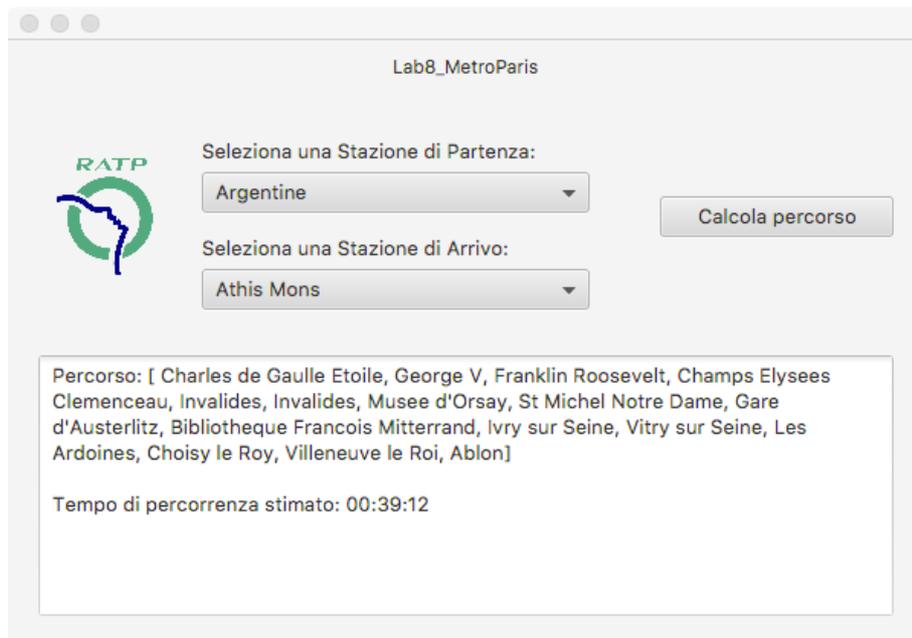
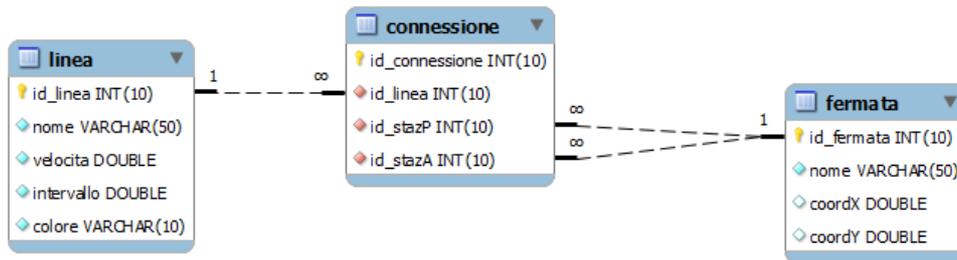


Fig. 1

Continua nella pagina seguente

Descrizione del database "metroparis":



La tabella *fermata* riporta la lista di tutte le fermate della metropolitana di Parigi. La tabella *linea* contiene la lista delle linee della metro. La tabella *connessione* contiene l'informazione sul collegamento tra due fermate, su una specifica linea.

Nota: per calcolare la distanza tra due fermate, è consigliato utilizzare la libreria **simplelatlng-1.3.0.jar**, già inclusa nel progetto di base. In particolare, utilizzare il metodo *distance*, come mostrato nell'esempio:

```
distance(new LatLng(x1, y1), new LatLng(x2, y2), LengthUnit.KILOMETER)
```

ESERCIZIO 2

Migliorare l'applicazione realizzata nel primo esercizio prendendo in considerazione nel calcolo del cammino minimo tra due fermate anche il tempo di attesa nel caso in cui fosse necessario cambiare linea. Per semplicità, nel cambio linea considerare il caso peggiore in cui il tempo di attesa è sempre pari all'*intervallo* indicato nella tabella *Linea*. Facendo click sul pulsante "Calcola percorso" visualizzare la lista delle fermate intermedie, il cambio di linea ed il tempo di percorrenza totale. Esempio in Fig 2.

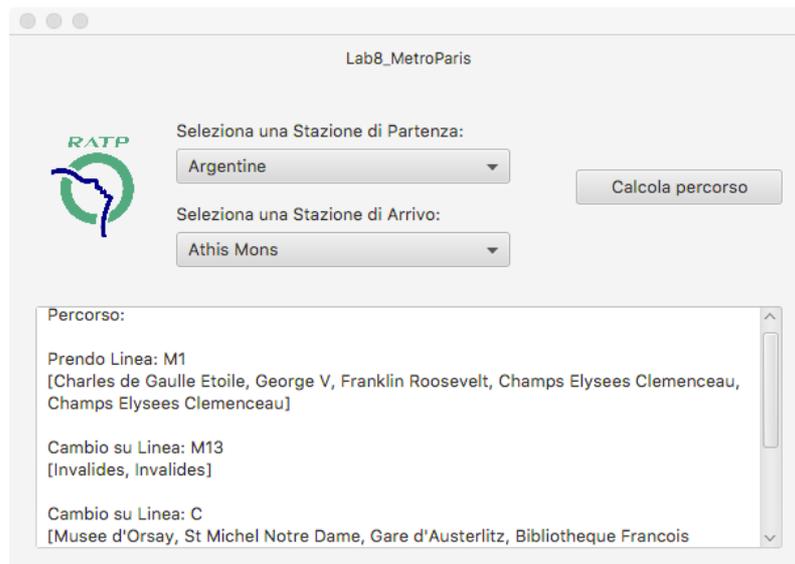


Fig. 2

Continua nella pagina seguente

Nota: per risolvere questa variante dell'esercizio, si consiglia di cambiare rappresentazione del grafo passando ad un grafo orientato e pesato, esempio *Fig. 3*. Nel caso in cui una stazione sia un punto di collegamento tra più linee, bisogna inserire nel grafo più vertici, uno relativo ad ogni linea della stazione. Il peso degli archi orientati che collegano la stessa stazione su diverse linee è pari al tempo di attesa del cambio linea (campo *intervallo*). I pesi degli archi orientati tra stazioni sulla stessa linea sono pari ai tempi di percorrenza della tratta, come nell'esercizio precedente.

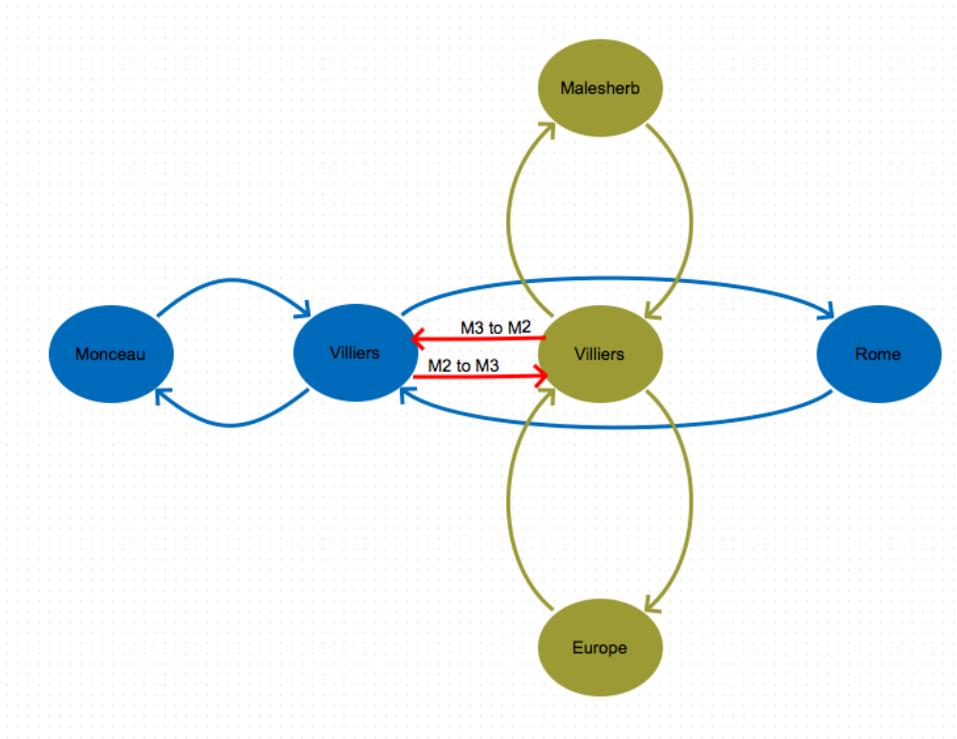


Fig. 3