

03FYZ TECNICHE DI PROGRAMMAZIONE

Istruzioni per la preparazione dell'ambiente di sviluppo di base

- Scaricare ed installare la piattaforma di sviluppo *Java Platform (JDK) 8* dal sito <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Selezionare JDK e poi la versione del *Java SE Development Kit 8u72* corrispondente al vostro sistema operativo. NON è necessario scaricare ed installare JavaFX, in quanto il pacchetto è già incluso nell'installazione di Java versione 8.
- Scaricare l'ambiente di sviluppo *Eclipse MARS*, dal sito <http://www.eclipse.org/downloads/> scegliendo la versione *Eclipse IDE for Java EE Developers*. Estrarre la cartella dall'archivio scaricato e lanciare l'ambiente di sviluppo.
- Dalla barra del menù, selezionare *Help -> Eclipse Marketplace*. Cercare “javafx” ed installare il pacchetto *e(fx)clipse 2.1.0* (dovrebbe essere presente tra i risultati della ricerca), che include una serie di plugin utili per sviluppare con JavaFX.
- Scaricare ed installare il tool *JavaFX Scene Builder 8.1*, raggiungibile tramite il sito <http://gluonhq.com/open-source/scene-builder/>

03FYZ TECNICHE DI PROGRAMMAZIONE

Esercitazione di Laboratorio 00 - 2 Marzo 2016

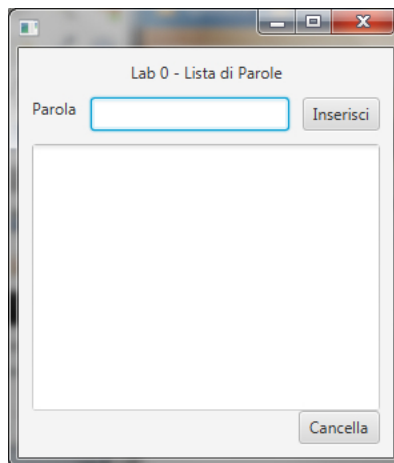
Obiettivi dell'esercitazione:

- Riprendere i concetti di sviluppo di applicazioni Java, già affrontati nel corso di Programmazione a Oggetti
 - Scaricare ed importare progetti Eclipse
 - Integrare le proprie classi in progetti esistenti
-

Esercizio 1

Funzionamento previsto al termine dell'esercizio uno: l'utente inserisce nell'apposito campo una parola e, cliccando sul *Button* "Inserisci", la parola appena inserita sarà aggiunta ad una struttura dati interna e visualizzata nel campo sottostante insieme alle altre parole precedentemente inserite, **ordinate alfabeticamente**.

Cliccando sul *Button* "Cancella", dovrà essere possibile cancellare tutte le parole precedentemente inserite e attualmente presenti nella struttura, ricominciando quindi da capo.



Esercizio 1.1 – Creare una classe *Parole* (dichiarandola in un package di nome *it.polito.tdp.parole.model*), in grado di gestire la struttura dati citata, che sia dotata dei seguenti metodi:

```
public void addParola(String p) ; // aggiunge nuova parola
public List<String> getElenco() ; // restituisce elenco parole ordinate
public void reset() ; // azzera elenco parole
```

Esercizio 1.2 – Creare una classe *Parole_Test* contenente un metodo *Main* eseguibile ed in grado di richiamare automaticamente i metodi della classe *Parole* per effettuare una serie test inserendo alcune parole e mostrando la lista ordinata, e testando quindi tutti i metodi implementati.

Esercizio 1.3 – Importare il progetto previsto per questa esercitazione scaricando l'archivio zip dal seguente link: <https://github.com/TdP-2016/Lab0>. Provare ad eseguire il progetto e verificare che l'interfaccia grafica parte, ma non fa nulla.

Esercizio 1.4 – Inserire nel progetto la classe *Parole* realizzata nel punto 1.1, sostituendola alla classe “stub” attualmente presente, sempre all’interno del package *it.polito.tdp.parole.model*.

Esercizio 1.5 – Verificare il funzionamento della interfaccia grafica, eseguendo il progetto completo.

Esercizio 2

Funzionamento previsto al termine dell’esercizio due: l’utente inserisce nell’apposito campo un numero e, cliccando sul *Button* “Test” viene generata una lista di stringhe su cui vengono eseguiti alcuni test. Le parole generate saranno inserite in una struttura dati interna. Nel campo sottostante vengono stampati i tempi di esecuzione di alcune operazioni di cancellazione di elementi della lista. Lo scopo dell’esercizio è analizzare le differenze di prestazioni tra *ArrayList* e *LinkedList*.

Esercizio 2.1 – Modificare l’interfaccia grafica del precedente esercizio. Il pulsante “Inserisci”, diventa “Test”, “Parola” diventa “Numero”.

Esercizio 2.2 – Implementare nel package *it.polito.tdp.parole.model* una classe *WordSet* dotata dei seguenti metodi:

```
// genera parole casuali e le inserisce in una lista  
public void generaParole(int numParole) ;
```

```
// elimina l’elemento in posizione i e restituisce il tempo di esecuzione  
public long cancella(int i) ;
```

Per la generazione di parole casuali fare riferimento all’algoritmo visto a lezione. Utilizzare la funzione *System.nanoTime()* per calcolare il tempo di esecuzione.

Esercizio 2.3 – Modificare il comportamento del pulsante “Test” nel package *controller* per effettuare alcuni test sulle prestazioni delle strutture dati. Utilizzare un oggetto di tipo *WordSet* per generare un set di parole, in seguito provare a cancellare l’elemento iniziale, centrale ed ultimo. Stampare in *txtResult* i tempi di esecuzione di ciascuna operazione. Ripetere l’esperimento cambiando la struttura dati da *LinkedList* ad *ArrayList*.