

## 03FYZ TECNICHE DI PROGRAMMAZIONE

### Istruzioni per la preparazione dell'ambiente di sviluppo di base

---

- Scaricare ed installare la piattaforma di sviluppo *Java Platform (JDK) 8.1* dal sito <http://www.oracle.com/technetwork/java/javase/downloads/index.html> . NON è necessario scaricare ed installare JavaFX, in quanto il pacchetto è già incluso nell'installazione di Java versione 8.1.
- 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* (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 2.0*, raggiungibile tramite il sito <http://www.oracle.com/technetwork/java/javafxscenebuilder-1x-archive-2199384.html>

## 03FYZ TECNICHE DI PROGRAMMAZIONE

### Esercitazione di Laboratorio 00 - 2 Marzo 2016

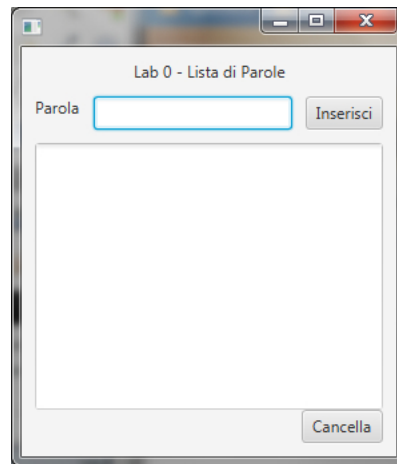
---

#### 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
- 

Funzionamento previsto al termine dell'esercizio: 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 1.6** – Modificare l’interfaccia grafica. Il pulsante “Inserisci”, diventa “Esegui Test”, “Parole” diventa “Numero”.
- Esercizio 1.7** – Implementare nel *Model* una classe *WordSet* per la generazione di parole casuali, da inserire in una lista, come visto a lezione.
- Esercizio 1.8** – Utilizzando il pulsante “Run”, effettuare dei test stampando il risultato nella casella di testo dell’interfaccia grafica: cancellare l’elemento iniziale, centrale ed ultimo. Utilizzare la funzione *System.nanoTime()* per confrontare le prestazioni tra la *LinkedList* e *ArrayList*.