

Programmazione

LAB 01

<http://www.iet.unipi.it/m.cimino/prg/>

Mario G.C.A. Cimino
Dipartimento di Ingegneria dell'Informazione

1. Presentazione. Ricevimento subito dopo i laboratori, oppure giovedì alle 14.30, su prenotazione via email;
2. Seguire i laboratori è fortemente consigliato per acquisire un buon metodo di lavoro; chi intende dare l'esame ai primi appelli può costruire progressivamente il progetto, chi intende darlo successivamente può svolgere gli esercizi seguendo step-by-step lo svolgimento dei tutorial da parte del docente; lo studente lavoratore può stare al passo con il corso mantenendosi in contatto con un collega che segue.
3. Nei laboratori si farà molta pratica di programmazione Java sperimentando le principali librerie del linguaggio per costruire incrementalmente delle applicazioni con interfaccia grafica, in grado di archiviare dati su file system o su database management system, di distribuire in rete parte della elaborazione o della archiviazione, di interfacciarsi con altre applicazioni tramite XML, di elaborare ingressi ed uscite multimediali. Si apprenderà come consultare la documentazione, come rilevare errori e collaudare il software.

4. Sito web del corso:

<http://elearn.ing.unipi.it/course/view.php?id=430>

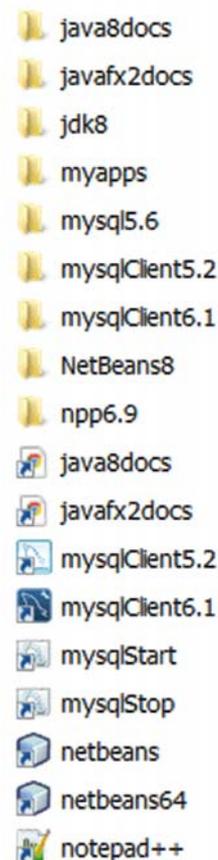
5. Download del pacchetto all-in-one, per sistemi Windows, contenente: Java 8; MySQL; Java API (documentazione di tutte le librerie); Notepad++; NetBeans (editor avanzato per laboratori e progetto).

6. Salvare sul desktop il pacchetto all-in-one e decomprimerlo in C. Apparirà una cartella C:\prg\ con il contenuto visualizzato in figura →

7. Nella cartella *myapps* si metteranno tutti i programmi sviluppati. Per compilarli ed eseguirli da riga di comando si dovrà invocare esplicitamente la versione di java contenuta nella cartella *jdk8*. Es. collocando un file *Test.java* in C:\myapps, questo si compila ed esegue così:

```
C:\prg\jdk8\bin\javac *.java
```

```
C:\prg\jdk8\bin\java Test
```



Ambiente di sviluppo e applicazione Hello World

8. Aprendo l'editor *NetBeans* si potrà invece gestire anche compilazione ed esecuzione all'interno dell'ambiente integrato.

9. Inizieremo costruendo una semplice applicazione con interfaccia grafica, mostrando la potenza e la espressività delle API Java FX.

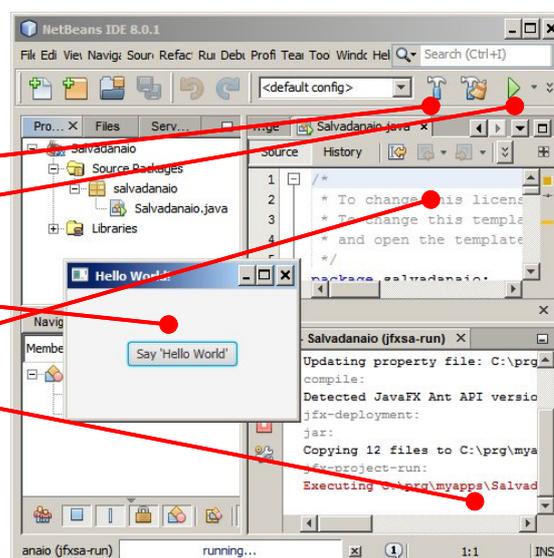
10. Dall'ambiente *NetBeans*: File → New Project; Categories: JavaFX; Projects: JavaFX Application; Next.11. Project Name: Salvadanaio;
Project Location: C:\prg\myapps;
Create Application Class: Salvadanaio;
Finish;

11. Compilare, e poi Eseguire

12. Si apre una semplice finestra

13. Finestra messaggi

14. Finestra del codice

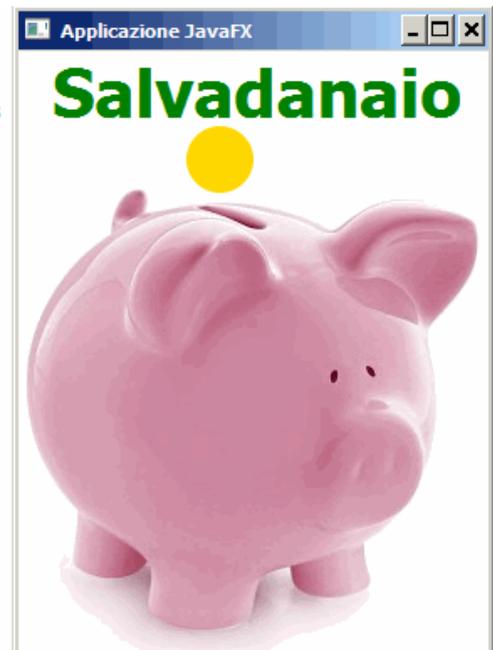
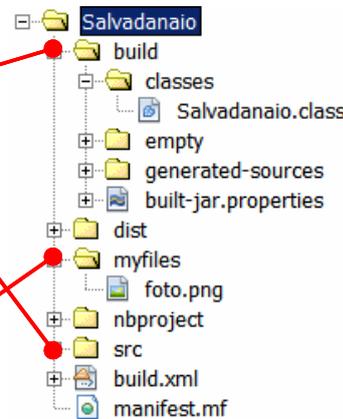
15. Chiudere l'applicazione demo e vuotare il file *Salvadanaio.java* per scrivere nuovo codice.

17. **Specifiche:** costruire la finestra raffigurata in basso, consultando le API JAVA FX (vedere prossima slide). **Alcune classi suggerite:** Application, Label, Group, Scene, Circle (tinyurl.com/javafx-controls).

18. NetBeans colloca i file class in *build* e i file java in *src*.

18. Scaricare foto.png da <http://www.iet.unipi.it/m.cimino/prg/foto.png> e per esso creare la cartella *myfiles*.

19. Si può compilare ed eseguire anche da *classes* digitando:



tinyurl.com/javafx-intro

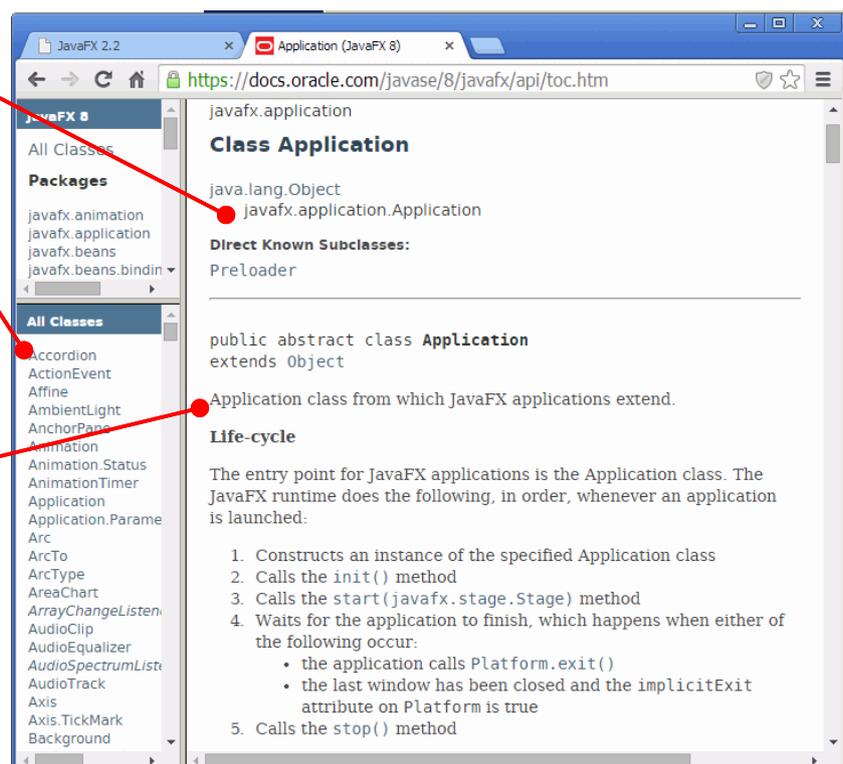
```
C:\prg\jdk8\bin\javac *.java
C:\prg\jdk8\bin\java Salvadanaio
```

20. Consultare spesso le API, per abituarsi alla gerarchia di classi

Package di appartenenza della classe

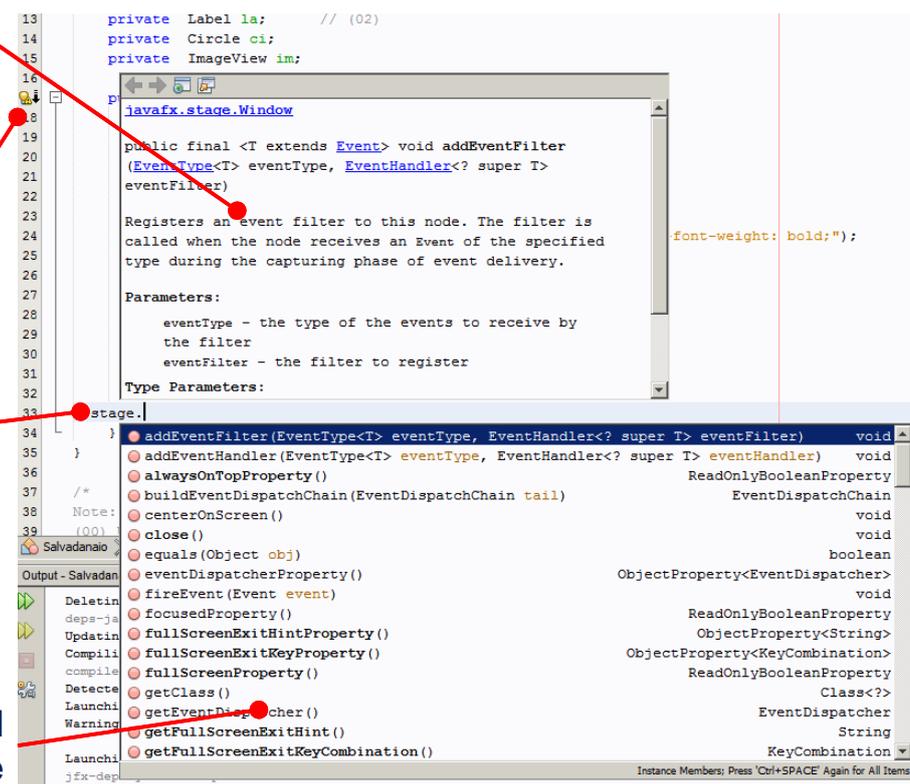
Tutte le classi (in ordine alfabetico)

Descrizione, esempi, elenco di attributi e metodi



21. L'autocompletamento si basa sulle API, segnala quelle deprecated.

Descrizione della classe
 warning ed errori appaiono spostando il mouse sulle icone accanto al numero di riga
 Digitare il punto per far apparire l'autocompletamento
 Metodi accessibili dal punto in cui si scrive



1. All'esame si adopera il pacchetto 'all-in-one' senza accesso Web. È bene consultare da subito la documentazione 'Java doc' che trovate in esso, e i suggerimenti dell'ambiente NetBeans.
2. Si parte sempre dalle **specifiche**, fornite in forma visuale o testuale, con **alcune classi suggerite**.
3. Dai nomi delle classi suggerite, nell'ordine, si consulta la Java doc, e si cercano in essa esempi base, con i quali scrivere da sé il codice.
4. Notare i suggerimenti dell'ambiente di sviluppo e cercare di risolverli prima di proseguire. Compilare ed eseguire il codice man mano, per vederne l'evoluzione, facendo pratica con l'ambiente di sviluppo.
5. Non inserire tutti gli import trovati negli esempi, ma lasciare che l'ambiente segnali qualcosa e poi trovare il giusto package attraverso la Java doc, confrontandosi solo dopo con la soluzione.
6. Se l'applicazione prevede più componenti dello stesso package (es. più componenti dell'interfaccia, più flussi), vengono suggerite solo alcune classi. Provare inizialmente a cercare da sé le altre classi sfogliando il package nella documentazione.

Salvadanaio\src\Salvadanaio.java

```
1 import javafx.application.*;
2 import javafx.stage.*;
3 import javafx.scene.*;
4 import javafx.scene.shape.*;
5 import javafx.scene.control.*;
6 import javafx.scene.paint.*;
7 import javafx.scene.image.*;
8
9 public class Salvadanaio extends Application { //01
10
11     private Label la; //02
12     private Circle ci;
13     private ImageView im;
14
15     public void start(Stage stage) { //03
16
17         la = new Label("Salvadanaio"); //04
18         ci = new Circle(120,65,20, Color.GOLD);
19         im = new ImageView("file:../../myfiles/foto.png"); //05
20         //im = new ImageView("http://www.iet.unipi.it/m.cimino/prg/foto.png");//06
21         //07
22         Group root = new Group(la, ci, im); //08
23         Scene scene = new Scene(root, 280,500); //09
24
25         stage.setTitle("Applicazione Java FX"); //10
26         stage.setScene(scene); //11
27         stage.show(); //12
28     }
29 }
30
31 /*
32 Note:
```

```
33 (01) Una applicazione Java con interfaccia grafica si realizza estendendo la
34     classe Application. Il metodo start parte automaticamente all'avvio e va
35     ridefinito (non e' obbligatorio un main)
36     https://docs.oracle.com/javafx/2/api/javafx/application/Application.html
37 (02) Struttura dell'interfaccia grafica o view
38     Principali componenti grafici di Java FX:
39     https://docs.oracle.com/javafx/2/ui\_controls/jfxpub-ui\_controls.htm
40 (03) Il metodo start viene chiamato automaticamente all'avvio, consente
41     di inizializzare l'applicazione
42 (04) Istanziamento degli oggetti grafici
43 (05) Il file foto.png ha un percorso relativo alla cartella classes (in Windows,
44     mentre su MACOS/Unix e' relativo alla cartella Salvadanaio).
45     Compilando da riga di comando il percorso e' relativo sempre al punto in
46     cui si generano i class (dentico per i vari sistemi operativi). Es.
47     compilando da src il percorso e' "file:../myfiles/foto.png".
48 (06) In alternativa si puo' usare un percorso web
49 (07) aggiunta degli elementi nella interfaccia:
50 (08) Un gruppo e' un contenitore di oggetti osservabili dentro la scena
51 (09) Una scena e' il contenitore principale della finestra, il nome discende
52     dal mondo delle classi per lo sviluppo di video game, librerie 3D
53     https://docs.oracle.com/javase/8/javafx/scene-graph-tutorial/scenegraph.htm
54 (10) Lo stage (palcoscenico) e' la finestra, il luogo dove si svolge la scena
55 (11) Inserisce la scena nello stage
56 (12) Mostra la finestra
57 */
```