Esercitazione 4

Programmazione 2

Prendete carta e penna...

Esercizi

Per ciascuno dei seguenti sorgenti Java dire se:

- genera un errore in fase di compilazione (quale errore e perché)
- genera un errore durante in fase di esecuzione (quale errore e perché)
- viene compilato ed eseguito correttamente (qual'è l'output)

Esercizio 1

```
public class Test1 extends Test1B {

    void f() {
        System.out.println("A");
    }

    public static void main(String a[]) {
        Test1B o = new Test1();
        o.f();
    }
}

class Test1B {

    void f() {
        System.out.println("B");
    }
}
```

Esercizio 2

```
public class Test2 extends Test2B {

    void f() {
        System.out.println("A");
    }

    public static void main(String a[]) {
        Test2B o = new Test2();
        o.f();
    }
}

class Test2B {

    final void f() {
        System.out.println("B");
    }
}
```

Esercizio 3

```
import java.util.*;

public class Test3 {

    Collection z;
    public Test3(int k){
        if (k==0) z = new HashSet();
        else z = new LinkedList();
    }
    public static void main(String arg[]) {
        Test3 a = new Test3(1);
        Test3 b = new Test3(0);
        for (int k=0;k<15;k++){
            Integer z = new Integer(k%5);
            a.z.add(z);
            b.z.add(z);
        }
        System.out.println(a.z.size()+b.z.size());
    }
}
```

Esercizio 4

```
import java.util.*;

public class Test4 {

   Collection z;
   public Test3(int k){
      if (k==0) z = new HashSet();
      else z = new LinkedList();
   }
   public static void main(String arg[]) {
      Test4 a = new Test4(1);
      Test4 b = new Test4(0);
      for (int k=0;k<15;k++){
         Integer z = new Integer(k%5);
         a.z.add(z);
         b.z.add(z);
      }
      System.out.println(a.z.size()+" "+b.z.size());
   }
}
```

Esercizio 5

```
class Test5b{
   Test5b(){
      System.out.print(4);
   }
   void f(){
      System.out.print(45);
   }
}
public class Test5 extends Test5b{
   Test5(){
      System.out.print(7);
   }
   void f(){
      System.out.print(47);
   }
   public static void main(String i[]){
      Test5b uno = new Test5b();
      Test5 due = (Test5)uno;
      due.f();
   }
}
```

Esercizio 6

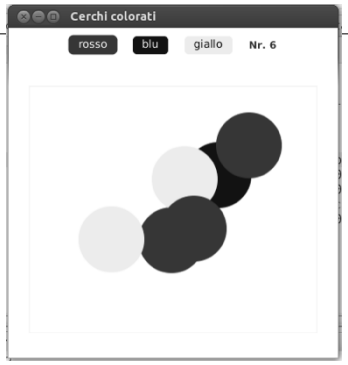
```
package prog2;
public class Test6{
   int c=4;
   public void test(){
      System.out.print(--c);
  }
   public static void main(String a[]){
      (new it.unitn.Test6()).test();
   }
}
```

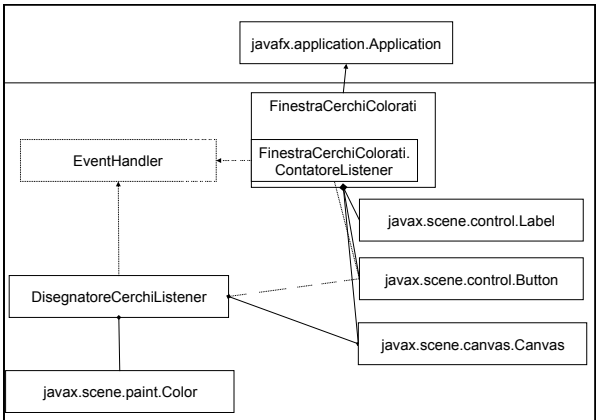
```
package it.unitn;
public class Test6  extends prog2.Test6{
   public void test(){
      System.out.print(c++);
   }
}
```

Gestione degli eventi

Esercizio

Creare un'applicazione che mostri una finestra (di dimensioni 400*400) contenente tre pulsanti, un testo e un'area disegnabile (canvas).
I tre pulsanti devono essere associati ai tre valori rosso, blu e giallo.
Il click su un pulsante deve disegnare un cerchio nell'area disegnabile del colore associato al pulsante stesso. La posizione del cerchio deve essere casuale.
Il testo deve riportare, in ogni momento, il numero totale di cerchi disegnati.






```
public class DisegnatoreCerchiListener
        implements javafx.event.EventHandler {

    protected int diametro = 80;
    protected Color colore;
    protected Canvas target;

    public DisegnatoreCerchiListener(Color colore, Canvas target) {
        this.colore = colore;
        this.target = target;
    }

    @Override
    public void handle(Event event) {
        int x = Math.max((int) (Math.random() * (target.getWidth() - diametro)), 0);
        int y = Math.max((int) (Math.random() * (target.getHeight() - diametro)), 0);

        target.getGraphicsContext2D().setFill(colore);
        target.getGraphicsContext2D().fillOval(x, y, diametro, diametro);
    }
}
```

```
public class FinestraCerchiColorati extends Application {

    Button blu, rosso, giallo;
    Canvas canvas;
    Label circlesNrDisplay;

    private void inizializzaComponentiGrafici() {...9 lines }

    private Pane doLayout() {...35 lines }

    private void registraAscoltatori() {...16 lines }

    @Override
    public void start(Stage primaryStage) {
        inizializzaComponentiGrafici();
        registraAscoltatori();
        Pane contenuto = doLayout();
        Scene scene = new Scene(contenuto, 400, 400);
        primaryStage.setTitle("Cerchi colorati");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    class ContatoreListener implements EventHandler {...11 lines }
}
```

```
class ContatoreListener implements EventHandler {

    int circlesNr = 0;

    @Override
    public void handle(Event event) {
        circlesNr++;
        circlesNrDisplay.setText("Nr. " + circlesNr);

    }
}
```

Esercizio

Aggiungere un controllo che riporti, in ogni momento, il numero di cerchi sovrapposti.

Riferimenti

Classi interne, locali, anonime:

The Java Tutorial →
tutorial/java/javaOO/nested.html

Eckel, Thinking in Java → cap. 8 →
"Inner classes"

Gestione degli eventi in JavaFX:

Sito oracle.com → JavaFX tutorials →
"Handling JavaFX Events"