

## Esercitazione n° 6

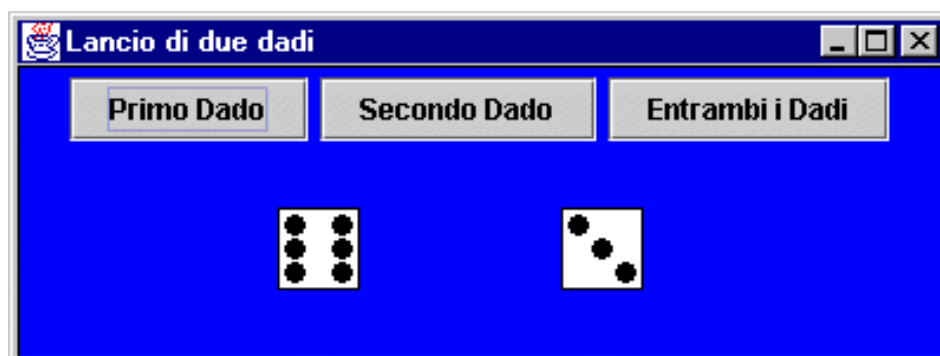
---

- Capacità di *analisi* e di *estensione* di progetti
- *Componenti grafici* e gestione di *eventi*
- Linguaggio Java:
  - il package `javax.swing` (`JFrame`, `JPanel`, `JButton`, ...)
  - `java.util.EventObject` e sue sottoclassi
- Esempio del lancio di due dadi
  - *Gestione di tre pulsanti* (lancio del primo dado, del secondo dado, di entrambi)
  - La pressione dei pulsanti deve scatenare l'aggiornamento di *1) frame 2) panel 3) valore del dado*

### Esercizio: lancio di una coppia di dadi (1)

---

- Componenti grafici di interfaccia:
  - **FrameConDadi** (con evento gestito da Terminator)
  - **PanelConDadi** (contenente il disegno dei due dadi)
  - tre **JButton** (il primo con gestione degli eventi già parzialmente realizzata)



## Esercizio: lancio di una coppia di dadi (2)

---

- Tre devono essere le conseguenze delle pressione dei pulsanti:
  - 1) aggiornamento del *valore del dado* corrispondente (valore estratto a sorte)
  - 2) cambio del *titolo del frame* (“premuto primo dado”, “premuto secondo dado”, “premuti entrambi i dadi”)
  - 3) cambio del *colore di background del panel* (rosso, blu, verde)
- *Quali gestori* per gli eventi (chi implementa `ActionListener`?)

## La classe `Dado` (1)

---

```
public class Dado {  
  
    private int val;  
    private FrameConDadi frame;  
  
    public Dado(int val) { this.val = val; }  
  
    public Dado(int val, FrameConDadi frame) {  
        this.val = val; this.frame = frame; }  
  
    public void setVal() {  
        val = (int)(6*Math.random()+1); }  
}
```

## La classe Dado (2)

---

```
public void disegna(Graphics g, int x, int y) {  
    g.setColor(Color.white); g.fillRect(x, y, 35, 35);  
    g.setColor(Color.black); g.drawRect(x, y, 34, 34);  
    if (val > 1)           // punto in alto a sinistra  
        g.fillOval(x+3, y+3, 9, 9);  
    if (val > 3)           // punto in alto a destra  
        g.fillOval(x+23, y+3, 9, 9);  
    if (val == 6)          // punto nel mezzo a sinistra  
        g.fillOval(x+3, y+13, 9, 9);  
    if (val % 2 == 1)      // punto al centro  
        g.fillOval(x+13, y+13, 9, 9);  
    if (val == 6)          // punto nel mezzo a destra  
        g.fillOval(x+23, y+13, 9, 9);  
    ... }  
}
```

## La classe GiocoDadi (1)

---

```
public class GiocoDadi {  
  
    public static void main(String args[]) {  
  
        JButton [] arrButt = new JButton [3];  
        arrButt[0] = new JButton ("Primo Dado");  
        arrButt[1] = new JButton ("Secondo Dado");  
        arrButt[2] = new JButton ("Entrambi i Dadi");  
  
        PanelConDadi panel=new PanelConDadi(4,2,arrButt);  
        FrameConDadi frame= new FrameConDadi ("Lancio di  
            due dadi", arrButt);  
        Container c=frame.getContentPane();  
        c.add(panel);  
        panel.add(arrButt[0]); panel.add(arrButt[1]);  
        panel.add(arrButt[2]);  
    }  
}
```

## La classe GiocoDadi (2)

---

```
// Collega gli eventi generati dal primo bottone
// all'oggetto panel, e gli eventi generati dalla
// finestra frame ad un oggetto di classe Terminator

    arrButt[0].addActionListener(panel);
    frame.addWindowListener(new Terminator ());

    frame.show();
}
}
```

## La classe PanelConDadi (1)

---

```
class PanelConDadi extends JPanel implements
    ActionListener {

    int larghezza=400, altezza=150;
    Dado[] arrDadi = new Dado[2];
    Color currColor = new Color(0,0,255);
    JButton arrButt[];

    public PanelConDadi(int val1, int val2, JButton[]
        arrButt) {
        this.arrButt = arrButt;
        arrDadi[0] = new Dado((int)(6*Math.random()+1));
        arrDadi[1] = new Dado((int)(6*Math.random()+1));
    }
    ...
}
```

## La classe PanelConDadi (2)

---

```
public void actionPerformed (ActionEvent e) {  
    // da completare...  
    Graphics g = getGraphics();  
    if (e.getSource()== arrButt[0]) {  
        currColor = Color.red;  
        arrDadi[0].setVal();  
        arrDadi[0].disegna(g, 110, 60); }  
    setBackground(currColor); }  
  
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    setBackground(currColor); g.setColor(Color.black);  
    g.drawRect(0,0,larghezza-1,altezza-1);  
    arrDadi[0].disegna(g, 110, 60);  
    arrDadi[1].disegna(g, 230, 60); } }
```

## La classe FrameConDadi

---

```
class FrameConDadi extends JFrame implements  
    ActionListener {  
  
    JButton arrButt [];  
  
    public FrameConDadi (String titolo, JButton []  
        arrButt) {  
        super(titolo);  
        setBounds(100,100,400,150);  
        this.arrButt = arrButt;  
    }  
  
    public void actionPerformed (ActionEvent e) {  
    }  
}
```

# La classe Terminator

---

```
class Terminator implements WindowListener {

    public void windowClosed(WindowEvent e) {}
    public void windowClosing(WindowEvent e) {
        System.exit(0); }
    public void windowOpened(WindowEvent e) {}
    public void windowIconified(WindowEvent e) {}
    public void windowDeiconified(WindowEvent e) {
        System.out.println("\nSono appena stato ingrandito!\n");
        FrameConDadi frame = (FrameConDadi) e.getSource();
        frame.setTitle("SONO QUI DI NUOVO!");
        frame.show(); }
    public void windowActivated(WindowEvent e) {}
    public void windowDeactivated(WindowEvent e) {}
}
```