

Grafici statistici

I grafici statistici forniscono una rappresentazione grafica dei dati ricavati da un fenomeno osservato. La scelta del tipo di grafico dipende da quale aspetto si vuole mettere in evidenza. Per esempio, un grafico a torta mostra le proporzioni dei vari elementi rispetto alla loro somma ed evidenzia la distribuzione della stessa quantità tra diverse entità. Un istogramma invece mostra le variazioni dei dati nel tempo oppure consente il confronto tra elementi diversi.

Il **grafico a torta** rappresenta i dati usando un cerchio e dividendolo in settori: ogni settore ha una dimensione che è proporzionale al dato e che corrisponde all'ampiezza dell'angolo che forma, la somma di tutti i settori deve formare il cerchio che corrisponde all'angolo di 360 gradi.

L'ampiezza x di un settore si calcola usando la proporzione:

$$x: 360 = \text{valore: totale dei valori}$$

Per rappresentare in Java i grafici a torta, si usa il metodo **fillArc**, specificando l'ampiezza del settore, calcolata con la formula precedente, e l'angolo da cui partire per disegnarlo.

Ogni settore viene disegnato con un colore diverso.

Il programma seguente descrive un'applet che, dati cinque valori di input, disegna il corrispondente grafico a torta.

L'interfaccia grafica per questo problema deve consentire l'inserimento dei cinque valori oltre alla visualizzazione del grafico statistico. Nella parte sinistra dell'applet vengono posizionate le cinque caselle di testo riservate all'inserimento dei valori e un bottone per confermare il disegno del grafico a torta. La parte centrale viene interamente occupata dall'area di disegno. L'applet *Statistica.class* viene richiamata da una pagina Web e posizionata al centro della finestra del browser.

APPLET JAVA (*Statistica.java*)

```
import java.awt.*;
import javax.swing.*;

public class Statistica extends JApplet
{
    private JPanel p = new JPanel();

    private JPanel comandi = new JPanel();
    private JTextField valori[] = new JTextField[5];
    private Color colori[] = new Color[5];
    private JButton disegna = new JButton("Disegna");

    private Torta t = new Torta(valori, colori);

    // dispone le componenti grafiche
    public void init()
    {
        disegna.addActionListener(t);
        inizializzaColori();
        comandi.setLayout(new GridLayout(7,1,0,10));
        comandi.add(new Label("— VALORI —", Label.CENTER));
        for(int i=0; i<valori.length; i++)
```

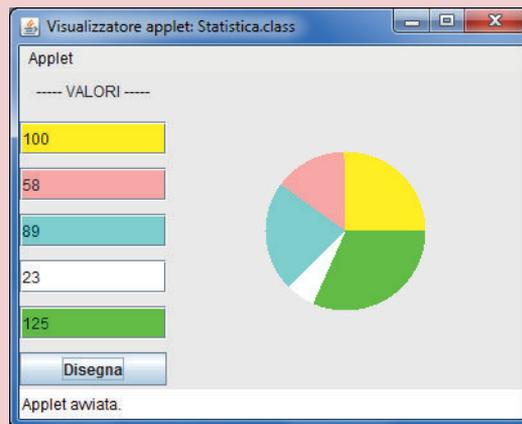
```

    {
        valori[i] = new JTextField(10);
        valori[i].setBackground(colori[i]);
        comandi.add(valori[i]);
    }
    comandi.add(disegna);

    p.setBackground(Color.lightGray);
    p.setLayout(new BorderLayout());
    p.add(comandi, "West");
    p.add(t, "Center");
    setContentPane(p);
}

public void inizializzaColori()
{
    colori[0] = Color.yellow;
    colori[1] = Color.pink;
    colori[2] = Color.cyan;
    colori[3] = Color.white;
    colori[4] = Color.green;
}
}

```



La seguente classe *Torta* rappresenta l'area grafica nella quale viene disegnato il grafico statistico.

IMPLEMENTAZIONE DELLA CLASSE (*Torta.java*)

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Torta extends JPanel implements ActionListener
{
    private JTextField valori[];
    private Color colori[];
    private double val[];
    private double totale = 0.0;
    private int x, y;
    private int dimensioni;
}

```

```

public Torta(JTextField valori[], Color colori[])
{
    this.valori = valori;
    this.colori = colori;
    val = new double[valori.length];
    for(int i=0; i<val.length; i++)
    {
        val[i] = 0.0;
    }
}

public void paintComponent(Graphics g)
{
    super.paintComponent(g);

    // calcola le dimensioni della torta
    int w = getSize().width;
    int h = getSize().height;
    if (w > h)
    {
        dimensioni = h/2;
    }
    else
    {
        dimensioni = w/2;
    }
    x = (w/2) - (dimensioni/2);
    y = (h/2) - (dimensioni/2);
    // se ci sono dei valori disegna la torta
    if (totale != 0.0)
    {
        disegnaTorta(g);
    }
}

// disegna il grafico a torta
public void disegnaTorta(Graphics g)
{
    int startAngle=0;
    int ampiezza=0;

    for(int i=0; i<val.length; i++)
    {
        if (val[i] != 0.0)
        {
            ampiezza = (int) Math.round(val[i]*360/totale);
            g.setColor(colori[i]);
            g.fillArc(x, y, dimensioni, dimensioni, startAngle, ampiezza);
            startAngle += ampiezza;
        }
    }
}
}

```

```

// gestore di eventi per il bottone
public void actionPerformed(ActionEvent e)
{
    String bottone = e.getActionCommand();
    String numero;

    if (bottone.equals("Disegna"))
    {
        totale = 0.0;
        for(int i=0; i<valori.length; i++)
        {
            try
            {
                numero = valori[i].getText();
                val[i] = Double.valueOf(numero).doubleValue();
            }
            catch(Exception exc)
            {
                val[i] = 0.0;
            }
            totale+=val[i];
        }

        repaint();
    }
}
}

```

Utilizzando il precedente esempio come traccia, si può disegnare l'istogramma, che è un altro grafico statistico di uso comune.

La rappresentazione grafica mediante un **istogramma** si ottiene visualizzando tanti rettangoli quanti sono i dati dell'osservazione statistica, aventi l'altezza uguale al valore che rappresentano.

Per calcolare le trasformazioni tra le coordinate del piano cartesiano e quelle dello schermo occorre tenere conto del fatto che, solitamente, il piano cartesiano usato per tracciare gli istogrammi possiede unità di misura diverse per l'asse x e per l'asse y .

La finestra del piano cartesiano che viene rappresentata sullo schermo è indicata dai quattro valori XMAX, YMAX, XMIN, YMIN. Gli istogrammi vengono rappresentati solitamente nel primo quadrante e nel caso ci siano valori negativi, nel quarto quadrante. Per questo motivo XMIN viene impostato a -1. Il valore di XMAX dipende dal numero dei rettangoli e viene fissato in base al numero di dati forniti. I valori di YMAX e YMIN sono legati ai valori massimi e minimi dei dati: al variare dei dati, cambiano anche i valori dell'ordinata massima e minima.

Per rappresentare le colonne dell'istogramma si usa il metodo **fillRect**, specificando come parametri le coordinate del vertice in alto a sinistra, la larghezza e l'altezza del rettangolo. Ogni rettangolo viene disegnato con un colore diverso.