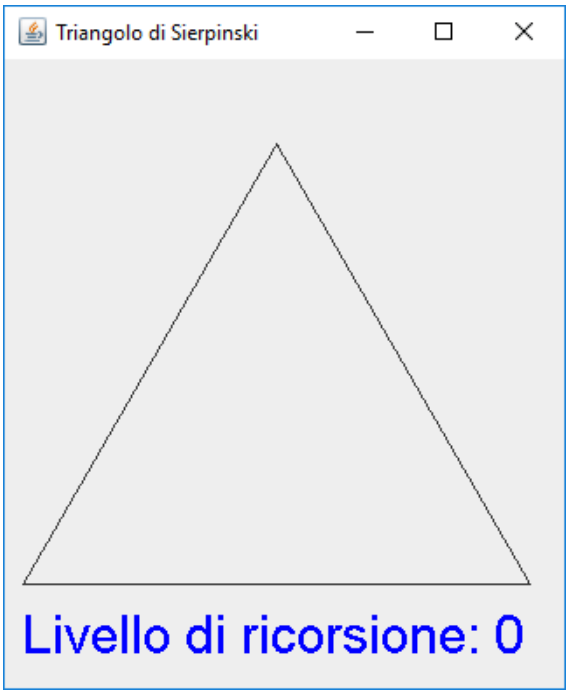
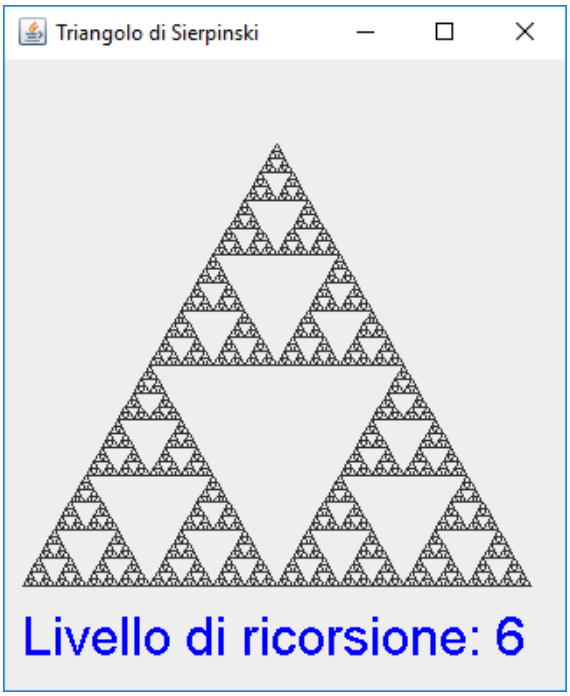
Triangolo di Sierpiński

```
/**
*   Frattale application
*   modifica da listing di José Juan Aliaga
*/
import javax.swing.*;
import java.awt.*;

public class Frattale extends JPanel {

        double xp1=300;
        double yp1=300;
        double xp2=10;
        double yp2=300;
        double sin60=Math.sin(3.14/3.0);
        int livello_ricorsione= 0;              // nb: se <=0 stesso triangolo iniziale  (*)
                                                // impostare inferiore a 13 per ritardo non eccessivo
        public Frattale() {
          setPreferredSize( new Dimension(320,360) );
        }
        public void paintComponent(Graphics g){
                super.paintComponent(g);
                paintRicorsivo(g,livello_ricorsione,xp1,yp1,xp2,yp2);
                g.setColor(Color.blue);   // colore scritta
                                                /* cambia la dimensione a 30pt del font corrente */
                g.setFont(new Font(g.getFont().getFontName(),Font.PLAIN,30) );
                g.drawString("Livello di ricorsione: " + livello_ricorsione,10,340);  // in fondo al pannello
        }
        private void paintRicorsivo(Graphics g, int i, double xp12, double yp12, double xp22, double yp22 ) {
                 double dx=(xp22-xp12)/2.0;
                 double dy=(yp22-yp12)/2.0;
                 double xp32=xp12+dx-2*dy*sin60;
                 double yp32=yp12+dy+2*dx*sin60;

                 double dx1=(xp22+xp12)/2.0;
                 double dy1=(yp22+yp12)/2.0;
                 double dx2=(xp32+xp22)/2.0;
                 double dy2=(yp32+yp22)/2.0;
                 double dx3=(xp12+xp32)/2.0;
                 double dy3=(yp12+yp32)/2.0;

                 if(i<=0){                                                      // (*)
                          g.drawLine((int)xp12,(int)yp12,(int)xp22,(int)yp22);
                          g.drawLine((int)xp22,(int)yp22,(int)xp32,(int)yp32);
                          g.drawLine((int)xp32,(int)yp32,(int)xp12,(int)yp12);
                 }
                 else{
                          paintRicorsivo(g,i-1,xp12,yp12,dx1,dy1);
                          paintRicorsivo(g,i-1,dx1,dy1,xp22,yp22);
                          paintRicorsivo(g,i-1,dx3,dy3,dx2,dy2);
                 }
        }
```

```
        public static void main(String[] args) {
                Frattale triangolo = new Frattale();                    // pannello personalizzato
                 JFrame fr = new JFrame("Triangolo di Sierpinski");
                 fr.setContentPane(triangolo);
                 fr.pack();
                 fr.setVisible(true);
                 fr.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
         }
}
```



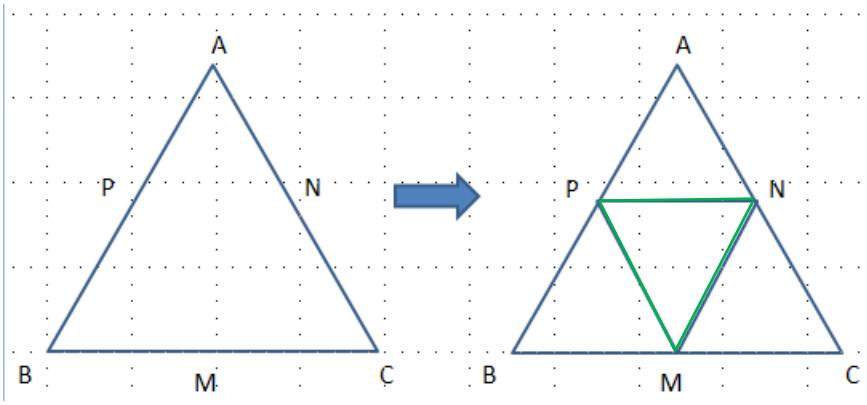


Il **metodo ricorsivo** riceve, oltreal ***contesto grafico*** per il disegno, i parametri che identificano le coordinate delle linee ed il numero delle ricorsioni da eseguire (**i**).

```
private void paintRicorsivo(Graphics g, int i, double xp12, double yp12, double xp22, double yp22 ) {}
```

All'inizio dell'esecuzione si verifica la ***condizione di terminazione***, vale a dire, se il metodo deve richiamare se stesso o meno.

Se la ***condizione di terminazione*** non è verificata, si tracciano le tre linee per costruire triangoli equilateri di lato 1/2 come in figura.

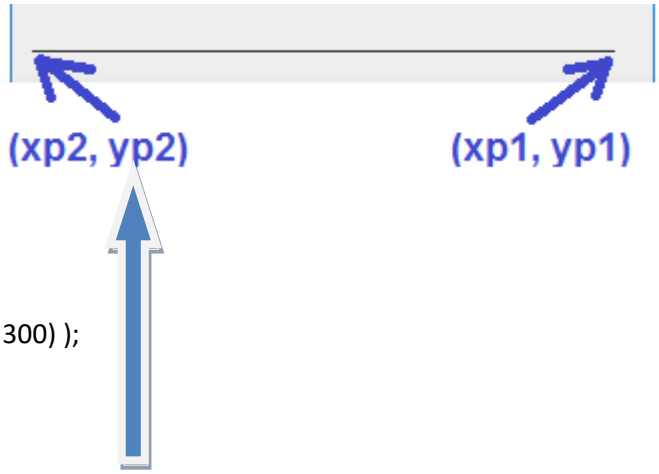


Si procede con chiamata del metodo in modo **ricorsivo** per ciascuna delle sezioni: ogni volta riducendo il numero delle ricorsioni.

```
paintRicorsivo(g,i-1,xp12,yp12,dx1,dy1);
paintRicorsivo(g,i-1,dx1,dy1,xp22,yp22);
paintRicorsivo(g,i-1,dx3,dy3,dx2,dy2);
```

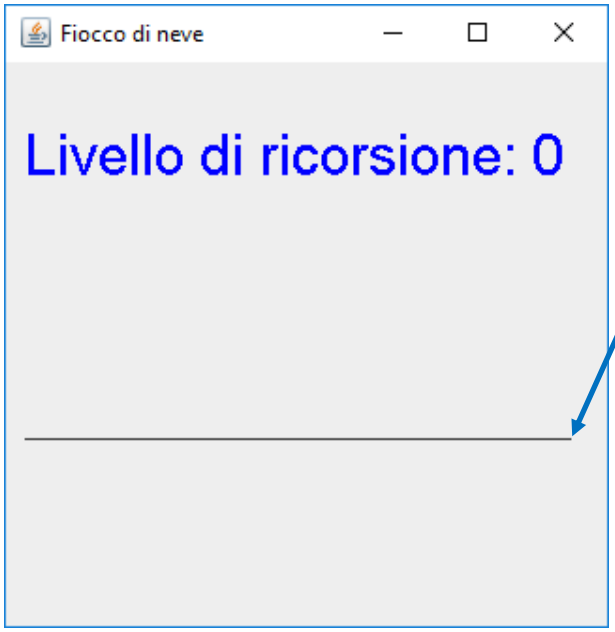
La **curva di Koch** o **fiocco di neve**

```
/**
*   Koch application
*   modifica da listing di José Juan Aliaga
*/
 import javax.swing.*;
import java.awt.*;

public class Koch extends JPanel {

        double xp1=300;
        double yp1=200;
        double xp2=10;
        double yp2=200;
        double sin60=Math.sin(3.14/3.);
        int livello_ricorsione=6;

        public Koch() {

                setPreferredSize( new Dimension(320, 300) );
         }

        public void paintComponent(Graphics g){
                super.paintComponent(g);
                 paintRicorsivo(g,livello_ricorsione,xp1,yp1,xp2,yp2);
                g.setColor(Color.blue);   // colore scritta
                                                        /* cambia la dimensione a 30pt del font corrente */
                 g.setFont(new Font(g.getFont().getFontName(),Font.PLAIN,30) );
                g.drawString("Livello di ricorsione: " + livello_ricorsione,10,60);  // in cima al pannello
        }
        private void paintRicorsivo(Graphics g, int i, double xp12, double yp12, double xp22, double yp22 ) {
                 double dx=(xp22-xp12)/3.0;
                 double dy=(yp22-yp12)/3.0;
                 double xx=xp12+3*dx/2.0-dy*sin60;
                 double yy=yp12+3*dy/2.0+dx*sin60;
                 if(i<=0){
                     g.drawLine((int)xp12,(int)yp12,(int)xp22,(int)yp22);
                 }
                 else{
                    paintRicorsivo(g,i-1,xp12,yp12,xp12+dx,yp12+dy);
                    paintRicorsivo(g,i-1,xp12+dx,yp12+dy,xx,yy);
                    paintRicorsivo(g,i-1,xx,yy,xp22-dx,yp22-dy);
                    paintRicorsivo(g,i-1,xp22-dx,yp22-dy,xp22,yp22);
                 }
        }
        public static void main(String[] args) {

                    Koch k  = new Koch();                // pannello personalizzato
                    JFrame fr = new JFrame("Fiocco di neve");
                    fr.setContentPane(k);
                    fr.pack();
                    fr.setVisible(true);
                    fr.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
         }
}
```



Il **metodo ricorsivo** riceve, oltreal ***contesto grafico*** per il disegno, i parametri che identificano le coordinate della *linea "iniziale"* ed il numero delle ricorsioni da eseguire (**i**).

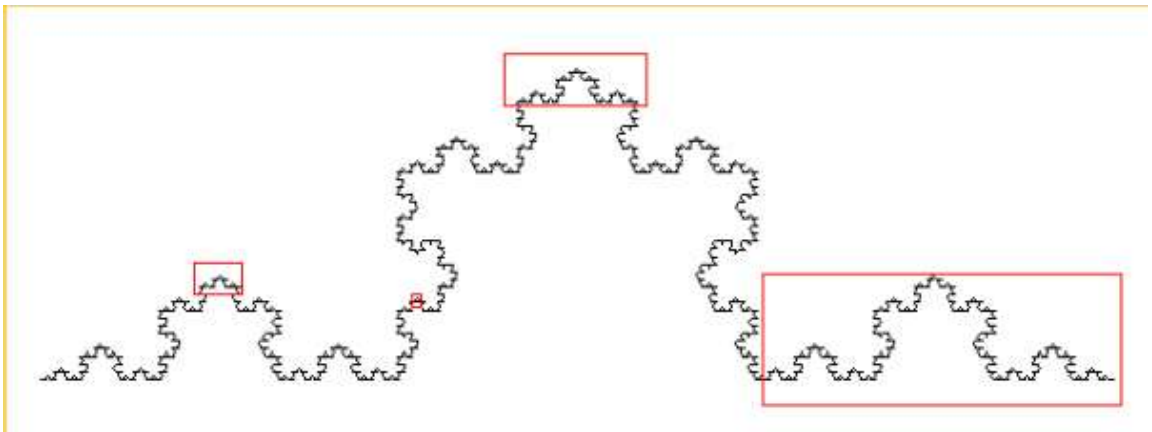
```
private void paintRicorsivo(Graphics g, int i, double xp12, double yp12, double xp22, double yp22 ) {}
```



All'inizio dell'esecuzione si verifica la ***condizione di terminazione***, vale a dire, se il metodo deve richiamare se stesso o meno.

Se la ***condizione di terminazione*** non è stata raggiunta, si divide la linea nelle quattro sezioni necessarie.

Si procede con chiamata del metodo in **modo ricorsivo** per ciascuna delle sezioni: ogni volta riducendo il numero delle ricorsioni.



Animazioni da [wikipedia](https://it.wikipedia.org/wiki/Frattale#/media/File:Von_Koch_curve.gif)
https://it.wikipedia.org/wiki/Frattale#/media/File:Von_Koch_curve.gif