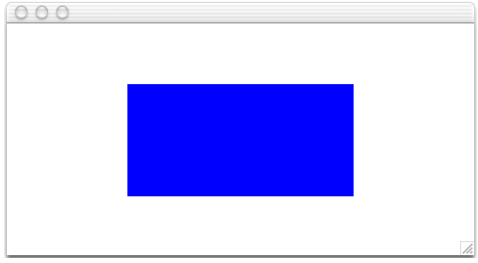
```
  public MidiEvent makeEvent(int comd, int chan, int one, int two, int tick) {
      MidiEvent event = null;
      try {
         ShortMessage a = new ShortMessage();
         a.setMessage(comd, chan, one, two);
          event = new MidiEvent(a, tick);

       }catch(Exception e) { }
       return event;
   } // close method


class MyDrawPanel extends JPanel implements ControllerEventListener {
     boolean msg = false;

     public void controlChange(ShortMessage event) {
        msg = true;
        repaint();
     }

     public void paintComponent(Graphics g) {
      if (msg) {

        Graphics2D g2 = (Graphics2D) g;

        int r = (int) (Math.random() * 250);
        int gr = (int) (Math.random() * 250);
        int b = (int) (Math.random() * 250);

        g.setColor(new Color(r,gr,b));

        int ht = (int) ((Math.random() * 120) + 10);
        int width = (int) ((Math.random() * 120) + 10);

        int x = (int) ((Math.random() * 40) + 10);
        int y = (int) ((Math.random() * 40) + 10);

        g.fillRect(x,y,ht, width);
        msg = false;

      } // close if
    } // close method
  }  // close inner class

} // close class
```

Pool Puzzle

Your ***job*** is to take code snippets from the pool and place them into the blank lines in the code. You **may** use the same snippet more than once, and you won't need to use all the snippets. Your ***goal*** is to make a class that will compile and run and produce the output listed.

Output

The Amazing, Shrinking, Blue Rectangle. This program will produce a blue rectangle that will shrink and shrink and disappear into a field of white.



```
import javax.swing.*;
import java.awt.*;
public class Animate {
    int x = 1;
    int y = 1;
    public static void main (String[] args) {
       Animate gui = new Animate ();
       gui.go();
   }
   public void go() {
       JFrame __________ = new JFrame();
       frame.setDefaultCloseOperation(
                    JFrame.EXIT_ON_CLOSE);
       ____________________________________;
       __________.getContentPane().add(drawP);
       ____________________________;
       __________.setVisible(true);
       for (int i=0; i<124; _________________) {
       ____________________;
       ____________________;
       try {
          Thread.sleep(50);
       } catch(Exception ex) { }
       }
   }
   class MyDrawP extends JPanel {
       public void paintComponent (Graphic
                                   __________) {
          ____________________________________;
          ____________________________________;
          ____________________________________;
          ____________________________________;
      }
   }
}
```

Note: Each snippet from the pool can be used more than once!

```
g.fillRect(x,y,x-500,y-250)
g.fillRect(x,y,500-x*2,250-y*2)
g.fillRect(500-x*2,250-y*2,x,y)
x++   g.fillRect(0,0,250,500)
y++   g.fillRect(0,0,500,250)

i++
i++, y++
i++, y++, x++

Animate frame = new Animate()
MyDrawP  drawP = new MyDrawP()
ContentPane drawP = new ContentPane()

drawP.setSize(500,270)
frame.setSize(500,270)
panel.setSize(500,270)

g.setColor(blue)
g.setColor(white)
g.setColor(Color.blue)
g.setColor(Color.white)

g
draw
frame
panel

drawP.paint()
draw.repaint()
drawP.repaint()
```