

```
/// <- LIBRERIE -> ///
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class PaintBrutto extends JFrame {
```

```
    private static final long serialVersionUID = 1L;
```



```
/// <- VARIABILI -> ///
```

```
// JPanel
```

```
// panel per disegno bottoni
```

```
JPanel pDisegno, pBottoni;
```

```
// JButton
```

```
// array che contiene tutti i bottoni per non creare 8 variabili diverse
```

```
JButton bottoni[] = new JButton[8];
```

```
// Color
```

```
// array di colori che verranno assegnati al bottone corrispettivo
```

```
Color colori[] = {Color.black, Color.red, Color.green, Color.blue, Color.cyan, Color.magenta, Color.yellow, Color.white};
```

```
// cliccando su un bottone si salverà il suo colore all'interno di questa variabile per poter poi colorare
```

```
// istanzio la variabile a nero così anche se non clicco su di un bottone inizio a colorare con quest'ultimo colore
```

```
Color coloreFinale = Color.black;
```

```
// array di stringhe che contiene i nomi che verranno assegnati al bottone corrispettivo
```

```
String nomiBottoni[] = {"Nero", "Rosso", "Verde", "Blu", "Ciano", "Magenta", "Giallo", "Cancella"};
```

```
/// <- MAIN -> ///
```

```
public static void main(String[] args) {
```

```
    new PaintBrutto();
```

```
}
```

```
/// <- PAINTBRUTTO -> ///
```

```
PaintBrutto() {
```

```
    setTitle("Paint Brutto");
```

```
    setSize(900, 900);
```

```
    setLocationRelativeTo(null);
```

```
    setExtendedState(JFrame.MAXIMIZED_BOTH);
```

```
    addWindowListener(new WindowAdapter() {
```

```
        public void windowClosing(WindowEvent e) {
```

```
            // chiedo conferma all'utente per chiudere la finestra
```

```
            int conferma = JOptionPane.showConfirmDialog(null,
```

```
                "Chiudere La Finestra?",
```

```

"Conferma Chiusura",
JOptionPane.YES_NO_OPTION);

// se l'utente conferma chiude la finestra, se rifiuta non fare niente
if ( conferma == JOptionPane.YES_OPTION ) System.exit(0);
else setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);

}
});

// creo il contenitore dei due panel
Container c = getContentPane();
c.setLayout(new BorderLayout());

// creo il pannello nel quale andrò a disegnare
pDisegno = new JPanel(null); // setto il layout null
// coloro
pDisegno.addMouseListener(new MouseAdapter() {
    public void mouseDragged(MouseEvent e) {
        Graphics g = getGraphics();

        g.setColor(coloreFinale);
        g.fillOval(e.getXOnScreen(), e.getYOnScreen(), 10, 10);
    }
});
c.add(pDisegno, BorderLayout.CENTER);

// creo il pannello nel quale andrò a inserire i bottoni
pBottoni = new JPanel(new GridLayout(8, 1)); // setto il GridLayout

// ciclo for per creare e inserire i mie bottoni all'interno del panel senza dover creare 8 bottoni
for (int i = 0; i < bottoni.length; i++) {

    final int iFinal = i; // devo creare una variabile final per poter richiamare i all'interno dei metodi

    // assegno nome e colore a ogni bottone
    bottoni[i] = new JButton(nomiBottoni[i]);
    bottoni[i].setBackground(colori[i]);

    // se fa parte di uno dei primi 7 bottoni gli assegno un ActionListener
    if (i < bottoni.length - 1) {
        bottoni[i].addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // salvo il colore del bottone in questa variabile per poi andare a colorare
                coloreFinale = colori[iFinal];
            }
        });
    } else { // nel caso il bottone schiacciato sia quello per cancellare
        bottoni[i].addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // cancello il panel dove disegno
                repaint();
            }
        });
    }
}
});

```

```

    }

    // aggiungo i bottoni al panel
    pBottoni.add(bottoni[i]);
    }
    c.add(pBottoni, BorderLayout.EAST);

    setVisible(true);
}

/// <- REPAINT -> ///
// quando questo metodo viene richiamato dal bottone cancella, cancellerà tutto quello che c'è sul pannello pDisegn
o
public void paint(Graphics g) {
    super.paint(g);

    // con il comando seguente posso cancellare tutto lo schermo o solo una parte di esso come in questo caso
    // io non l'ho utilizzato perchè avrei dovuto strutturare l'esercizio in maniera diversa
    // g.clearRect(100, 100, 200, 150); // in questo caso cancello soltanto un rettangolo 200x150 partendo da x,y = 100
    }

}

```