JAVA LAB 3 (2)

**Problem:**

Write a JAVA Application that user can draw something on the **JPanel**.

**Requirement:**

1. You can’t use any static field and method except the main function.
2. You need to provide four “button” including **three color pens (black, red, blue) and an eraser** for user to choose (just like picture below).
3. The “button” is painted on JPanel , and is activated by mouse event.
4. The size of “button” is **20 \* 20px**
5. Stroke width need to be **4.0f**.
6. **Paint on the JPanel directly**, you don’t have to store the brush path.
7. **You need to make the painting smoothly**.

|  |  |
| --- | --- |
|  |  |
| You need to do like this !! | Not good |

**Class & method:**

Class

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.\*;

Interface

**MouseListener**

public void mousePressed(MouseEvent event){}

public void mouseReleased(MouseEvent event){}

public void mouseClicked(MouseEvent event){}

public void mouseEntered(MouseEvent event){}

public void mouseExited(MouseEvent event){}

**MouseMotionListener**

public void mouseDragged( MouseEvent event ){}

public void mouseMoved( MouseEvent event ){}

1. addMouseListener()
2. addMouseMotionListener()
3. event.getX()
4. event.getY()
5. Graphics g = getGraphics();
6. Graphics2D g2 = (Graphics2D) g;
7. clearRect(int x, int y, int width, int height);
8. fillOval(int x, int y, int width, int height);
9. drawLine(int x1, int y1, int x2, int y2)
10. setStroke(new BasicStroke(lineWidth)); // lineWidth = 4.0f
11. drawRect (int x, int y, int width, int height);
12. fillRect (int x, int y, int width, int height);
13. setColor(Color color);
14. setVisible();
15. paintComponent();
16. setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE)

**Referance:**

1. **Use JAVA Doc**

<http://docs.oracle.com/javase/7/docs/api/overview-summary.html>

1. **getGraphics** VS **paintComponent**

getGraphics method

* Can be null
* Is a "snap shot" of the last paint process
* Anything painted to it will be lost on the next paint cycle

You should avoid using getGraphics and simply use what is past to the paintComponent method.

In theory, there is no difference between them, but if you want what you have painted to survive between repaints, then you should be using paintComponent

[source](http://stackoverflow.com/questions/21129138/difference-between-graphics-object-of-getgraphics-and-paintcomponent)

1. **Graphics** VS **Graphics2D**

[See here](http://docs.oracle.com/javase/7/docs/api/java/awt/Graphics2D.html)