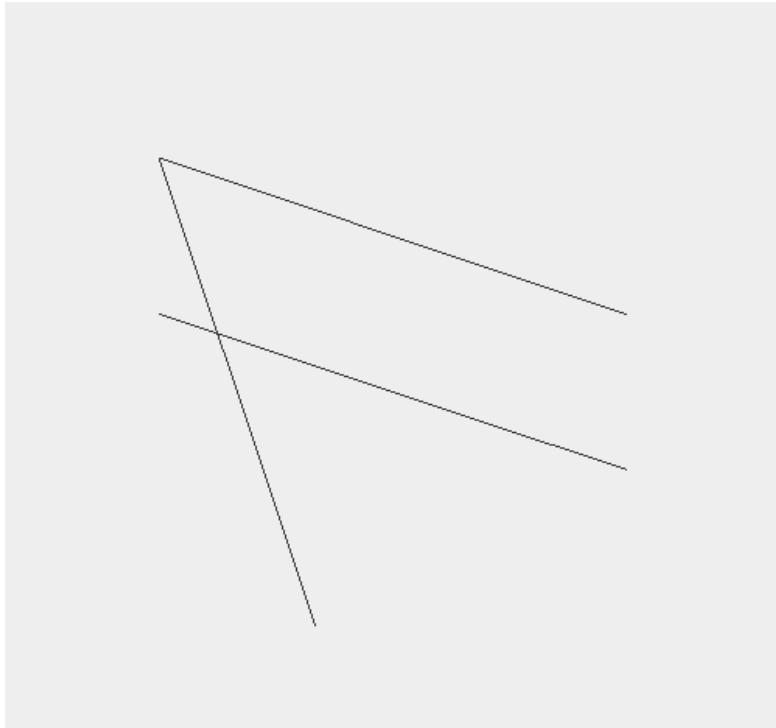


Exercise 2

```
1  import javax.swing.*;
2  import java.awt.*;
3
4  public class Lab2_Exercise2 extends JPanel {
5
6      public static void main(String[] args) {
7          JFrame frame = new JFrame();
8          Lab2_Exercise2 lab = new Lab2_Exercise2();
9          frame.setTitle("Lab2");
10         frame.setSize(1000, 1000);
11         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12         frame.add(lab);
13         frame.setVisible(true);
14     }
15
16     public void DDALine(Graphics g, int x1, int y1, int x2, int y2) {
17         double dx = x2 - x1;
18         double dy = y2 - y1;
19
20         double x,y;
21         double m = dy / dx;
22
23         if (m <= 1 && m >= 0) {
24             y = Math.min(y1,y2);
25             for (x = Math.min(x1,x2); x <= Math.max(x1,x2); x++) {
26                 y = y + m;
27                 plot(g, (int)x, (int)y);
28             }
29         } else if (m >= -1 && m < 0) {
30             y = Math.max(y1,y2);
31             for (x = Math.max(x1,x2); x >= Math.min(x1,x2); x--) {
32                 y = y + m;
33                 plot(g, (int)x, (int)y);
34             }
35         } else if (m > 1) {
36             x = Math.min(x1,x2);
37             for (y = Math.min(y1,y2); y <= Math.max(y1,y2); y++) {
38                 x = x + 1 / m;
39                 plot(g, (int)x, (int)y);
40             }
41         } else {
42             x = Math.max(x1,x2);
43             for (y = Math.max(y1,y2); y <= Math.min(y1,y2); y--) {
44                 x = x + 1 / m;
45                 plot(g, (int)x, (int)y);
46             }
47         }
48     }
49
50     public void plot(Graphics g, int x, int y) {
51         g.drawLine(x, y, x, y); // Draw a point at (x, y)
52     }
53
54     @Override
55     public void paintComponent(Graphics g) {
56         DDALine(g, 100, 100, 400, 200);
57         DDALine(g, 400, 300, 100, 200);
58         DDALine(g, 100, 100, 200, 400);
59     }
60 }
```



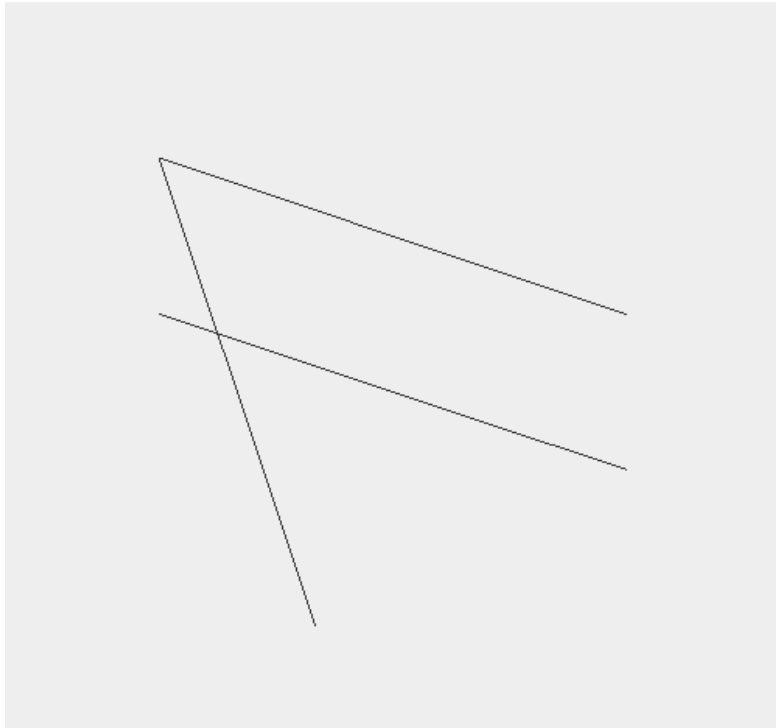
5. How can we improve DDALine method to be more readable?

ตอบ ทำให้ค่าที่รับมาจาก parameter สามารถรับจำนวนเต็มได้โดยไม่ต้องนำมาแปลงจากทศนิยมเป็นจำนวนเต็มทีหลัง เพราะ การเก็บใช้ byte เยอะ และมีปัญหาเรื่องความแม่นยำ

Exercise 3



```
1  import javax.swing.*;
2  import java.awt.*;
3
4  public class Lab2_Exercise3 extends JPanel {
5      public static void main(String[] args) {
6          JFrame frame = new JFrame();
7          Lab2_Exercise3 lab = new Lab2_Exercise3();
8          frame.setTitle("Lab2");
9          frame.setSize(1000, 1000);
10         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11         frame.add(lab);
12         frame.setVisible(true);
13     }
14
15     public void bresenhamLine(Graphics g ,int x1, int y1, int x2, int y2) {
16         int dx = Math.abs(x2 - x1);
17         int dy = Math.abs(y2 - y1);
18
19         int sx = (x1 < x2) ? 1 : -1;
20         int sy = (y1 < y2) ? 1 : -1;
21         boolean isSwap = false;
22
23         if (dy > dx) {
24             // Swap
25             int temp = dx;
26             dx = dy;
27             dy = temp;
28             isSwap = true;
29         }
30         int D = 2 * dy - dx;
31         int x = x1;
32         int y = y1;
33
34         for (int i = 1; i <= dx; i++) {
35             plot(g ,x, y);
36             if (D >= 0) {
37                 if (isSwap) x += sx;
38                 else y += sy;
39                 D -= 2 * dx;
40             }
41             if (isSwap) y += sy;
42             else x += sx;
43             D += 2 * dy;
44         }
45     }
46
47     public void plot(Graphics g,int x, int y) {
48         g.drawLine(x, y, x, y); // Draw a point at (x, y)
49     }
50
51     @Override
52     public void paintComponent(Graphics g) {
53         super.paintComponent(g);
54
55         bresenhamLine(g ,100, 100, 400, 200);
56         bresenhamLine(g, 400, 300, 100, 200);
57         bresenhamLine(g, 100, 100, 200, 400);
58     }
59 }
```



5. Why the bresenhamLine method should be drawn as the pseudo code above?

ตอบ เพราะเป็นวิธีที่เร็ว และหลีกเลี่ยงการใช้ทศนิยมกับการปัดเศษ คำนวณแค่จำนวนเต็มเท่านั้น ทำให้มีความแม่นยำ และใช้พื้นที่น้อย