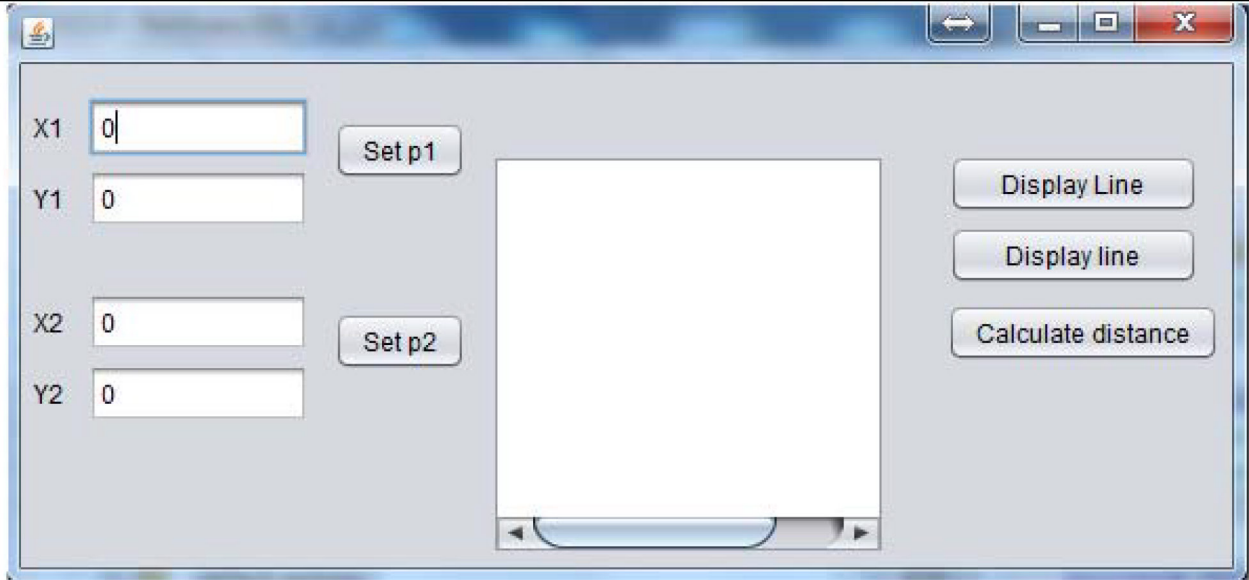


Fundamentals of Programming II Lab 08

13

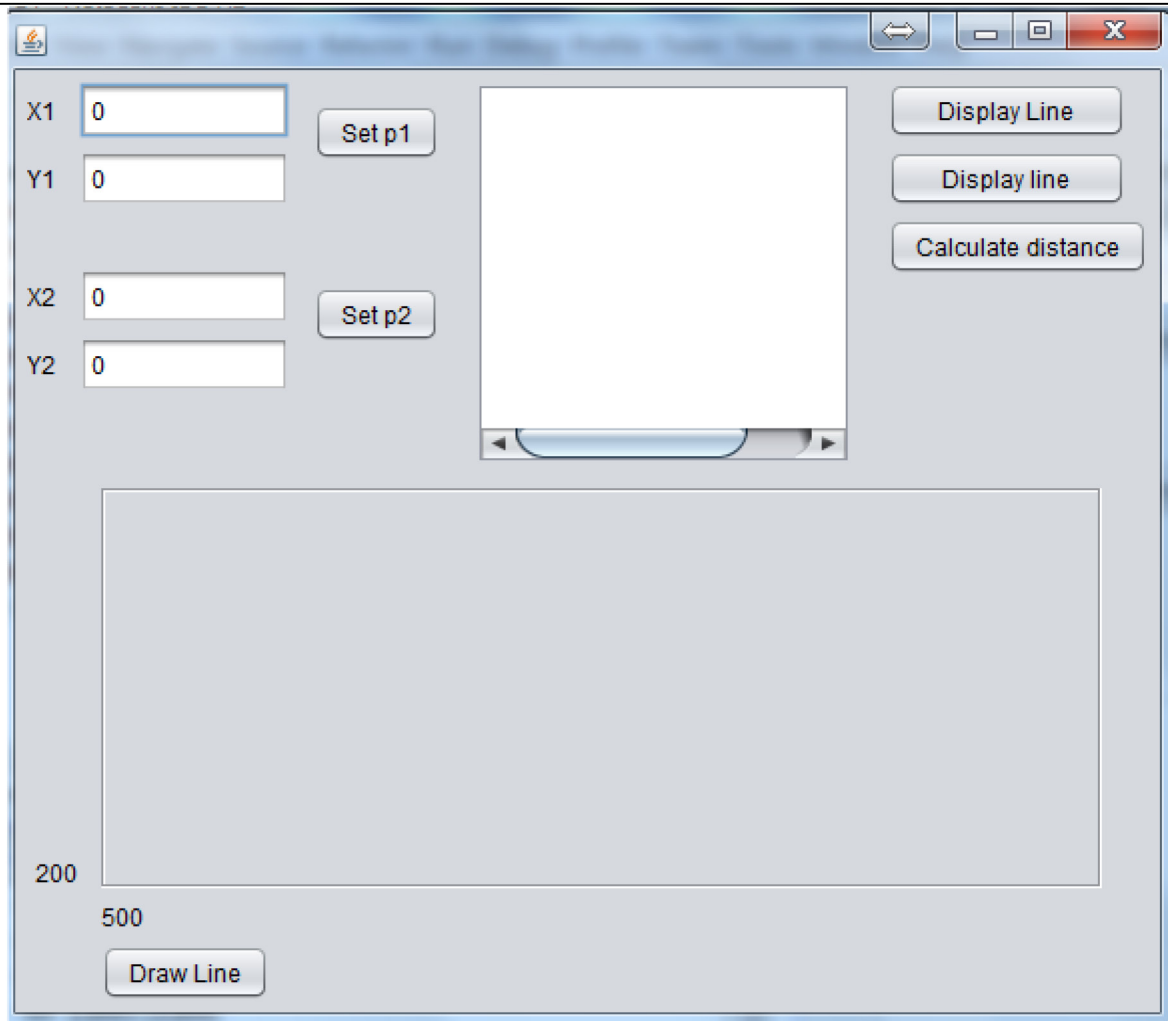


```

public class LineXY {
    int x1 = 0, x2 = 0, y1 = 0, y2 = 0;
    void setP1(int x, int y) {
        x1 = x;
        y1 = y;
    }
    void setP2(int x, int y) {
        x2 = x;
        y2 = y;
    }
    String getP1() {
        return "P1: x1 = " + x1 + "\ty1 = " + y1;
    }
    void getP1(int p[])
    {
        p[0]=x1;
        p[1]=y1;
    }
    String getP2() {
        return "P2: x2 = " + x2 + "\ty2 = " + y2;
    }
    void getP2(int p[])
    {
        p[0]=x2;
    }
}
    
```

		<pre> p[1]=y2; } double getLength() { int dx=x2-x1; int dy=y2-y1; double l=Math.sqrt((dx*dx) + (dy*dy)); return l; } }</pre>
	object creation	LineXY line=new LineXY();
	Set P1	<pre> line.setP1(Integer.parseInt(jTextField1.getText()), Integer.parseInt(jTextField2.getText())); jTextArea1.append(line.getP1()+"\n");</pre>
	Set P2	<pre> line.setP2(Integer.parseInt(jTextField3.getText()), Integer.parseInt(jTextField4.getText())); jTextArea1.append(line.getP2()+"\n");</pre>
	display line	<pre> jTextArea1.append(line.getP1()+"\n"); jTextArea1.append(line.getP2()+"\n");</pre>
	Display line	<pre> int p[]=new int[2]; line.getP1(p); jTextArea1.append("P1 : X = "+p[0]+" \tY = "+p[1]+" \n"); line.getP2(p); jTextArea1.append("P2 : X = "+p[0]+" \tY = "+p[1]+" \n");</pre>
	Calculate Distance	<pre> jTextArea1.append("Line length = "+line.getLength()+" \n");</pre>

14



```
public class LineXY {

    int x1 = 0, x2 = 0, y1 = 0, y2 = 0;

    void setP1(int x, int y) {
        x1 = x;
        y1 = y;
    }

    void setP2(int x, int y) {
        x2 = x;
        y2 = y;
    }

    int getPX(String P) {
        if (P.equals("P1")) {
```

```

        return x1;
    } else {
        return x2;
    }
}

int getPY(String P) {
    if (P.equals("P1")) {
        return y1;
    } else {
        return y2;
    }
}

String getP1() {
    return "P1: x1 = " + x1 + "\ty1 = " + y1;
}

void getP1(int p[]) {
    p[0] = x1;
    p[1] = y1;
}

String getP2() {
    return "P2: x2 = " + x2 + "\ty2 = " + y2;
}

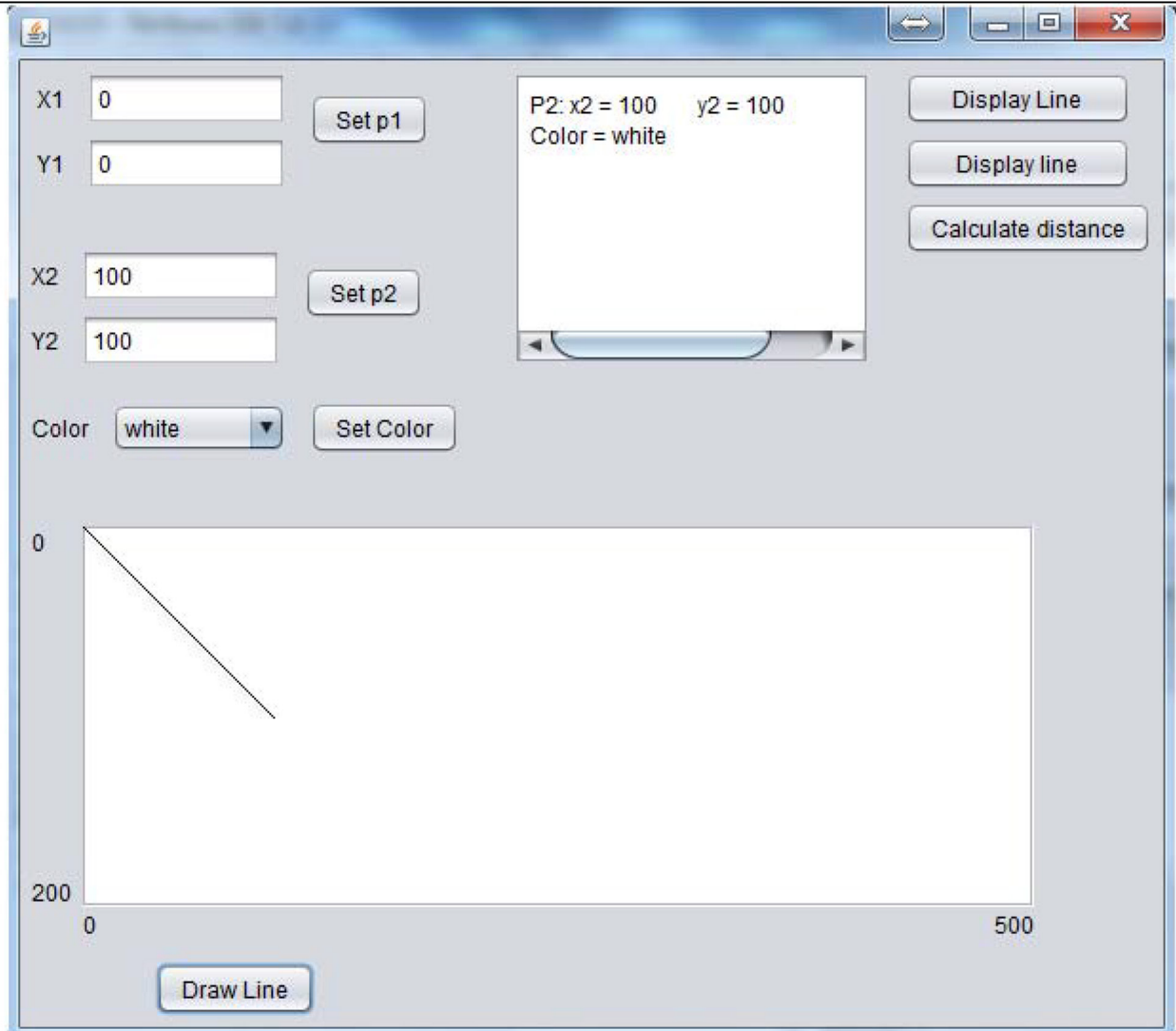
void getP2(int p[]) {
    p[0] = x2;
    p[1] = y2;
}

double getLength() {
    int dx = x2 - x1;
    int dy = y2 - y1;

```


		<pre> double l = Math.sqrt((dx * dx) + (dy * dy)); return l; } }</pre>
Draw line		<pre> jPanel1.getGraphics().drawLine(line.getPX("P1"), line.getPY("P1"), line.getPX("P2"), line.getPY("P2"));</pre>

15



LineXY

```

public class LineXY {
    int x1, x2, y1, y2;
    String color;
    LineXY()
    {
        x1=0;y1=0;x2=0;y2=0;
        color="gray";
    }
}
```

```

}
LineXY(int x1,int y1,int x2,int y2, String color)
{
    this.x1=x1;
    this.x2=x2;
    this.y1=y1;
    this.y2=y2;
    this.color=color;
}
void setP1(int x, int y) {
    x1 = x;
    y1 = y;
}

void setP2(int x, int y) {
    x2 = x;
    y2 = y;
}
void setColor(String color)
{
    this.color=color;
}
int getPX(String P) {
    if (P.equals("P1")) {
        return x1;
    } else {
        return x2;
    }
}

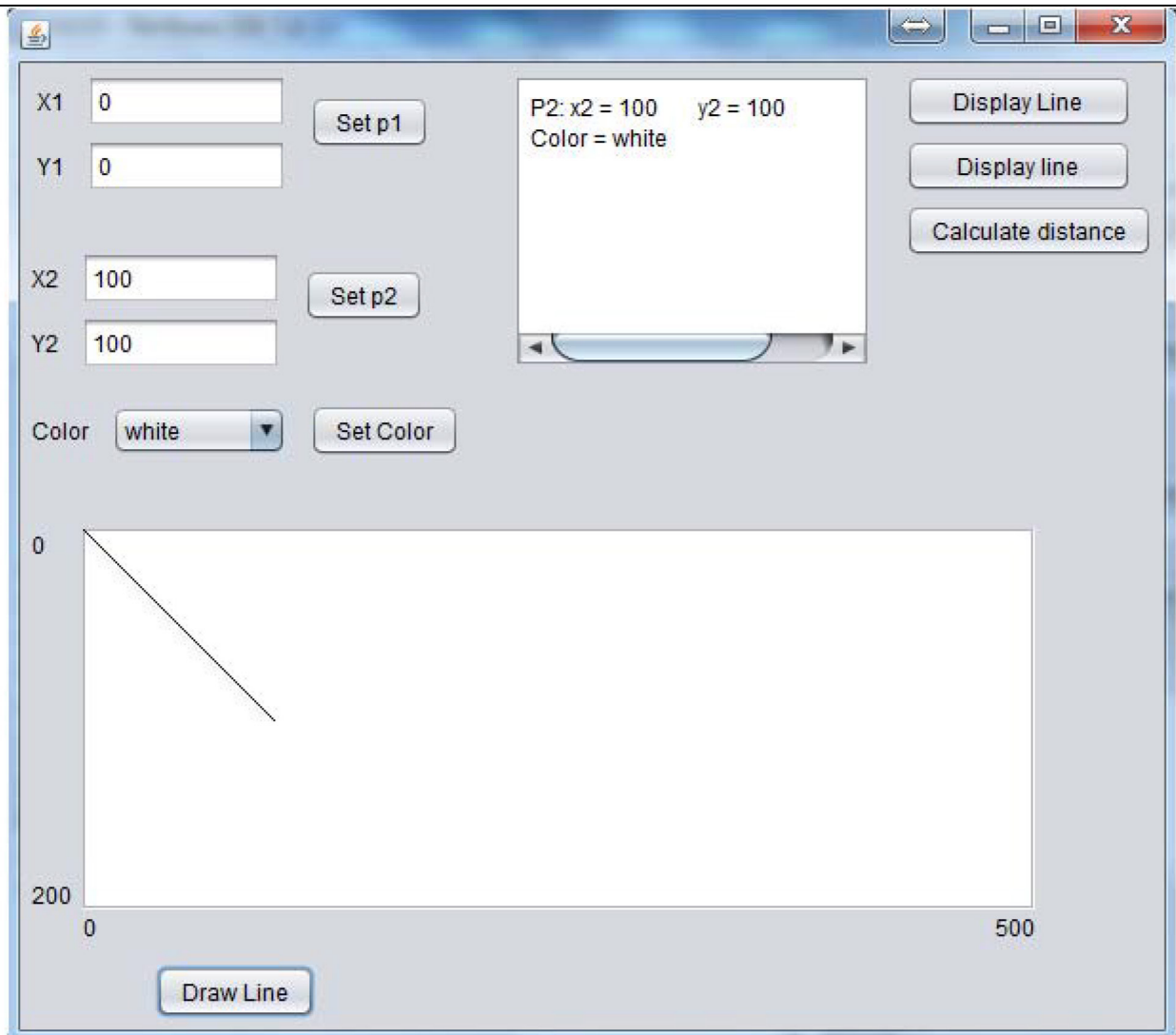
int getPY(String P) {
    if (P.equals("P1")) {
        return y1;
    } else {
        return y2;
    }
}

```

	<pre> } String getP1() { return "P1: x1 = " + x1 + "\ty1 = " + y1; } void getP1(int p[]) { p[0] = x1; p[1] = y1; } String getP2() { return "P2: x2 = " + x2 + "\ty2 = " + y2; } void getP2(int p[]) { p[0] = x2; p[1] = y2; } double getLength() { int dx = x2 - x1; int dy = y2 - y1; double l = Math.sqrt((dx * dx) + (dy * dy)); return l; } String getColor() { return color; } } </pre>
Object declaration	<pre> LineXY line= new LineXY(0, 0, 0, 0, "gray"); </pre>
Set color	<pre> line.setColor(jComboBox1.getSelectedItem().toString()); jTextArea1.append("Color = " + line.getColor() + "\n"); </pre>

		<pre> if(line.getColor().equalsIgnoreCase("grey")) jPanel1.setBackground(Color.gray); else if(line.getColor().equalsIgnoreCase("white")) jPanel1.setBackground(Color.white); else if(line.getColor().equalsIgnoreCase("red")) jPanel1.setBackground(Color.red); else if(line.getColor().equalsIgnoreCase("green")) jPanel1.setBackground(Color.green); else if(line.getColor().equalsIgnoreCase("blue")) jPanel1.setBackground(Color.blue); </pre>
	Display line	<pre> jTextArea1.append("Color = " + line.getColor() + "\n"); </pre>
	Display line	<pre> jTextArea1.append("Color = " + line.getColor() + "\n"); </pre>

16



```
public class PointXY {
    public String coordinateType="cartesian ";
    int X;
    int Y;
    public void initPointXY()
    {
        this.X=0;
        this.Y=0;
    }
    int getX()
    {
        return X;
    }
    int getY()
    {
```

	<pre> return Y; } String getCoordinateType() { return coordinateType; } void setX(int X) { this.X=X; } void setY(int Y) { this.Y=Y; } void setCoordinateType(String st) { coordinateType=st; } } </pre>
	<pre> public class LineXY { PointXY p1=new PointXY(); PointXY p2=new PointXY(); Color color; LineXY() { p1.setX(0); p1.setY(0); p2.setX(0); p2.setY(0); color = Color.gray; } LineXY(int x1, int y1, int x2, int y2, String color) { p1.setX(x1); </pre>

```

    p1.setY(y1);
    p2.setX(x2);
    p2.setY(y2);
    setColor(color);
}

void setP1(int x, int y) {
    p1.setX(x);
    p1.setY(y);
}

void setP2(int x, int y) {
    p2.setX(x);
    p2.setY(y);
}

void setColor(String color) {
    if (color.equalsIgnoreCase("grey")) {
        this.color = (Color.gray);
    } else if (color.equalsIgnoreCase("white")) {
        this.color = (Color.white);
    } else if (color.equalsIgnoreCase("red")) {
        this.color = (Color.red);
    } else if (color.equalsIgnoreCase("green")) {
        this.color = (Color.green);
    } else if (color.equalsIgnoreCase("blue")) {
        this.color = (Color.blue);
    }
}

int getPX(String P) {
    if (P.equals("P1")) {
        return p1.getX();
    } else {
        return p2.getX();
    }
}

```

```

    }
}

int getPY(String P) {
    if (P.equals("P1")) {
        return p1.getY();
    } else {
        return p2.getY();
    }
}

String getP1() {
    return "P1: x1 = " + p1.getX() + "\ty1 = " + p1.getY();
}

void getP1(int p[]) {
    p[0] = p1.getX();
    p[1] = p1.getY();
}

String getP2() {
    return "P2: x2 = " + p2.getX() + "\ty2 = " + p2.getY();
}

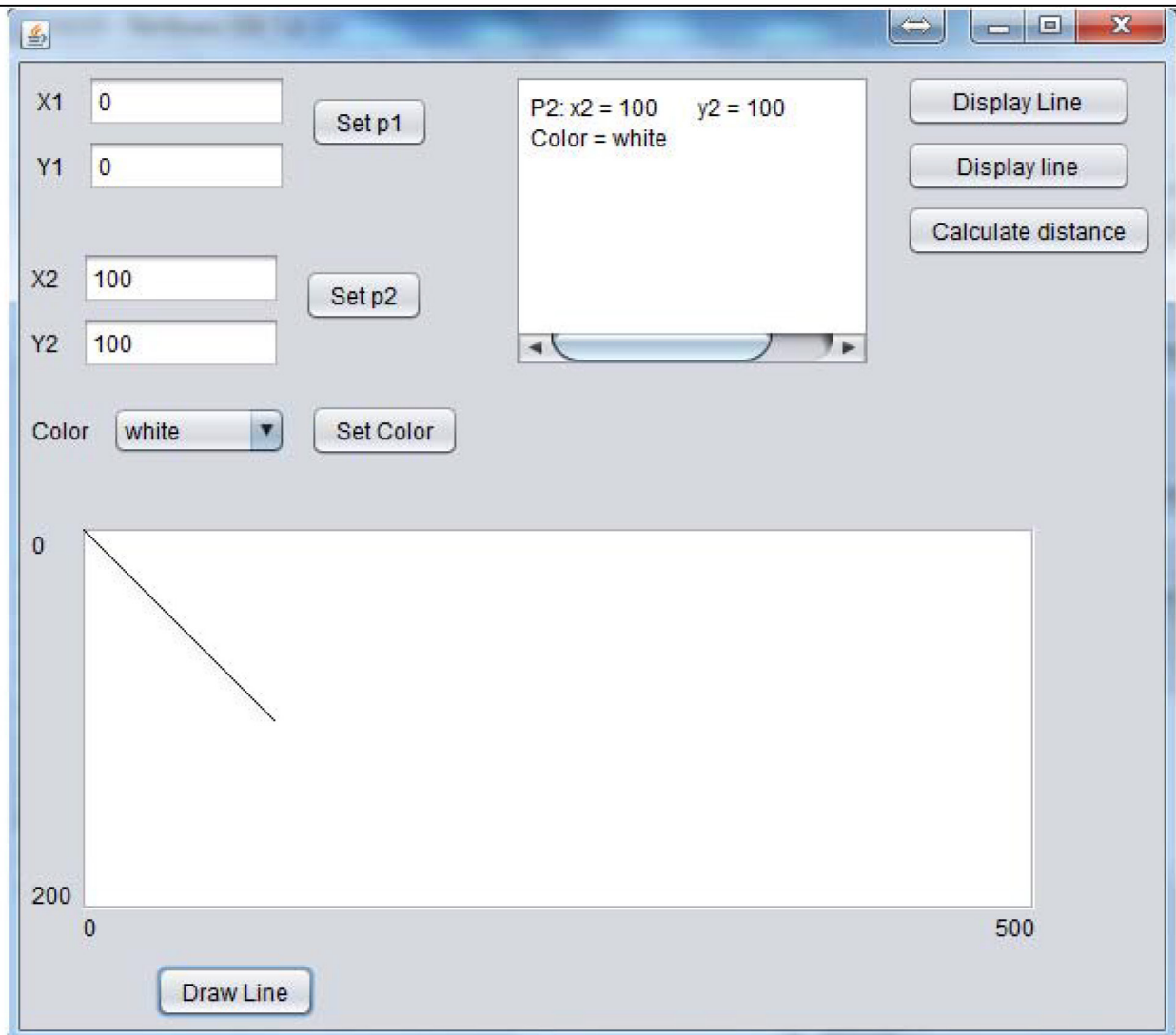
void getP2(int p[]) {
    p[0] = p2.getX();
    p[1] = p2.getY();
}

double getLength() {
    int dx = p2.getX() - p1.getX();
    int dy = p2.getY() - p1.getY();
    double l = Math.sqrt((dx * dx) + (dy * dy));
    return l;
}

```


		<pre>String getColor() { if (color == (Color.gray)) { return "gray"; } else if (color == (Color.white)) { return "white"; } else if (color == (Color.red)) { return "red"; } else if (color == (Color.blue)) { return "blue"; } else if (color == (Color.green)) { return "green"; } else return "gray"; } }</pre>
		<pre>LineXY line= new LineXY(0, 0, 0, 0, "gray");</pre>

17



```
public class LineXY {

    PointXY p1;
    PointXY p2;
    Color color;

    LineXY() {
        p1=new PointXY(0,0);
        p2=new PointXY(0,0);
        color = Color.gray;
    }

    LineXY(PointXY P1,PointXY P2, Color C) {
        p1=new PointXY(P1.getX(),P1.getY());
        p2=new PointXY(P2.getX(),P2.getY());
```

```

        color=C;
    }
    void SetP1(PointXY p){
        p1.setX(p.getX());
        p1.setY(p.getY());
    }
    void SetP2(PointXY p){
        p2.setX(p.getX());
        p2.setY(p.getY());
    }

    void setColor(Color C) {
        this.color=C;
    }

    int getPX(String P) {
        if (P.equals("P1")) {
            return p1.getX();
        } else {
            return p2.getX();
        }
    }

    int getPY(String P) {
        if (P.equals("P1")) {
            return p1.getY();
        } else {
            return p2.getY();
        }
    }

    String getP1() {
        return "P1: x1 = " + p1.getX() + "\ty1 = " + p1.getY();
    }

```

```

void getP1(int p[]) {
    p[0] = p1.getX();
    p[1] = p1.getY();
}

String getP2() {
    return "P2: x2 = " + p2.getX() + "\ty2 = " + p2.getY();
}

void getP2(int p[]) {
    p[0] = p2.getX();
    p[1] = p2.getY();
}

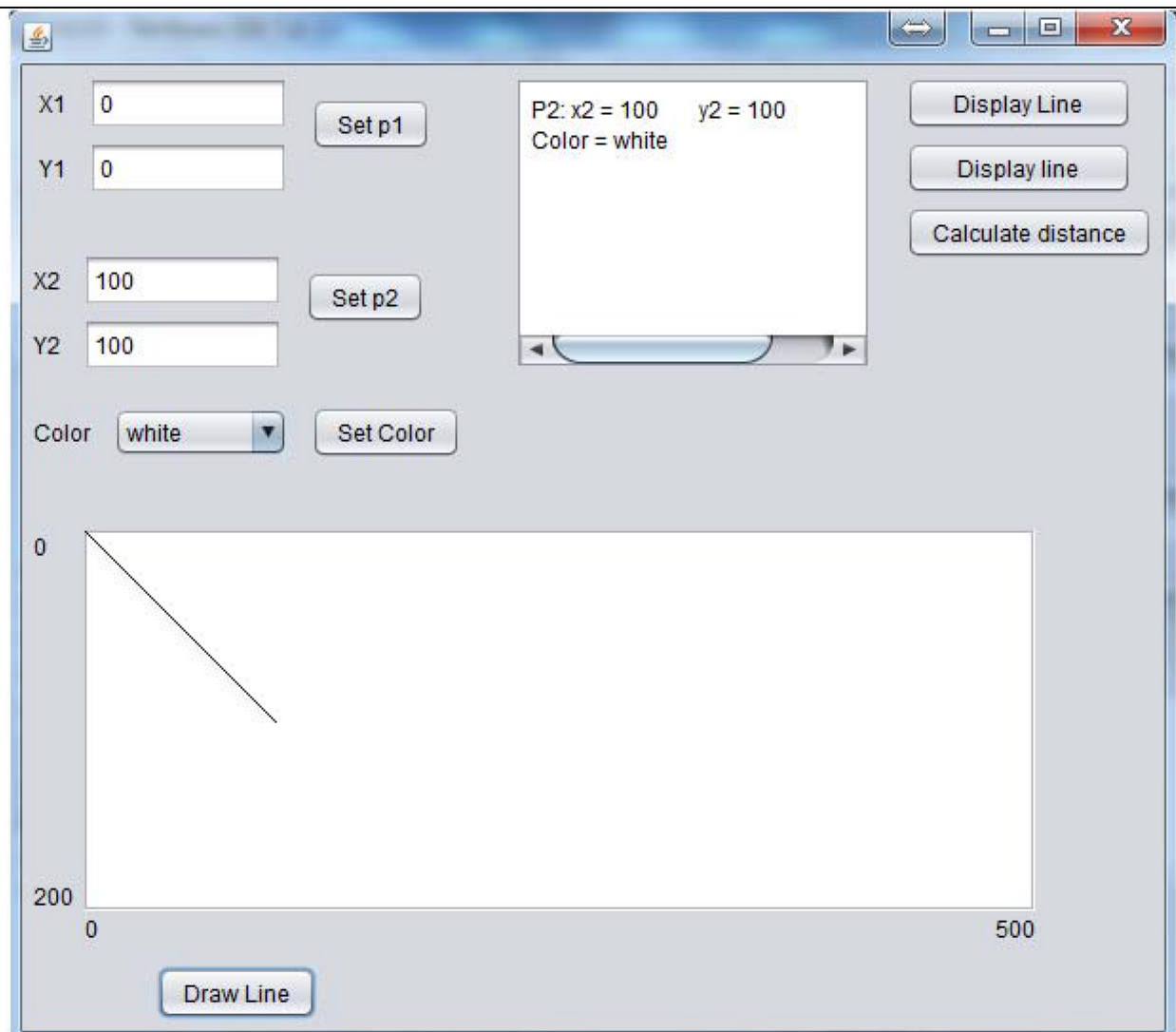
double getLength() {
    int dx = p2.getX() - p1.getX();
    int dy = p2.getY() - p1.getY();
    double l = Math.sqrt((dx * dx) + (dy * dy));
    return l;
}

String getColor() {
    if (color == (Color.gray)) {
        return "gray";
    } else if (color == (Color.white)) {
        return "white";
    } else if (color == (Color.red)) {
        return "red";
    } else if (color == (Color.blue)) {
        return "blue";
    } else if (color == (Color.green)) {
        return "green";
    }
    else return "gray";
}
}

```

Set P1	<pre>line.SetP1(new PointXY(Integer.parseInt(jTextField1.getText()), Integer.parseInt(jTextField1.getText()))); jTextArea1.append(line.getP1() + "\n");</pre>
Set P2	<pre>line.SetP2(new PointXY(Integer.parseInt(jTextField3.getText()), Integer.parseInt(jTextField4.getText()))); jTextArea1.append(line.getP2() + "\n");</pre>

18



```
public class LineXY {
```

```
    PointXY p1;
    PointXY p2;
    Color color;
```

```

LineXY() {
    p1 = new PointXY(0, 0);
    p2 = new PointXY(0, 0);
    color = Color.gray;
}

LineXY(PointXY P1, PointXY P2, Color C) {
    p1 = new PointXY(P1.getX(), P1.getY());
    p2 = new PointXY(P2.getX(), P2.getY());
    color = C;
}

void SetP1(PointXY p) {
    p1.setX(p.getX());
    p1.setY(p.getY());
}

void SetP2(PointXY p) {
    p2.setX(p.getX());
    p2.setY(p.getY());
}

void setColor(Color C) {
    this.color = C;
}

PointXY getP1() {
    return p1;
}

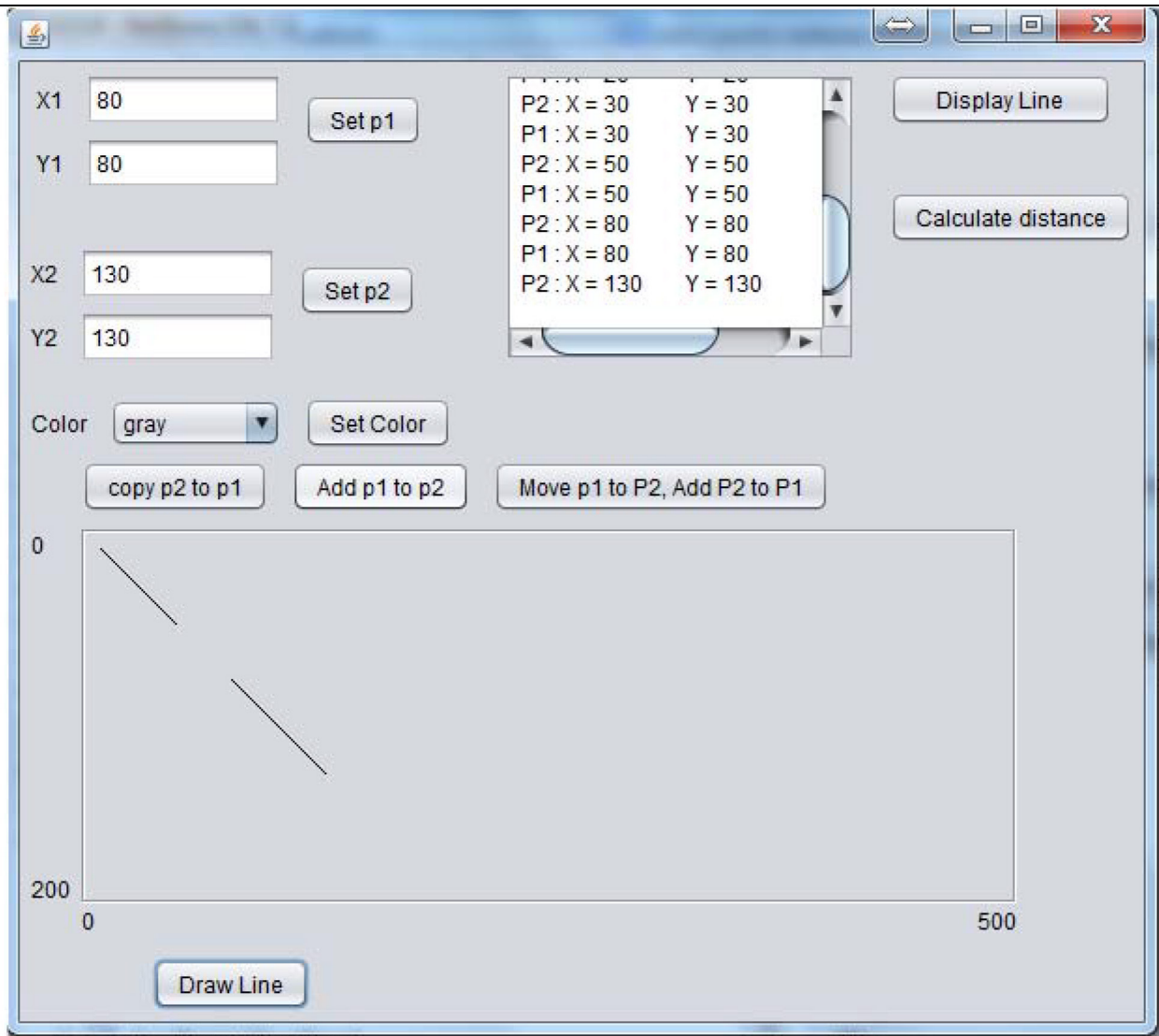
PointXY getP2() {
    return p2;
}

double getLength() {
    int dx = p2.getX() - p1.getX();

```

		<pre> int dy = p2.getY() - p1.getY(); double l = Math.sqrt((dx * dx) + (dy * dy)); return l; } Color getColor() { return color; } } </pre>
Display line		<pre> jTextArea1.append("P1 : X = "+line.getP1().getX()+"\tY = " +line.getP1().getY()+ "\n"); jTextArea1.append("P2 : X = "+line.getP2().getX()+"\tY = " +line.getP2().getY()+ "\n"); jTextArea1.append("Color = " + line.getColor().toString() + "\n"); </pre>
Draw line		<pre> jPanel1.getGraphics().drawLine(line.getP1().getX(), line.getP1().getY(), line.getP2().getX(), line.getP2().getY()); </pre>

19

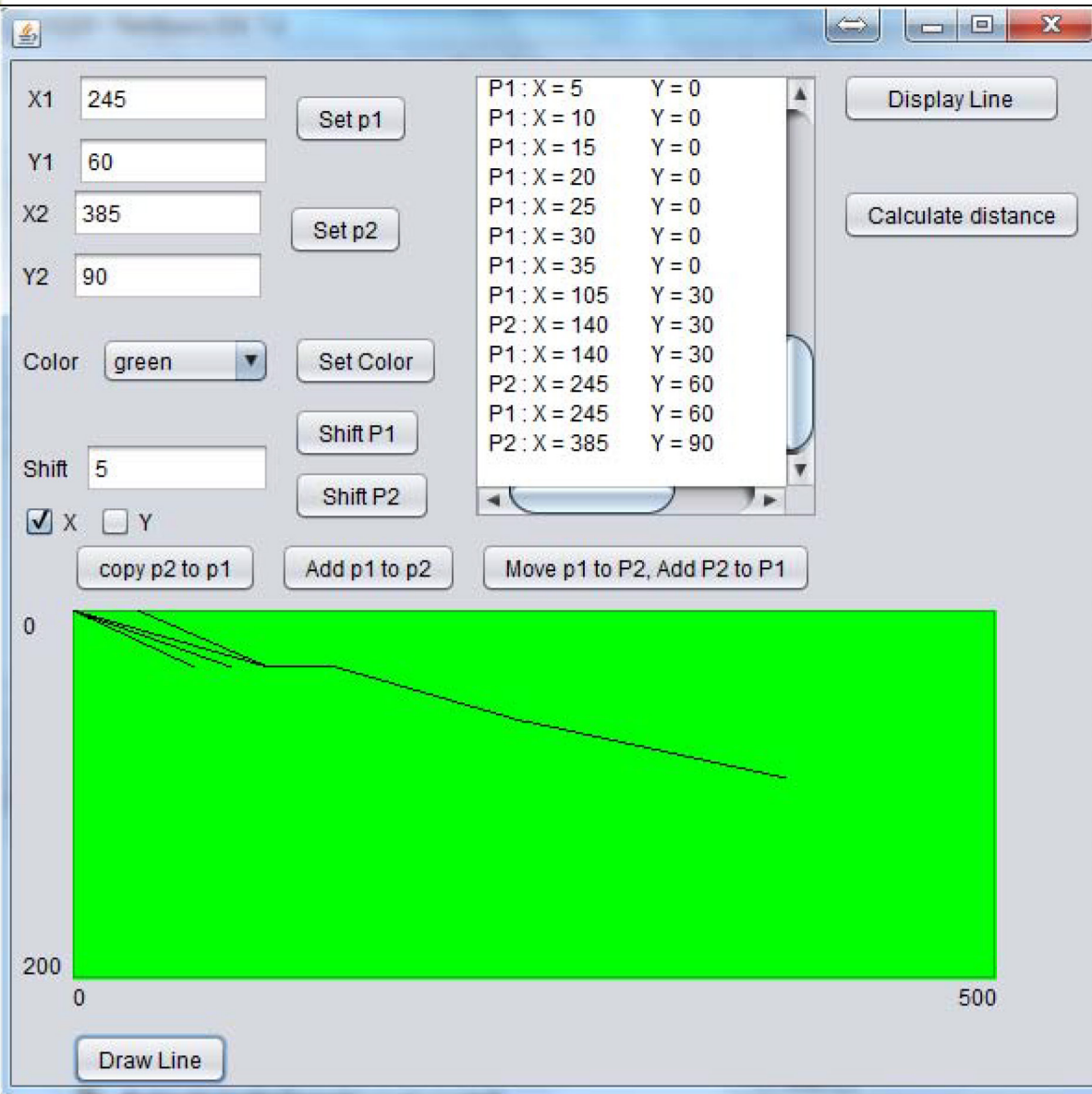


```
public class PointXY {
    public String coordinateType="cartesian ";
    int X;
    int Y;
    PointXY(int x, int y)
    {
        X=x;
        Y=y;
    }
    public void initPointXY()
    {
        this.X=0;
        this.Y=0;
    }
    int getX()
```


	<pre> { return X; } int getY() { return Y; } String getCoordinateType() { return coordinateType; } void setX(int X) { this.X=X; } void setY(int Y) { this.Y=Y; } void setCoordinateType(String st) { coordinateType=st; } void add(PointXY p) { X=p.getX()+X; Y=p.getY()+Y; } void copy(PointXY p) { X=p.getX(); Y=p.getY(); } } </pre>
Copy	line.p1.copy(line.p2);

p2 to p1	<pre> jTextArea1.append("P1 : X = " + line.getP1().getX() + "\tY = " + line.getP1().getY() + "\n"); jTextField1.setText(Integer.toString(line.getP1().getX())); jTextField2.setText(Integer.toString(line.getP1().getY())); </pre>
Add p1 to p2	<pre> line.p2.add(line.p1); jTextArea1.append("P2 : X = " + line.getP2().getX() + "\tY = " + line.getP2().getY() + "\n"); jTextField3.setText(Integer.toString(line.getP2().getX())); jTextField4.setText(Integer.toString(line.getP2().getY())); </pre>
Move p1 to P2, Add P2 to P1	<pre> PointXY p = new PointXY(line.p1.getX(), line.p1.getY()); line.p1.copy(line.p2); line.p2.add(p); jTextArea1.append("P1 : X = " + line.getP1().getX() + "\tY = " + line.getP1().getY() + "\n"); jTextArea1.append("P2 : X = " + line.getP2().getX() + "\tY = " + line.getP2().getY() + "\n"); jTextField1.setText(Integer.toString(line.getP1().getX())); jTextField2.setText(Integer.toString(line.getP1().getY())); jTextField3.setText(Integer.toString(line.getP2().getX())); jTextField4.setText(Integer.toString(line.getP2().getY())); </pre>

2
0



```
public class PointXY {
    public String coordinateType="cartesian ";
    int X;
    int Y;
    PointXY(int x, int y)
    {
        X=x;
        Y=y;
    }
    public void initPointXY()
    {
        this.X=0;
        this.Y=0;
    }
}
```

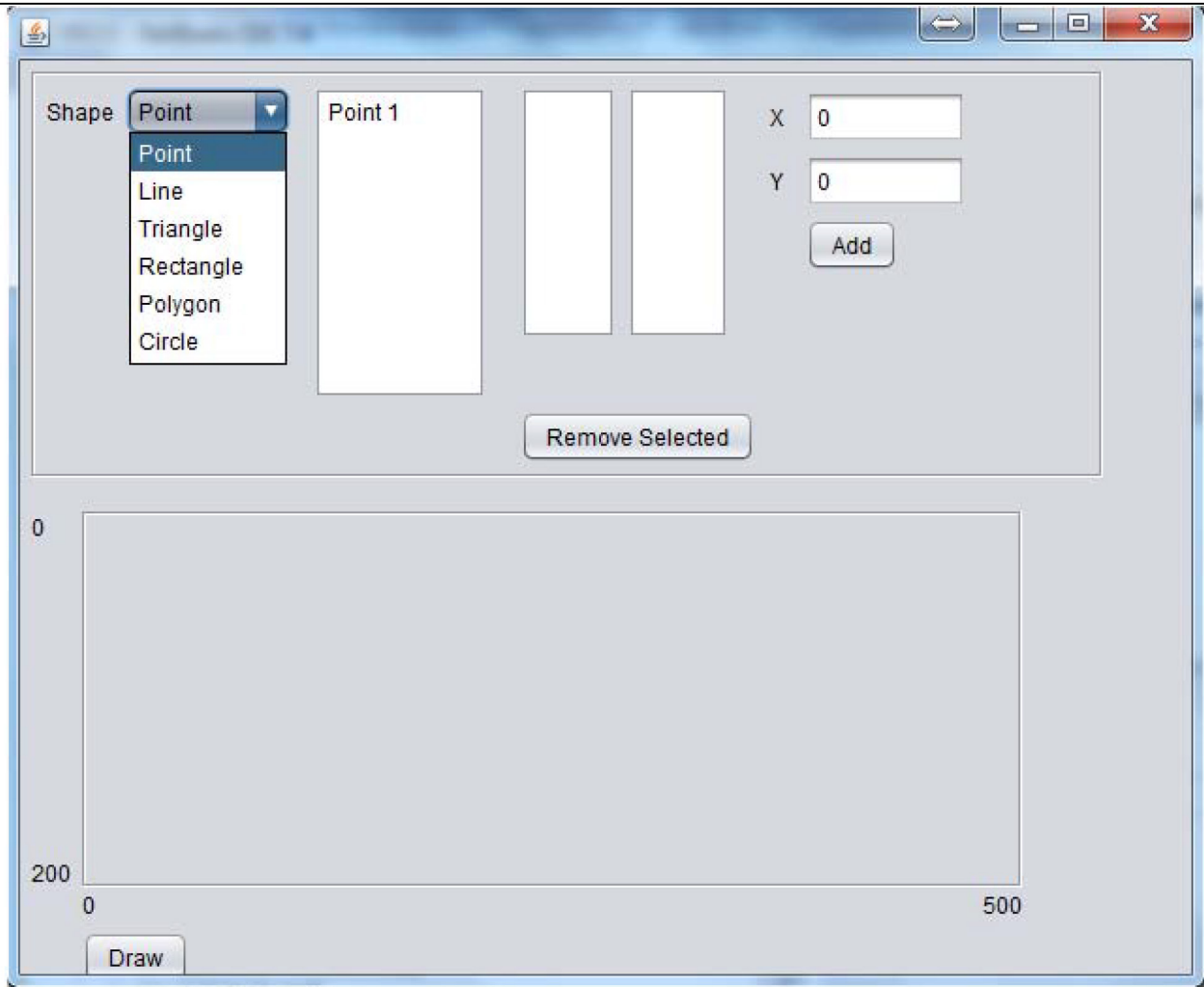
```

    }
    int getX()
    {
        return X;
    }
    int getY()
    {
        return Y;
    }
    String getCoordinateType()
    {
        return coordinateType;
    }
    void setX(int X)
    {
        this.X=X;
    }
    void setY(int Y)
    {
        this.Y=Y;
    }
    void setCoordinateType(String st)
    {
        coordinateType=st;
    }
    void add(PointXY p)
    {
        X=p.getX()+X;
        Y=p.getY()+Y;
    }
    void copy(PointXY p)
    {
        X=p.getX();
        Y=p.getY();
    }
    PointXY shift(int s,boolean x, boolean y)

```

		<pre> { PointXY p=new PointXY((this.X),(this.Y)); if (x) p.setX(p.getX()+s); if(y) p.setY(p.getY()+s); return p; } </pre>
Shift P1		<pre> line.p1=line.p1.shift(Integer.parseInt(jTextField5.getText()),jCheckBox1.isSelected(),jCheckBox2.isSelected()); jTextArea1.append("P1 : X = " + line.getP1().getX() + "\tY = " + line.getP1().getY() + "\n"); jTextField1.setText(Integer.toString(line.getP1().getX())); jTextField2.setText(Integer.toString(line.getP1().getY())); </pre>
Shift P2		<pre> line.p2=line.p2.shift(Integer.parseInt(jTextField5.getText()),jCheckBox1.isSelected(),jCheckBox2.isSelected()); jTextArea1.append("P2 : X = " + line.getP2().getX() + "\tY = " + line.getP2().getY() + "\n"); jTextField3.setText(Integer.toString(line.getP2().getX())); jTextField4.setText(Integer.toString(line.getP2().getY())); </pre>

21



PointXY

```
public class PointXY {
    public String coordinateType="cartesian ";
    int X;
    int Y;
    PointXY(int x, int y)
    {
        X=x;
        Y=y;
    }
    public void initPointXY()
    {
        this.X=0;
        this.Y=0;
    }
    int getX()
    {
```

```

        return X;
    }
    int getY()
    {
        return Y;
    }
    String getCoordinateType()
    {
        return coordinateType;
    }
    void setX(int X)
    {
        this.X=X;
    }
    void setY(int Y)
    {
        this.Y=Y;
    }
    void setCoordinateType(String st)
    {
        coordinateType=st;
    }
    void add(PointXY p)
    {
        X=p.getX()+X;
        Y=p.getY()+Y;
    }
    void copy(PointXY p)
    {
        X=p.getX();
        Y=p.getY();
    }
    PointXY shift(int s,boolean x, boolean y)
    {
        PointXY p=new PointXY((this.X),(this.Y));
        if (x) p.setX(p.getX()+s);
    }

```

	<pre> if(y) p.setY(p.getY()+s); return p; } } </pre>
jComboBox1ActionPerformed	<pre> DefaultListModel dlm= new DefaultListModel(); DefaultListModel edlm= new DefaultListModel(); jList2.setModel(edlm); jList3.setModel(edlm); jTextField1.setText("0"); jTextField2.setText("0"); switch (jComboBox1.getSelectedItem().toString()) { case "Point": dlm.addElement("Point 1"); break; case "Line": dlm.addElement("Point 1"); dlm.addElement("Point 2"); break; case "Triangle": dlm.addElement("Point 1"); dlm.addElement("Point 2"); dlm.addElement("Point 3"); break; case "Rectangle": dlm.addElement("Point 1"); dlm.addElement("Point 2"); break; case "Polygon": dlm.addElement("Point 1"); dlm.addElement("Point 2"); dlm.addElement("Point 3"); dlm.addElement("Point 4"); break; case "Circle": dlm.addElement("Center"); dlm.addElement("Radius"); </pre>

	break; } jList1.setModel(dlm);
jList2MouseClicked	jList3.setSelectedIndex(jList2.getSelectedIndex());
jList3MouseClicked	jList2.setSelectedIndex(jList3.getSelectedIndex());
add	<pre> if (jList1.getModel().getSize() == jList2.getModel().getSize()) { JOptionPane.showMessageDialog(this, "End of list", "Error", JOptionPane.ERROR_MESSAGE); } else { DefaultListModel dlm1 = new DefaultListModel(); DefaultListModel dlm2 = new DefaultListModel(); for (int n = 0; n < jList2.getModel().getSize(); n++) { dlm1.addElement(jList2.getModel().getElementAt(n)); } dlm1.addElement(jTextField1.getText()); jList2.setModel(dlm1); for (int n = 0; n < jList3.getModel().getSize(); n++) { dlm2.addElement(jList3.getModel().getElementAt(n)); } dlm2.addElement(jTextField2.getText()); jList3.setModel(dlm2); } for (int n = 0; n < jList2.getModel().getSize(); n++) { switch (n) { case 1: P1.setX(Integer.parseInt(jList2.getModel().getElementAt(n).toString())); P1.setY(Integer.parseInt(jList3.getModel().getElementAt(n).toString())); </pre>

	<pre> ng())); break; case 2: P2.setX(Integer.parseInt(jList2.getModel().getElementAt(n).toString())); P2.setY(Integer.parseInt(jList3.getModel().getElementAt(n).toString())); break; case 3: P3.setX(Integer.parseInt(jList2.getModel().getElementAt(n).toString())); P3.setY(Integer.parseInt(jList3.getModel().getElementAt(n).toString())); break; case 4: P4.setX(Integer.parseInt(jList2.getModel().getElementAt(n).toString())); P4.setY(Integer.parseInt(jList3.getModel().getElementAt(n).toString())); break; default: break; } } </pre>
remove	<pre> DefaultListModel dlm1 = new DefaultListModel(); DefaultListModel dlm2 = new DefaultListModel(); int sel = jList2.getSelectedIndex(); for (int n = 0; n < jList2.getModel().getSize(); n++) { dlm1.addElement(jList2.getModel().getElementAt(n)); } </pre>

	<pre> d1m1.removeElementAt(sel); jList2.setModel(d1m1); for (int n = 0; n < jList3.getModel().getSize(); n++) { d1m2.addElement(jList3.getModel().getElementAt(n)); } d1m2.removeElementAt(sel); jList3.setModel(d1m2); </pre>
draw	<pre> if (jComboBox1.getSelectedIndex() == 3) { jPanel2.getGraphics().fillOval(P1.getX(), P1.getY(), 5, 5); jPanel2.getGraphics().fillOval(P2.getX(), P2.getY(), 5, 5); jPanel2.getGraphics().fillOval(P1.getX(), P2.getY(), 5, 5); jPanel2.getGraphics().fillOval(P2.getX(), P1.getY(), 5, 5); jPanel2.getGraphics().drawRect(P1.getX(), P1.getY(), P2.getX(), P2.getY()); } else if (jComboBox1.getSelectedIndex() == 5) { jPanel2.getGraphics().fillOval(P1.getX(), P1.getY(), 5, 5); jPanel2.getGraphics().drawOval(P1.getX(), P1.getY(), P2.getX(), P2.getX()); } else { switch (jList2.getModel().getSize()) { case 1: jPanel2.getGraphics().fillOval(P1.getX(), P1.getY(), 5, 5); break; case 2: jPanel2.getGraphics().fillOval(P1.getX(), P1.getY(), 5, 5); jPanel2.getGraphics().fillOval(P2.getX(), P2.getY(), 5, 5); jPanel2.getGraphics().drawLine(P1.getX(), P1.getY(), P2.getX(), P2.getY()); break; case 3: jPanel2.getGraphics().fillOval(P1.getX(), P1.getY(), 5, </pre>

```

5);
        jPanel2.getGraphics().fillOval(P2.getX(), P2.getY(), 5,
5);
        jPanel2.getGraphics().fillOval(P3.getX(), P3.getY(), 5,
5);
        jPanel2.getGraphics().drawLine(P1.getX(), P1.getY(),
P2.getX(), P2.getY());
        jPanel2.getGraphics().drawLine(P2.getX(), P2.getY(),
P3.getX(), P3.getY());
        jPanel2.getGraphics().drawLine(P3.getX(), P3.getY(),
P1.getX(), P1.getY());
        break;
    case 4:
        jPanel2.getGraphics().fillOval(P1.getX(), P1.getY(), 5,
5);
        jPanel2.getGraphics().fillOval(P2.getX(), P2.getY(), 5,
5);
        jPanel2.getGraphics().fillOval(P3.getX(), P3.getY(), 5,
5);
        jPanel2.getGraphics().fillOval(P4.getX(), P4.getY(), 5,
5);
        jPanel2.getGraphics().drawLine(P1.getX(), P1.getY(),
P2.getX(), P2.getY());
        jPanel2.getGraphics().drawLine(P2.getX(), P2.getY(),
P3.getX(), P3.getY());
        jPanel2.getGraphics().drawLine(P3.getX(), P3.getY(),
P4.getX(), P4.getY());
        jPanel2.getGraphics().drawLine(P4.getX(), P4.getY(),
P1.getX(), P1.getY());
        break;
    }
}

```