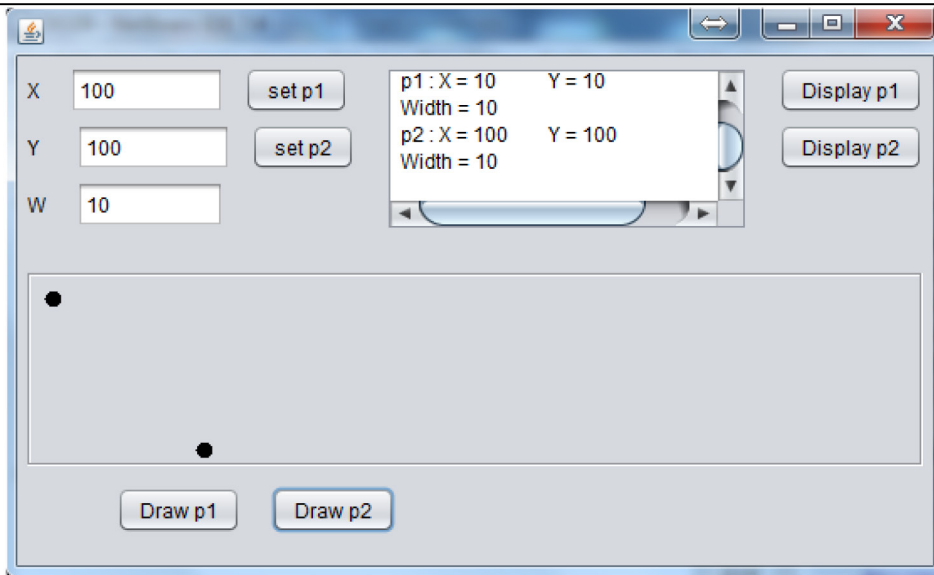


Fundamentals of Programming II Lab 10

0029



```

public class PointXYW extends PointXY{
    int w=0;
    PointXYW(int x,int y,int w)
    {
        super(x,y);
        this.w=w;
    }
    void setW(int w)
    {
        this.w=w;
    }
    int getW()
    {
        return w;
    }
}
    
```

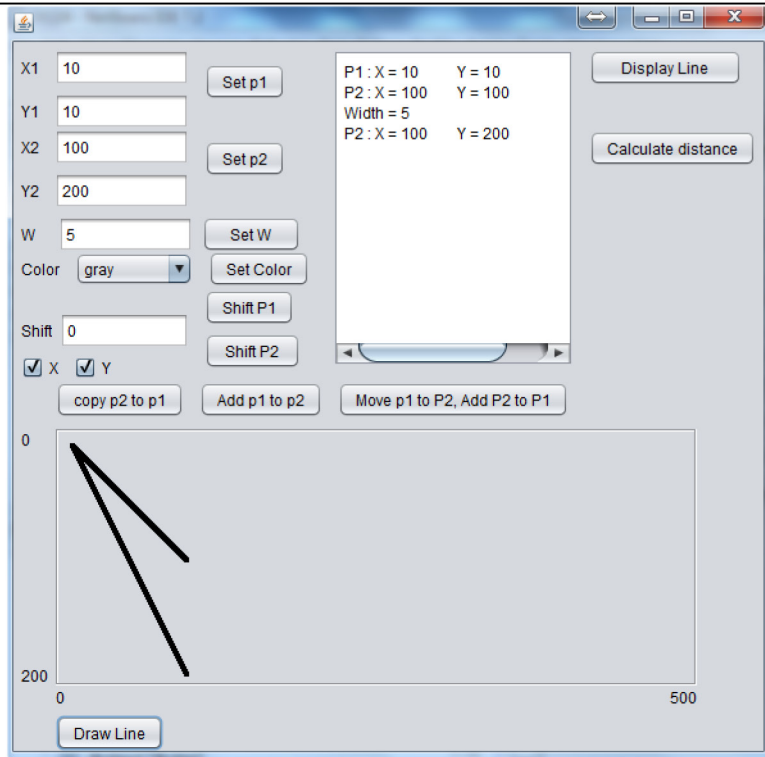
Set P1 `p1.setW(Integer.parseInt(jTextField4.getText()));`

Set P2 `p2.setW(Integer.parseInt(jTextField4.getText()));`

Draw P1 `jPanel1.getGraphics().fillOval(p1.getX(), p1.getY(), p1.getW(), p1.getW());`

Draw P2	jPanel1.getGraphics().fillOval(p2.getX(), p2.getY(), p2.getWidth(), p2.getHeight());
---------	--

0030

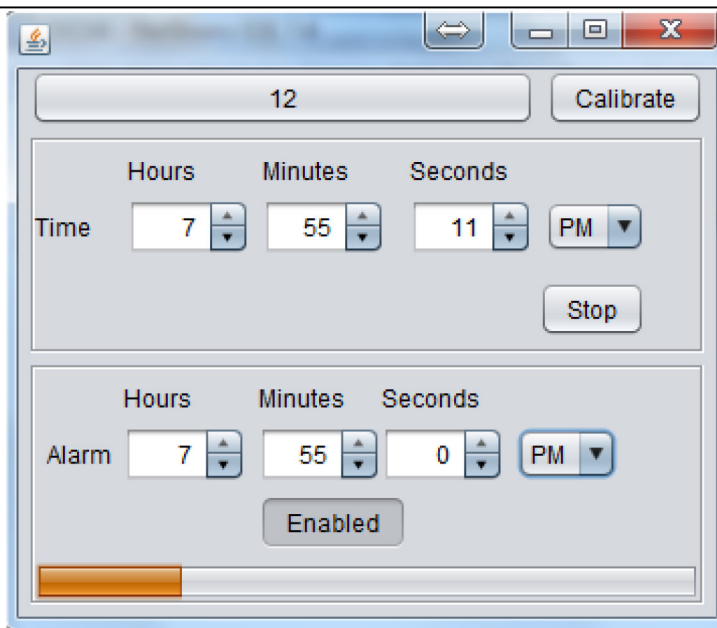


```

public class LineXYW extends LineXY{
    private int W;
    LineXYW()
    {
        super();
        W=0;
    }
    LineXYW(PointXY P1,PointXY P2, Color c, int w)
    {
        this.SetP1(P1);
        this.SetP2(P2);
        this.setColor(c);
        W=w;
    }
    void setW(int w)
    {
        W=w;
    }
}
    
```

	<pre> int getW() { return W; } </pre>
Set W	<pre> line.setW(Integer.parseInt(jTextField6.getText())); jTextArea1.append("Width = " + line.getW() + "\n"); </pre>
Display line	<pre> jTextArea1.append("Width = " + line.getW() + "\n"); </pre>
Draw line	<pre> //jPanel1.getGraphics().drawLine(line.getP1().getX(), line.getP1().getY(), line.getP2().getX(), line.getP2().getY()); float slope=(line.getP2().getY()- line.getP1().getY())/((line.getP2().getX()-line.getP1().getX())); int intercept=(int)(line.getP1().getY()- (slope*line.getP1().getX())); for(int n=0;n<((line.getP2().getX()-line.getP1().getX()));n++) { jPanel1.getGraphics().fillOval(line.getP1().getX()+n, (int)((line.getP1().getX()+n)*slope)+intercept, line.getW(), line.getW()); } </pre>

0034



CTime	<pre> CTime12 convertToCtime12() { int h; boolean am = true; if (this.h > 12) { h = this.h - 12; am = false; } else { h = this.h; } return new CTime12(h, this.m, this.s, am); } </pre>
CTime12	<pre> CTime convertToCTime24() { int h; if(!this.AM) h=this.h+12; else h=this.h; return new CTime(h,this.m,this.s); } </pre>
Form attribute s	<pre> CTime12 ct12 = new CTime12(); CTime12 alarm12 = new CTime12(); CTime ct24 = new CTime(); CTime alarm24 = new CTime(); </pre>
Calibrate ()	<pre> void calibrate() { Calendar c = Calendar.getInstance(); if (c.getTime().getHours() > 13) { ct12.setHours(c.getTime().getHours() - 12); ct12.setAM(false); } else { ct12.setHours(c.getTime().getHours()); ct12.setAM(true); } } </pre>

	<pre> ct24.setHours(c.getTime().getHours()); ct12.setMinutes(c.getTime().getMinutes()); ct24.setMinutes(c.getTime().getMinutes()); ct12.setSeconds(c.getTime().getSeconds()); ct24.setSeconds(c.getTime().getSeconds()); } </pre>
Form construc tor	<pre> calibrate(); t = new Timer(1000, new ActionListener() { @Override public void actionPerformed(ActionEvent e) { if (!jToggleButton2.isSelected()) { ct12.tick(); jSpinner1.setValue(ct12.getSeconds()); jSpinner2.setValue(ct12.getMinutes()); jSpinner3.setValue(ct12.getHours()); int i = 0; if (ct12.getAM()) { i = 0; } else { i = 1; } jComboBox1.setSelectedIndex(i); if ((jToggleButton1.isSelected()) && (ct12.compareToM(alarm12))) { Toolkit.getDefaultToolkit().beep(); jProgressBar1.setValue((jProgressBar1.getValue() + 1) % 10); } else { jProgressBar1.setValue(0); } } else { ct24.tick(); jSpinner1.setValue(ct24.getSeconds()); jSpinner2.setValue(ct24.getMinutes()); jSpinner3.setValue(ct24.getHours()); </pre>

	<pre> if ((jToggleButton1.isSelected()) && (ct24.compareToM(alarm24))) { Toolkit.getDefaultToolkit().beep(); jProgressBar1.setValue((jProgressBar1.getValue() + 1) % 10); } else { jProgressBar1.setValue(0); } } } }); t.start(); </pre>
jSpinner 1StateCh anged	<pre> jSpinner1.setValue(Integer.parseInt(jSpinner1.getValue().toString()) % 60); if (!jToggleButton2.isSelected()) { ct12.setSeconds(Integer.parseInt(jSpinner1.getValue().toString())); } else { ct24.setSeconds(Integer.parseInt(jSpinner1.getValue().toString())); } </pre>
jSpinner 2StateCh anged	<pre> jSpinner2.setValue(Integer.parseInt(jSpinner2.getValue().toString()) % 60); if (!jToggleButton2.isSelected()) { ct12.setMinutes(Integer.parseInt(jSpinner2.getValue().toString())); } else { ct24.setMinutes(Integer.parseInt(jSpinner2.getValue().toString())); } </pre>
jSpinner 3StateCh anged	<pre> if (!jToggleButton2.isSelected()) { jSpinner3.setValue(Integer.parseInt(jSpinner3.getValue().toString()) % 12); </pre>

	<pre> ct12.setHours(Integer.parseInt(jSpinner3.getValue().toString())); } else { jSpinner3.setValue(Integer.parseInt(jSpinner3.getValue().toString()) % 24); ct24.setHours(Integer.parseInt(jSpinner3.getValue().toString())); } </pre>
jSpinner 4StateCh anged	<pre> jSpinner4.setValue(Integer.parseInt(jSpinner4.getValue().toString()) % 60); if (!jToggleButton2.isSelected()) { alarm12.setSeconds(Integer.parseInt(jSpinner4.getValue().toStrin g()) % 60); } else { alarm24.setSeconds(Integer.parseInt(jSpinner4.getValue().toStrin g()) % 60); } </pre>
jSpinner 5StateCh anged	<pre> jSpinner5.setValue(Integer.parseInt(jSpinner5.getValue().toString()) % 60); if (!jToggleButton2.isSelected()) { alarm12.setMinutes(Integer.parseInt(jSpinner5.getValue().toStrin g()) % 60); } else { alarm24.setMinutes(Integer.parseInt(jSpinner5.getValue().toStrin g()) % 60); } </pre>
jSpinner 6StateCh anged	<pre> if (!jToggleButton2.isSelected()) { jSpinner6.setValue(Integer.parseInt(jSpinner6.getValue().toString()) % 12); alarm12.setHours(Integer.parseInt(jSpinner6.getValue().toString(</pre>

	<pre>) % 12); } else { jSpinner6.setValue(Integer.parseInt(jSpinner6.getValue().toString()) % 24); alarm24.setHours(Integer.parseInt(jSpinner6.getValue().toString() % 24); } </pre>
jComBo ox1Actio nPerfor med	<pre> if (!jToggleButton2.isSelected()) { if (jComboBox1.getSelectedIndex() == 0) { ct12.setAM(true); } else { ct12.setAM(false); } } </pre>
jComBo ox2Actio nPerfor med	<pre> if (!jToggleButton2.isSelected()) { if (jComboBox2.getSelectedIndex() == 0) { alarm12.setAM(true); } else { alarm12.setAM(false); } } </pre>
jToggleB utton2Ac tionPerf ormed	<pre> if (jToggleButton2.isSelected()) { jToggleButton2.setText("24"); ct24 = ct12.convertToCTime24(); alarm24 = alarm12.convertToCTime24(); jComboBox1.setSelectedIndex(-1); jComboBox2.setSelectedIndex(-1); jComboBox1.setEnabled(false); jComboBox2.setEnabled(false); jSpinner6.setValue(alarm24.getHours()); jSpinner3.setValue(ct24.getHours()); } else { jToggleButton2.setText("12"); ct12 = ct24.convertToCtime12(); alarm12 = alarm24.convertToCtime12(); } </pre>

		<pre> jComboBox1.setEnabled(true); jComboBox2.setEnabled(true); jSpinner6.setValue(alarm12.getHours()); jSpinner3.setValue(ct12.getHours()); if (alarm12.getAM()) { jComboBox2.setSelectedIndex(0); } else { jComboBox2.setSelectedIndex(1); } </pre>
	jButton2 ActionPerformed	calibrate();