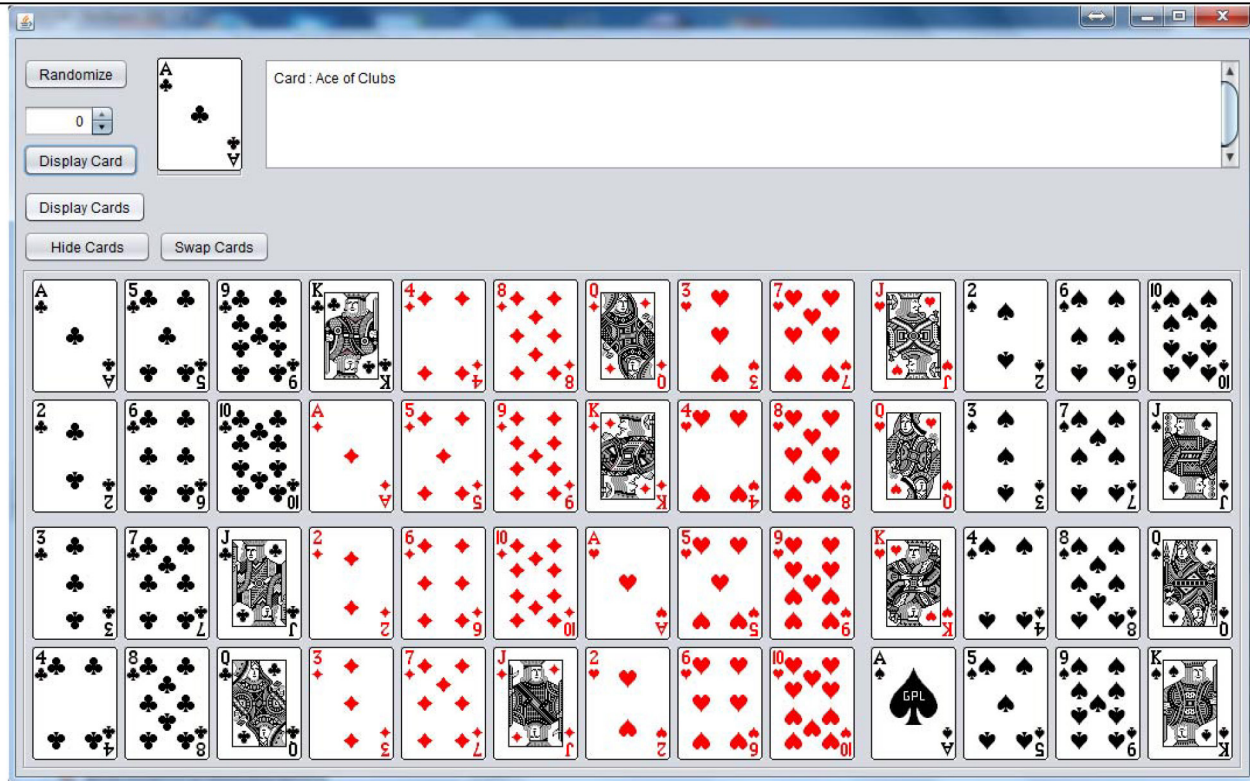


Fundamentals of Programming II Lab 09

0024



```

public class Card {

    int suit, rank;
    String[] suits = { "Clubs", "Diamonds", "Hearts", "Spades" };
    String[] ranks = { "Ace", "2", "3", "4", "5", "6",
        "7", "8", "9", "10", "Jack", "Queen", "King" };

    Card() {
        this.suit = 0;
        this.rank = 0;
    }

    Card(int suit, int rank) {
        this.suit = suit;
        this.rank = rank;
    }

    void setCard(int suit, int rank)
    {
        this.suit = suit;
        this.rank = rank;
    }
}
    
```

	<pre> } String getCard() { return ranks[rank] + " of " + suits[suit]; } String getCardFileName() { String s=""; s = String.format("%02d%c.gif", rank+1,suits[suit].charAt(0)); return s; } Card copyCard() { return new Card(this.suit,this.rank); } boolean equalRank(Card c) { if(this.rank==c.rank) return true; else return false; } } </pre>
	<pre> Card[] cards = new Card[52]; boolean swap = true; </pre>
Form constructor	<pre> int n = 0; for (int suit = 0; suit < 4; suit++) { for (int rank = 0; rank < 13; rank++) { cards[n] = new Card(suit, rank); n++; } } </pre>

Randomize	<pre> int n = 0; for (int suit = 0; suit < 4; suit++) { for (int rank = 0; rank < 13; rank++) { cards[n] = new Card(suit, rank); n++; } } for (n = 0; n < 100; n++) { int r1 = ((int) (Math.random() * 100)) % 52; int r2 = ((int) (Math.random() * 100)) % 52; //jTextArea1.append("Swap : " + r1 + "\twith : " + r2 + "\n"); Card c = new Card(); c = cards[r1].copyCard(); cards[r1] = cards[r2].copyCard(); cards[r2] = c.copyCard(); } </pre>
Display card	<pre> jTextArea1.append("Card : " + cards[Integer.parseInt(jSpinner1.getValue().toString())].getCard() + "\n"); Image image; image = new ImageIcon("Cards//" + cards[Integer.parseInt(jSpinner1.getValue().toString())].getCardFileName()).getImage(); jPanel2.getGraphics().drawImage(image, 0, 0, null); </pre>
Display	<pre> swap=true; for (int n = 0; n < 52; n++) { renderCard(n); } </pre>

renderCard()	<pre> if (cards[n] == null) { return; } Image image; if (swap) { image = new ImageIcon("Cards//" + cards[n].getCardFileName()).getImage(); } else { image = new ImageIcon("Cards//back001.gif").getImage(); } switch (n) { case 0: B1.getGraphics().drawImage(image, 0, 0, null); break; case 1: B2.getGraphics().drawImage(image, 0, 0, null); break; case 2: B3.getGraphics().drawImage(image, 0, 0, null); break; case 3: B4.getGraphics().drawImage(image, 0, 0, null); break; case 4: B5.getGraphics().drawImage(image, 0, 0, null); break; } } </pre>
HideCards	<pre> B1.repaint(); B2.repaint(); B3.repaint(); B4.repaint(); </pre>

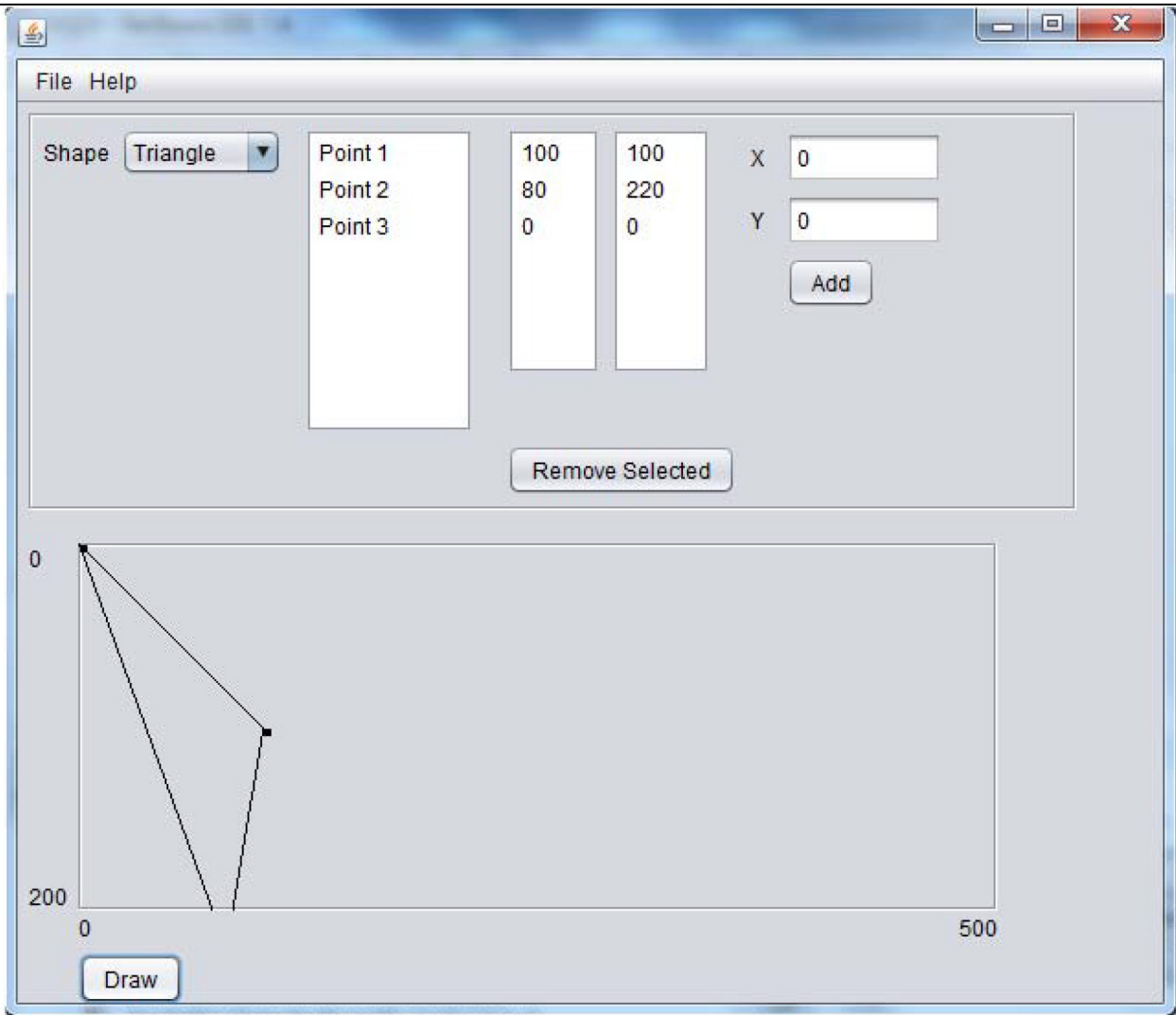
SwapCards

```

swap = !swap;
    for (int n = 0; n < 52; n++) {

        renderCard(n);
    }
    
```

0027



```

Open if (jFileChooser1.showOpenDialog(this) == 0) {
    try {
        FileInputStream fis = new
        FileInputStream(jFileChooser1.getSelectedFile().toString());
        ObjectInputStream ois = new ObjectInputStream(fis);
        jComboBox1.setSelectedIndex(ois.readInt());

        P1 = (PointXY) ois.readObject();
        P2 = (PointXY) ois.readObject();
    }
}
    
```

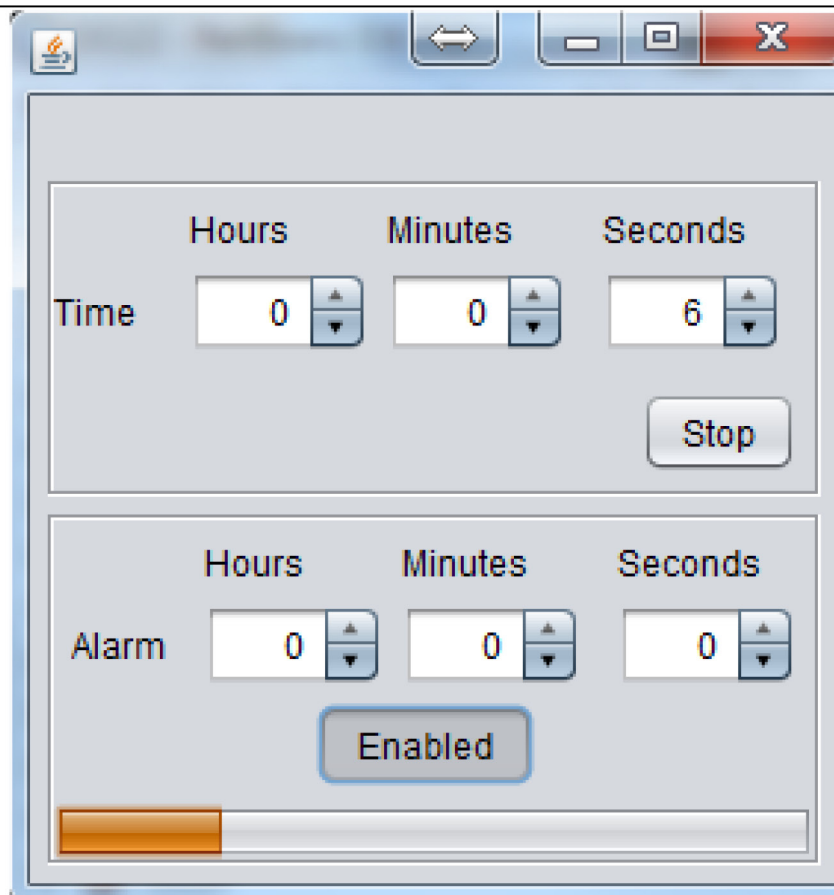
		<pre> P3 = (PointXY) ois.readObject(); P4 = (PointXY) ois.readObject(); ois.close(); fis.close(); DefaultListModel dlm1 = new DefaultListModel(); DefaultListModel dlm2 = new DefaultListModel(); for (int n = 0; n < jList1.getModel().getSize(); n++) { if (n == 0) { dlm1.addElement(P1.getX()); dlm2.addElement(P1.getY()); } else if (n == 1) { dlm1.addElement(P2.getX()); dlm2.addElement(P2.getY()); } else if (n == 2) { dlm1.addElement(P3.getX()); dlm2.addElement(P3.getY()); } else if (n == 3) { dlm1.addElement(P4.getX()); dlm2.addElement(P4.getY()); } jList2.setModel(dlm1); jList3.setModel(dlm2); } } catch (Exception e) { JOptionPane.showMessageDialog(this, e.getMessage(), "About", JOptionPane.ERROR_MESSAGE); } } </pre>
Save		<pre> if (jFileChooser1.showSaveDialog(this) == 0) { try { FileOutputStream fos = new FileOutputStream(jFileChooser1.getSelectedFile().toString()); ObjectOutputStream oos = new ObjectOutputStream(fos); oos.writeInt(jComboBox1.getSelectedIndex()); } } </pre>

```

        oos.writeObject(P1);
        oos.writeObject(P2);
        oos.writeObject(P3);
        oos.writeObject(P4);
        oos.close();
        fos.close();

    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage(),
        "About", JOptionPane.ERROR_MESSAGE);
    }
}
    
```

0028



```

CTime
e
    public class CTime {
        private int s,m,h;
        CTime()
        {
    
```

```

s=0;
m=0;
h=0;
}
CTime(int h, int m, int s)
{
    this.h=h%24;
    this.m=m%60;
    this.s=s%60;
}
void tick()
{
    s++;
    m=m+(s/60);
    s=s%60;
    h=h+(m/60);
    m=m%60;
    h=h%24;
}
int getMinutes()
{
    return m;
}
int getSeconds()
{
    return s;
}
int getHours()
{
    return h;
}
void setHours(int h)
{
    this.h=h;
}
void setMinutes(int m)

```

		<pre> { this.m=m; } void setSeconds(int s) { this.s=s; } long convert2Minutes() { long M; M=(h*60*60)+(m); return M; } Boolean compareToM(CTime ct) { long M1,M2; M1=ct.convert2Minutes(); M2=this.convert2Minutes(); if(M1==M2) return true; else return false; } } </pre>
	Form attribu	<pre> CTime ct = new CTime(); CTime alarm=new CTime(); Timer t; </pre>

Form Constructor	<pre> t = new Timer(1000, new ActionListener() { @Override public void actionPerformed(ActionEvent e) { ct.tick(); jSpinner1.setValue(ct.getSeconds()); jSpinner2.setValue(ct.getMinutes()); jSpinner3.setValue(ct.getHours()); if((jToggleButton1.isSelected())&&(ct.compareToM(alarm))) { Toolkit.getDefaultToolkit().beep(); jProgressBar1.setValue((jProgressBar1.getValue()+1)%10); } else jProgressBar1.setValue(0); } }); t.start(); </pre>
jSpinner1State changed	<pre> jSpinner1.setValue(Integer.parseInt(jSpinner1.getValue().toString()) %60); ct.setSeconds(Integer.parseInt(jSpinner1.getValue().toString())); </pre>
jSpinner2State changed	<pre> jSpinner2.setValue(Integer.parseInt(jSpinner2.getValue().toString()) %60); ct.setMinutes(Integer.parseInt(jSpinner2.getValue().toString())); </pre>
jSpinner3 StateChanged	<pre> jSpinner3.setValue(Integer.parseInt(jSpinner3.getValue().toString()) %24); ct.setHours(Integer.parseInt(jSpinner3.getValue().toString())); </pre>
jSpinner4StateCh anged	<pre> jSpinner4.setValue(Integer.parseInt(jSpinner4.getValue().toString()) %60); alarm.setSeconds(Integer.parseInt(jSpinner4.getValue().toString())% 60); </pre>

	jsSpinner5StateCh anged	<pre> jSpinner5.setValue(Integer.parseInt(jSpinner5.getValue().toString()) %60); alarm.setMinutes(Integer.parseInt(jSpinner5.getValue().toString())% 60); </pre>
	jsSpinner6StateCh anged	<pre> jSpinner6.setValue(Integer.parseInt(jSpinner6.getValue().toString()) %24); alarm.setHours(Integer.parseInt(jSpinner6.getValue().toString())%2 4); </pre>
	Stop	<pre> String st; if(jButton1.getText().equalsIgnoreCase("Stop")) { t.stop(); jButton1.setText("Start"); } else { t.start(); jButton1.setText("Stop"); } </pre>