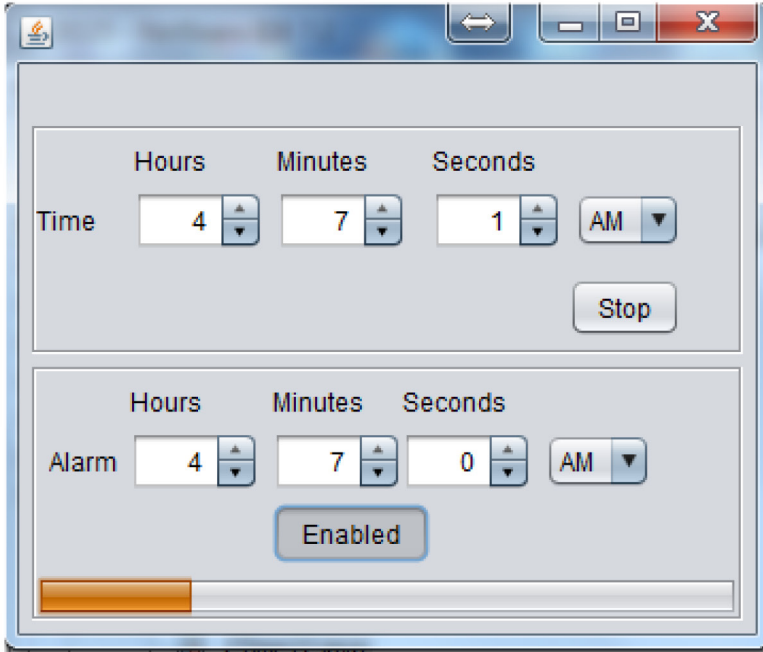


Fundamentals of Programming II Lab 10

0031



CTime12

```

private Boolean AM;
CTime12()
{
    super();
    AM=true;
}
CTime12(int h, int m, int s, boolean am)
{
    this.setHours(h%12);
    this.setMinutes(m%60);
    this.setSeconds(s%60);
    AM=am;
}
@Override
void tick()
{
    int s=this.getSeconds()+1;
    this.setSeconds(s%60);
    int m=this.getMinutes()+(s/60);
    this.setMinutes(m%60);
    int h=this.getHours()+(m/60);
    this.setHours(h%12);
    
```

	<pre> if((h/12)==1) AM=!AM; } @Override long convert2Seconds() { long S; S=(this.getHours()*60*60)+(this.getMinutes()*60)+this.getSeconds(); if(!AM)S=S+((12*60*60)+(59*60))+59; return S; } @Override long convert2Minutes() { long M; M=(this.getHours()*60)+(this.getMinutes()); if(!AM)M=M+(12*60)+59; return M; } void setAM(boolean am) { AM=am; } boolean getAM() { return AM; } </pre>
Form class declaration	<pre> CTime12 ct = new CTime12(); CTime12 alarm=new CTime12(); </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()); } }); </pre>

	<pre> jSpinner3.setValue(ct.getHours()); int i=0; if(ct.getAM())i=0; else i=1; jComboBox1.setSelectedIndex(i); if((jToggleButton1.isSelected())&&(ct.compareToM(alarm))) { Toolkit.getDefaultToolkit().beep(); jProgressBar1.setValue((jProgressBar1.getValue()+1) %10); } else jProgressBar1.setValue(0); } }); t.start(); </pre>
jSpinner3StateChanged	<pre> jSpinner3.setValue(Integer.parseInt(jSpinner3.getVal ue().toString())%12); ct.setHours(Integer.parseInt(jSpinner3.getValue().to String())); </pre>
jSpinner6StateChanged	<pre> jSpinner6.setValue(Integer.parseInt(jSpinner6.getVal ue().toString())%12); alarm.setHours(Integer.parseInt(jSpinner6.getValue().toString())%12); </pre>
jComboBox1ActionPerformed	<pre> if(jComboBox1.getSelectedIndex()==0) ct.setAM(true); else ct.setAM(false); </pre>
jComboBox2ActionPerformed	<pre> if(jComboBox2.getSelectedIndex()==0) alarm.setAM(true); else alarm.setAM(false); </pre>

0033



Card1

```

public class Card1 extends Card {

    Card1() {
        super();
    }

    Card1(int suit, int rank) {
        super(suit, rank);
    }

    @Override
    Card1 copyCard() {
        return new Card1(this.suit, this.rank);
    }

    boolean equalRank(Card c) {
        if (this.rank == c.rank) {
            return true;
        } else {
            return false;
        }
    }
}
  
```

Form attributes	Card1[] cards = new Card1[52];
checkWin	<pre> void checkWin() { for (int n = 0; n < 52; n++) { if (cards[n] == null) { continue; } for (int i = n + 1; i < 52; i++) { if (cards[i] == null) { continue; } if (face[n] && face[i]) { if (cards[n].equalRank(cards[i])) { cards[n] = null; cards[i] = null; face[n] = false; face[i] = false; clearCard(i); clearCard(n); jTextField1.setText("0"); jTextField2.setText(Integer.toString(((Integer.parseInt(jTextField1.getText()) - 2))); if((Integer.parseInt(jTextField2.getText()))==0) { JOptionPane.showMessageDialog(this, "You Win", "Congratulations",JOptionPane.INFORMATION_MESSAGE); } } } } } } </pre>

B1MouseClicked	<pre> int i = 0; if (face[i]) { face[i] = false; jTextField1.setText(Integer.toString(((Integer.parseInt(jTextField1.getText())) - 1))); } else { if (Integer.parseInt(jTextField1.getText()) < 2) { face[i] = true; jTextField1.setText(Integer.toString(((Integer.parseInt(jTextField1.getText())) + 1))); } } renderCard(i); checkWin(); </pre>
B2MouseClicked	<pre> int i = 1; if (face[i]) { face[i] = false; jTextField1.setText(Integer.toString(((Integer.parseInt(jTextField1.getText())) - 1))); } else { if (Integer.parseInt(jTextField1.getText()) < 2) { face[i] = true; jTextField1.setText(Integer.toString(((Integer.parseInt(jTextField1.getText())) + 1))); } } renderCard(i); checkWin(); </pre>