

Jukebox

Inhaltsverzeichnis

1	Java Jukebox	3
2	Code	4

1 Java Jukebox

Folgende Komponenten wurden erfolgreich eingebaut:

1. JFrame
2. JPanel
3. Thread
4. Thread.sleep()
5. Synthesizer synth=MidiSystem.getSynthesizer()
6. MidiChannel channels[]
7. synth.open()
8. channels = synth.getChannels()
9. channels[ch].programChange()
10. channels[ch].noteOff()
11. channels[ch].noteOn()
12. Exception
13. InterruptedException
14. paintTimer
15. paintComponent

2 Code

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.Random;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;
import java.util.Set;
import java.util.Iterator;
import java.net.*;
import javax.sound.midi.*;
public class jukebox{
    public static void main(String[] args)throws Exception{
        A a=new A();
        a.start();
    }
}
class B{
    public Synthesizer synth;
    public MidiChannel channels[];
    public int sleep=0;
    public Random rand;
    public int instrumentIndex =0;
    public B()throws Exception{
        rand=new Random();
        try{
            synth = MidiSystem.getSynthesizer();
            synth.open();
            channels = synth.getChannels();
            channels[1].noteOn(sleep, 0);
        }
        catch(Exception exc){}
    }
    public void setSound(int x,int ch){
        channels[ch].programChange(instrumentIndex);
        System.out.println(instrumentIndex+"");
        channels[ch].noteOff(sleep);
        channels[ch].noteOn(x,35);
        this.sleep=x;
    }
    public void setInIndex(int x){this.instrumentIndex=x;}
}
```

```

class A extends Thread{
public JFrame frame;
public JPanel p, ccp;
public drawPanel p2;
public A()throws Exception{
    p2=new drawPanel();
    ccp=new JPanel();
    frame=new JFrame();
    p=new drawPanel();
    frame.add(p);
    frame.setSize(200,150);
    frame.setVisible(true);
}
public void run(){
try{
    for(int i=0;i<1000000000;i++){
        for(int j=0;j<1000000000;j++){
            Thread.sleep(1);
p2.newTimer();
        }
    }
}
catch(InterruptedException e){}
}
}
class drawPanel extends JPanel{
public B b;
public Graphics g;
public Action paintTimer;
public Timer timer;
public Random rand;
public ArrayList<Integer> l1;
public int listIndex=0;
public drawPanel()throws Exception{
    rand=new Random();
    b=new B();
    paintTimer=new AbstractAction(){
        public void actionPerformed(ActionEvent e){repaint();}
    };
    timer=new Timer(500,paintTimer);
    timer.start();
    l1=new ArrayList<Integer>();
    int x=30;
    l1.add(x);
    l1.add(x+3);
    l1.add(x+2);
    l1.add(x+1);
    l1.add(x);
    l1.add(x+3);
    l1.add(x+2);
    l1.add(x+1);
    l1.add(x);
    l1.add(x+3);
    l1.add(x+2);
    l1.add(x+1);
    l1.add(x);
    l1.add(x+3);
    l1.add(x+2);
    l1.add(x+1);
}
//Instrument[] orchestra = synthesizer.getAvailableInstruments();

```

```

public void newTimer(){
    timer.stop();
    timer=new Timer(0,paintTimer);
    timer.start();

}
public int fourIndex=0;
public int inIndex=0;
@Override
public void paintComponent(Graphics g){
    super.paintComponent(g);
    int x=l1.get(listIndex);
    fourIndex++;
    if(fourIndex==3){listIndex++;fourIndex=0;}
    b.setInIndex(7);
    inIndex++;
    if(listIndex==16){listIndex=0;}
    b.setSound(x,1);
    b.setSound((x+4),2);
    b.setSound((x+7),3);
    b.setInIndex(1);
    b.setSound((x+15),4);
    b.setSound((x+19),6);
    b.setSound((x+22),7);
    b.setSound((rand.nextInt(100)),8);
    b.setSound((rand.nextInt(100)),9);
    b.setSound((rand.nextInt(100)),10);
    b.setSound((rand.nextInt(100)),11);
    g.drawLine(100,x,200,x);
}
}
}

```