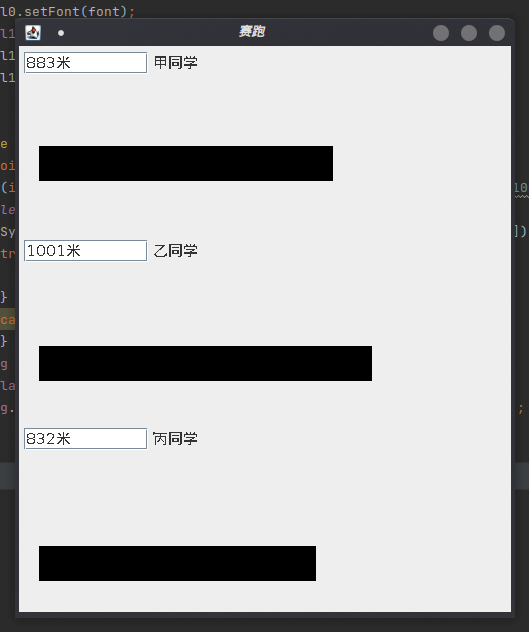
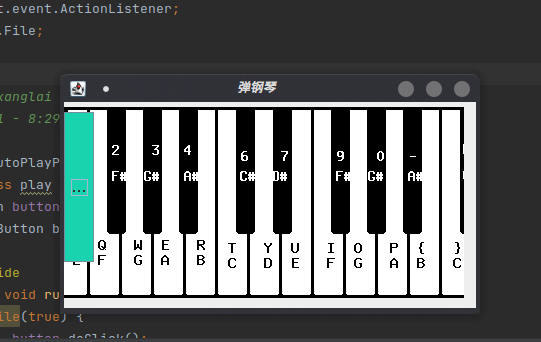
1.

import javax.swing.\*;  
import java.awt.\*;  
  
*/\*\**  
 *\** ***@author*** *liukanglai*  
 *\** ***@date*** *5/9/21 - 9:45 AM*  
 *\*/*  
public class RunGame {  
  
 static int[] *length* = new int[]{0, 0, 0};  
 static class SomeOneRun extends Thread {  
 String player;  
 int who = 0;  
 JPanel panel0;  
 JPanel panel1;  
 JTextField label1 = new JTextField(8);  
 Graphics g;  
  
 SomeOneRun(int who, String player, JPanel panel0, JPanel panel1) {  
 this.who = who;  
 this.player = player;  
 this.panel0 = panel0;  
 this.panel1 = panel1;  
 //g = panel0.getGraphics();  
 Font font = new Font(null, Font.*PLAIN*, 15);  
 JLabel label0 = new JLabel(player);  
 label0.setFont(font);  
 label1.setFont(font);  
 panel1.add(label1);  
 panel1.add(label0);  
 }  
  
 @Override  
 public void run() {  
 for (int i = 0; *length*[0] < 1000 && *length*[1] < 1000 && *length*[2] < 1000; i++) {  
 *length*[who] += (int) (Math.*random*() \* 21);  
 System.*out*.println(player + "run, the length1 is: " + *length*[who]);  
 try {  
 Thread.*sleep*(100);  
 }  
 catch (InterruptedException e){  
 }  
 g = panel0.getGraphics();  
 label1.setText(String.*valueOf*(*length*[who])+"米");  
 g.fillRect(20, 100 + who\*200, *length*[who]/3, 35);  
 }  
 }  
  
 }  
  
 public static void main(String[] args) {  
 String player0 = "甲同学";  
 String player1 = "乙同学";  
 String player2 = "ͬ丙同学";  
  
 JFrame frame = new JFrame("赛跑");  
 frame.setSize(500, 600);  
 frame.setLocationRelativeTo(null);  
 frame.setDefaultCloseOperation(3);  
  
 JPanel big\_panel = new JPanel(new GridLayout(3, 1));  
 JPanel in\_panel0 = new JPanel(new FlowLayout(FlowLayout.*LEFT*));  
 JPanel in\_panel1 = new JPanel(new FlowLayout(FlowLayout.*LEFT*));  
 JPanel in\_panel2 = new JPanel(new FlowLayout(FlowLayout.*LEFT*));  
 big\_panel.add(in\_panel0);  
 big\_panel.add(in\_panel1);  
 big\_panel.add(in\_panel2);  
 SomeOneRun run0 = new SomeOneRun(0, player0, big\_panel, in\_panel0);  
 SomeOneRun run1 = new SomeOneRun(1, player1, big\_panel, in\_panel1);  
 SomeOneRun run2 = new SomeOneRun(2, player2, big\_panel, in\_panel2);  
  
 run0.start();  
 run1.start();  
 run2.start();  
  
 frame.setContentPane(big\_panel);  
 frame.setVisible(true);  
 }  
}



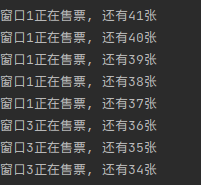
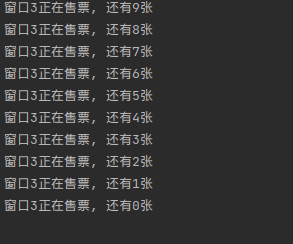
2.

import javax.swing.\*;  
import java.applet.Applet;  
import java.applet.AudioClip;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.File;  
  
*/\*\**  
 *\** ***@author*** *liukanglai*  
 *\** ***@date*** *5/9/21 - 8:29 PM*  
 *\*/*  
public class AutoPlayPiano {  
 static class play implements Runnable{  
 JButton button;  
 play(JButton button){  
 this.button = button;  
 }  
  
 @Override  
 public void run() {  
 while(true) {  
 button.doClick();  
 try {  
 Thread.*sleep*(100);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
 public static void main(String[] args) {  
 JFrame win = new JFrame("弹钢琴");  
 win.setSize(420, 240);  
 win.setLocationRelativeTo(null);  
 win.setDefaultCloseOperation(3);  
  
 JPanel panel = new JPanel(null);  
 ImageIcon image = new ImageIcon("/home/liukanglai/Downloads/钢琴素材/keyBoard.jpg");  
 JLabel piano = new JLabel(image);  
 piano.setBounds(0, -150, 400, 500);  
 panel.add(piano);  
 JButton[] buttons = new JButton[13];  
 play[] startPlay = new play[13];  
 for(int i = 0; i < 13; i++) {  
 buttons[i] = new JButton(String.*valueOf*((char)('a'+i)));  
 buttons[i].setBounds(30\*i, 10, 30, 150);  
 panel.add(buttons[i]);  
 startPlay[i] = new play(buttons[i]);  
 }  
 int[] flag = new int[]{0};  
 for(int i = 0; i < 13; i++) {  
 int color0 = (int)(Math.*random*()\*255);  
 int color1 = (int)(Math.*random*()\*255);  
 int color2 = (int)(Math.*random*()\*255);  
 flag[0]++;  
 File file = new File("/home/liukanglai/Downloads/钢琴素材/"+flag[0]+".wav");  
 buttons[i].setBackground(new Color(color0, color1, color2));  
 buttons[i].addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent actionEvent) {  
 try {  
 /\*  
 AudioInputStream audioInputStream = AudioSystem.getAudioInputStream(file.getAbsoluteFile());  
 Clip clip = AudioSystem.getClip();  
 clip.open(audioInputStream);  
 clip.start();  
 clip.close();  
  
 \*/  
 AudioClip audioClip = null;  
 audioClip = Applet.*newAudioClip*(file.toURL());  
 //循环播放 播放一次可以使用  
 audioClip.loop();  
 Thread.*sleep*(100);  
 } catch (Exception ex) {  
 System.*out*.println("Error with playing sound.");  
 ex.printStackTrace();  
 }  
 }  
 });  
 buttons[i].doClick();  
 }  
 /\*  
 Thread[] threads = new Thread[13];  
 for (int i = 0; i < 13; i++) {  
 threads[i] = new Thread(startPlay[i]);  
 threads[i].start();  
 }  
  
 \*/  
 win.setContentPane(panel);  
 win.setVisible(true);  
 }  
}



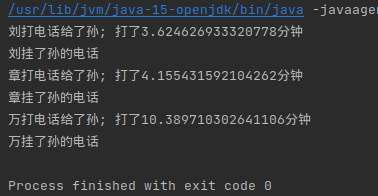
3.

public class SaleTickets {  
  
 public static class Sale implements Runnable {  
 static int *saleTickets* = 0;  
 String name;  
  
 Sale(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void run() {  
 while (true) {  
 synchronized (Sale.class) {  
 if(*saleTickets* < 100) {  
 *saleTickets*++;  
 System.*out*.println(name + "正在售票, 还有" + String.*valueOf*(100 - *saleTickets*) + "张");  
 }  
 else{  
 break;  
 }  
 try {  
 Thread.*sleep*(100);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
 }  
 public static void main(String[] args) {  
  
 Sale win1 = new Sale("窗口1");  
 Sale win2 = new Sale("窗口2");  
 Sale win3 = new Sale("窗口3");  
 Thread thread1 = new Thread(win1);  
 Thread thread2 = new Thread(win2);  
 Thread thread3 = new Thread(win3);  
 thread1.start();  
 thread2.start();  
 thread3.start();  
 }  
}



4.

public class Call {  
  
  
 static class CallSomeone extends Thread{  
 String[] callMan = new String[]{"章", "刘", "万"};  
 String beMan = new String("孙");  
 int who;  
 CallSomeone(int who) {  
 this.who = who;  
 }  
  
 @Override  
 public void run() {  
 synchronized (Lock.*lock*) {  
 String time = String.*valueOf*(Math.*random*() \* 20);  
 System.*out*.println(callMan[who] + "打电话给了" + beMan + "; 打了" + time + "分钟");  
 System.*out*.println(callMan[who] + "挂了" + beMan + "的电话");  
 }  
 }  
 }  
  
 public static void main(String[] args) {  
 CallSomeone call0 = new CallSomeone(0);  
 CallSomeone call1 = new CallSomeone(1);  
 CallSomeone call2 = new CallSomeone(2);  
  
 call0.start();  
 call1.start();  
 call2.start();  
 }  
}  
class Lock {  
 public static final Object *lock* = new Object();  
 public static Boolean *flag* = true;  
 //static int y = 150, x = 20;  
}



5.

import javax.swing.\*;  
import java.awt.\*;  
  
*/\*\**  
 *\** ***@author*** *liukanglai*  
 *\** ***@date*** *5/9/21 - 11:13 PM*  
 *\*/*  
public class MoveCar {  
 public static void main(String[] args) {  
 JFrame frame = new JFrame("Move car");  
 frame.setSize(500, 400);  
 frame.setLocationRelativeTo(null);  
 frame.setDefaultCloseOperation(3);  
 frame.setVisible(true);  
  
 Graphics g = frame.getGraphics();  
 Move move = new Move(g);  
 move.start();  
  
 }  
}  
  
class Move extends Thread {  
 Graphics g;  
  
 Move(Graphics g) {  
 this.g = g;  
 }  
  
 @Override  
 public void run() {  
 int y = 150, x = 20;  
 synchronized (this) {  
 while (y < 500) {  
 if (x < 200) {  
 g.drawRect(x, y, 50, 50);  
 g.drawArc(x + 5, y - 10, 10, 10, 0, 360);  
 g.drawArc(x + 35, y - 10, 10, 10, 0, 360);  
 g.drawArc(x + 5, y + 50, 10, 10, 0, 360);  
 g.drawArc(x + 35, y + 50, 10, 10, 0, 360);  
 try {  
 Thread.*sleep*(30);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 g.clearRect(x, y-10, 100, 100);  
 x += 1;  
 }  
 else{  
 g.drawRect(x, y, 50, 50);  
 g.drawArc(x + 5, y - 10, 10, 10, 0, 360);  
 g.drawArc(x + 35, y - 10, 10, 10, 0, 360);  
 g.drawArc(x + 5, y + 50, 10, 10, 0, 360);  
 g.drawArc(x + 35, y + 50, 10, 10, 0, 360);  
 try {  
 Thread.*sleep*(30);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 g.clearRect(x, y-10, 100, 100);  
 y += 1;  
 }  
 }  
 }  
 }  
}

