

《Java程序设计》上机实验报告

第 八 次上机

学号： 201611130126

姓名： 袁宇昊

学院： 信息科学学院

专业： 计算机科学

教师： 栾华

日期： 2019.5.21

1. 实验要求

1.上机之前应做好充分准备，认真思考所需的上机题目，提高上机效率。

2.独立上机输入和调试自己所编的程序，切忌抄袭、拷贝他人程序。

3.上机结束后，整理出实验报告。书写报告时，重点放在实验的方法、思路以及总结反思上，以达到巩固课堂学习、提高动手能力的目的。

4.红字标出部分是本节实验中必须提交截图或者说明的部分，请同学们认真解答。

1. 实验过程

（对于每个问题，可以包括:

1.实验步骤，如对某个问题的分析、算法思路；

2.实验过程，如关键代码。

3.运行结果，如相关问题运行结果的截图，结果说明等等。）

1.运行下面的程序，理解用创建 Thread 子类的方法实现多线程:

class SimpleThread extends Thread {

public SimpleThread(String str) {

super(str); }

public void run() {

for (int i = 0; i < 10; i++) {

System.out.println(i + " " + getName());

try {

sleep((int)(Math.random() \* 1000));

} catch (InterruptedException e) {}

}

System.out.println("DONE! " + getName());

}

}

public class TwoThreadsTest {

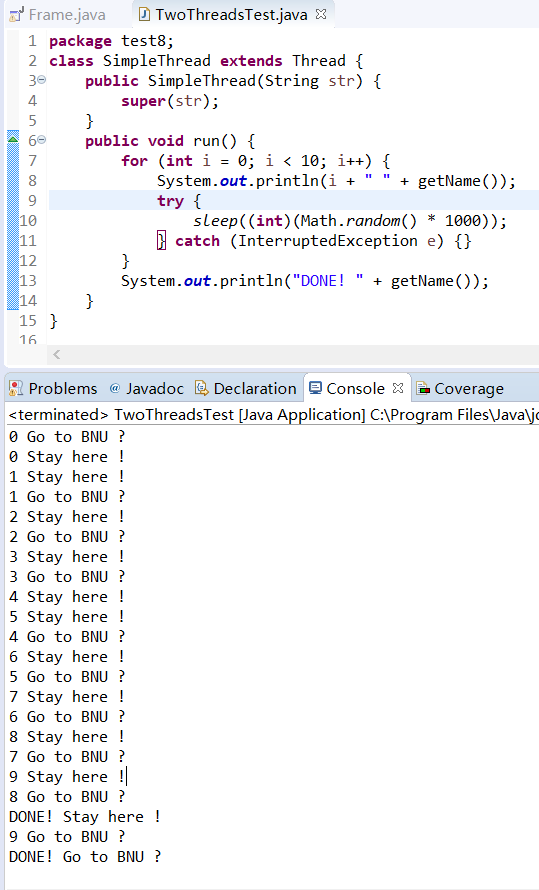
public static void main (String[] args) {

new SimpleThread("Go to BNU ?").start();

new SimpleThread("Stay here !").start();

}

}



2、运行下面的程序，理解用实现 Runnable 接口的方法实现多线程:

import java.awt.\*;

import java.util.\*;

import javax.swing.\*;

import java.text.\*;

public class ShowTimeSample extends JPanel implements Runnable {

Date timenow;

public void run(){

while(true) {

repaint();

try{

Thread.sleep(1000);

}

catch(InterruptedException e){

}

}

}

public void paint(Graphics g){

super.paint(g);

timenow = new Date();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy 年 MM 月 dd 日hh:mm:ssa”);

String s\_time\_msg = sdf.format(timenow);

g.drawString(s\_time\_msg, 50, 50);

}

public static void main(String[] args) {

JFrame jf = new JFrame();

jf.setDefaultCloseOperation(3);

jf.setTitle("时钟显示");

jf.setSize(300,300);

jf.setVisible(true);

ShowTimeSample sts = new ShowTimeSample();

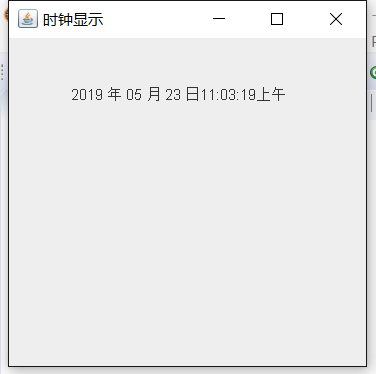
sts.setVisible(true);

sts.setBounds(0, 0, 300, 300); jf.add(sts);

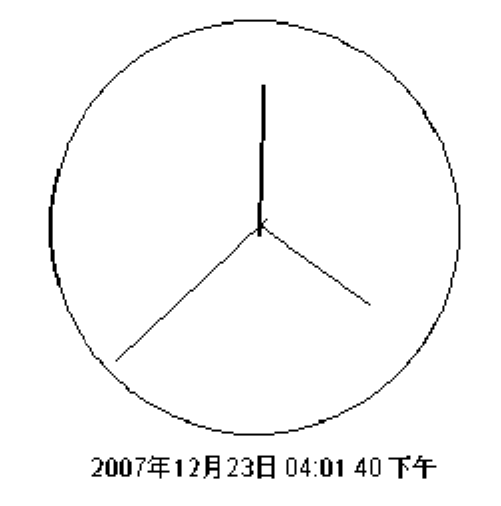
new Thread(sts).start();

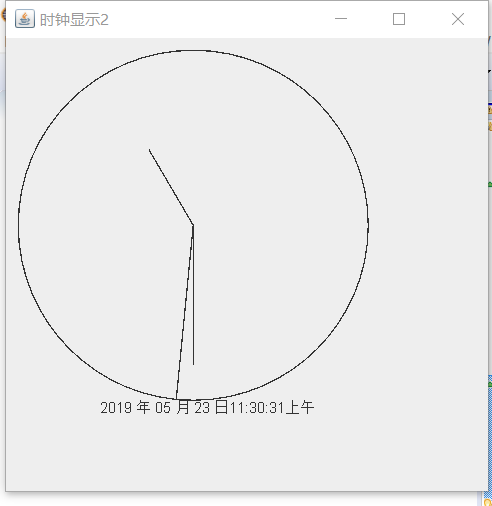
}

}



3、以图形方式模拟钟表显示时间，钟表有时针、分针、秒针，如下图所示。





4、准备一幅 GIF 格式的图像，以 Image.gif 为文件名保存，阅读并调试下述程序，观 察程序运行结果，描述程序实现的功能。

import java.awt.\*;

import javax.swing.\*;

public class WaveSample extends JPanel implements Runnable{

Image imgf,imgv;

Graphics gv;

int gw,gh,n,k,d[];

final int w = 4,b = 5;

Color c1;

public void init() {

gw = getSize().width;

gh = getSize().height;

n = gw/w;

k = ­­­n-1;

d = new int[n];

for(int i = 1;i < n;i++){

d[i] = (int)(Math.sin(Math.PI\*2/n\*i)\*b);

}

imgf = getToolkit().getImage(ClassLoader.getSystemResource("Image.gif"));

imgv = createImage(gw,gh\*2-b\*2);

gv = imgv.getGraphics();

c1 = new Color(0xe8, 0xff, 0xff);

gv.setColor(c1);

}

public void paint(Graphics g){

super.paint(g);

gv.fillRect(0,0,gw,gh);

gv.drawImage(imgf,0,gh,this);

for(int i = 0;i < n;i++){

gv.copyArea(i\*w,gh,w,gh-2\*b,0,-gh+b+d[(i+k)%n]);

}

g.drawImage(imgv,0,0,this); }

public void run(){

while (true){

try{

Thread.sleep(50);

k--;

if(k<0) k = n-1;

repaint();

}

catch (InterruptedException ie){

}

}

}

public static void main(String args[]) {

JFrame jf = new JFrame("动态图像");

WaveSample ws = new WaveSample();

jf.add("Center", ws);

jf.setSize(400,300);

jf.setVisible(true);

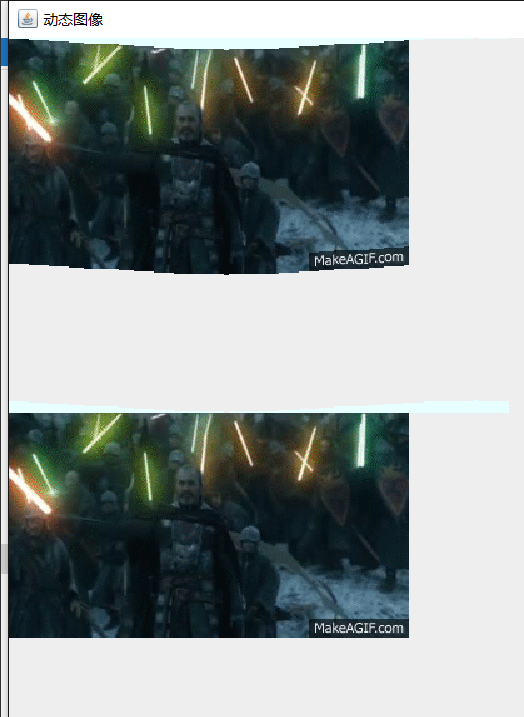
ws.setSize(400,300);

ws.init();

new Thread(ws).start();

}

}



5、设计文字(字符串)从右向左动态滚动显示的程序。



字符串从右下到左上滚动。

三.思考与分析

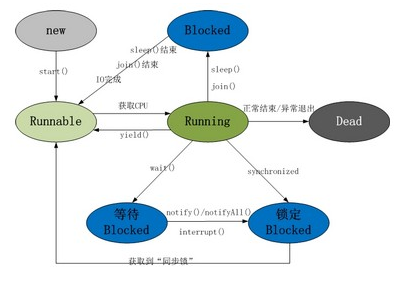
(1)简述进程和线程概念。

进程是具有独立功能的程序在某个数据集合上的一次执行过程。进程是系统进行资源分配和调度的一个独立单位。

线程是进程内的一个执行实体或执行单元，是比进程更小的能独立运行的基本单位

(2)线程有哪几个基本的状态?Java 中线程调度遵循何种原则?

**java中,每个线程都需经历新生、就绪、运行、阻塞和死亡五种状态。**



java中各状态转化如图所示。

(3)实现多线程可以用哪两种基本方法?将这两种方法进行比较。

继承Thread类是一种很常见的创建方式，它的具体做法是，该类继承Thread类并重写run（）方法。

实现Runnable接口，此方法实现多线程需要实现Runnable接口，然后重写run()方法。

两种实现方法的区别

相比第一种方法，第二种方法拥有一些优势，具体体现如下：

1）：适合多个相同的程序代码的线程去处理同一个资源

2）：可以避免java中的单继承的限制（如果该类要继承多个类，那么第一种方法就不适用了）

3）：增加程序的健壮性，代码可以被多个线程共享，代码和数据独立

4）：线程池只能放入实现Runable或callable类线程，不能直接放入继承Thread的类

四、总结（实验中遇到的问题、取得的经验、感想等）

多问黄文诚大佬

**提交说明：**  
提交一个rar 或zip压缩文件，其中包括：实验报告、源程序等，rar 或zip文件名为学生学号和姓名。

例如，20192255225\_姓名\_实验x

文件夹内包括所有源程序、工程文件等也用此命名。  
发送到：yueer0538@126.com