学习报告

计算机学院

1160301023 安宏展

由于我是计算机学院的学生，所以在上这门课之前就有一定的java的基础，另外我们本学期部分课程也涉及java编程，所以对于java还是有一定的了解的，本门课程的前面几个部分之前都有过接触，例如 基本类型与操作符、方法、类和对象、继承、多态一直到本门课程所涉及到的用户界面部分都有所了解，多线程接触的不多，但是接触过售卖汽车票问题，接触过线程和lock（锁）的概念，下面进行简要概述剩余部分的学习过程：

多线程：了解多线程的概念，我们就要先了解一下进程的概 念进而了解多线程的具体意义以及多线程的实际应 用，比如cpu的开发等等，实际上多线程在日程生 活中处处都在，我们每个人都不可能只做一件事情， 一定是同时完成多件事情，毫无疑问这边是多线程 的意义之所在

java小程序：java小程序是对于前面知识的综合，对java基 础知识的要求比较高，因为涉及方方面面的问 题需要解决，所以小程序无疑是对java能力的 训练，但是java小程序并不是主流

java网络编程：java网络编程与java web方面比较接近，我 对于java web还是了解一些的，网络编程其 实就是数据传输的问题，我们所谓的传输， 其实就是在传输数据，请求与回复 (request\response)。当然，协议也不只有TCP 和 UDP这两类，最起码我们还知道经常用到 的http协议，每种协议都存在自己特点与不 足之处，在选择传输连接的需要进行权衡

下面对于五个作业进行简单分析（作业的顺序可能与老师留的作业的顺序有些不同）

Question 1:

算法问题：按步长k输出数组中的元素，已输出的元素跳过 去，不要重复输出

控制台输入不必多说，主要讲解一下按照步长输出的方法

**int** k=0;//已经输出的个数

**int** now = 0;//当前所在位置

**while**(k<n)

{

**for**(**int** z=0;z<x-1;z++)

{

**while**(list.get(now).isVisited)

now = (now+1)%n;

now = (now+1)%n;

**while**(list.get(now).isVisited)

now = (now+1)%n;

}

k++;

System.out.print(list.get(now).number+" ");

list.get(now).isVisited = **true**;

}

首先k代表已经输出的元素的个数，当到达n时退出循环，z为步长，当输出步长为z时我们需要向后移动z-1“步”然后进行输出。这里的移动并不是普通意义上的移动，我们定义一个类c（名称的定义应该和具体应用关联，但是这次忘了）

**class** c

{

**int** number;

**boolean** isVisited;

**public** c(**int** number,**boolean** isVisited){

**this**.number = number;

**this**.isVisited = isVisited;

}

}

这个类里面有一个数字用来输出，另外一个isVisited代表是否该数字是否已经输出过，所以我们的移动一步便是指向后移动一个未输出过的数字，并不是单纯的移动，另外，当k到达n时，k应该变成0继续遍历

Question 2:

swing 编程，实现图片轮换

首先，需要显示图片，我的方法是使用一个Jlabel来显示图片，然后使用JPanel来承接Jlabel，再然后将JPanel add进入Jframe里面便可以显示，然后想JFrame里面加入两个Button用来作为 上一张\下一张 的按钮，使用BorderLayout的布局将panel放在中间，Button放在两边即可，对与button添加监听，点击时间发生是进行图片的轮换

String[] fileList = dirFile.list();

jButton1 = **new** JButton("上一张");

jButton2 = **new** JButton("下一张");

draw("images/"+fileList[number]);

jButton1.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

JFrame jFrame = **new** JFrame(fileList[number]);

jFrame1.dispose();

**if**(number > 1){

number--;

jLabel = **new** JLabel(**new** ImageIcon("images/"+fileList[number]));

}**else** {

jLabel = **new** JLabel(**new** ImageIcon("images/"+fileList[number]));

}

jLabel.setSize(d);

jFrame.setTitle(fileList[number]);

jPanel = **new** JPanel();

jPanel.add(jLabel);

jPanel.setSize(d.width,d.height);

jFrame.add(jPanel,BorderLayout.CENTER);

jFrame.add(jButton1,BorderLayout.WEST);

jFrame.add(jButton2,BorderLayout.EAST);

jFrame.setVisible(**true**);

jFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jFrame.setSize(d.width, d.height);

jFrame1 = jFrame;

}

});

jButton2.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

JFrame jFrame = **new** JFrame();

jFrame1.dispose();

**if**(number<fileList.length){

number++;

jLabel = **new** JLabel(**new** ImageIcon("images/"+fileList[number]));

}**else** {

jLabel = **new** JLabel(**new** ImageIcon("images/"+fileList[number]));

}

jLabel.setSize(d);

jFrame.setTitle(fileList[number]);

jPanel = **new** JPanel();

jPanel.add(jLabel);

jPanel.setSize(d.width,d.height);

jFrame.add(jPanel,BorderLayout.CENTER);

jFrame.add(jButton1,BorderLayout.WEST);

jFrame.add(jButton2,BorderLayout.EAST);

jFrame.setVisible(**true**);

jFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jFrame.setSize(d.width, d.height);

jFrame1 = jFrame;

}

});

然后就是文件的路径读取问题，我们将所有的图片都放在src同名的images文件夹里面，然后将images文件下面的所有目录（文件名）存放到一个字符串数组里面然后进行轮换即可

dirFile = **new** File("images");

//判断该文件或目录是否存在，不存在时在控制台输出提醒

**if** (!dirFile.exists()) {

System.out.println("do not exit");

**return** ;

}

//判断如果不是一个目录，就判断是不是一个文件时文件则输出文件路径

**if** (!dirFile.isDirectory()) {

**if** (dirFile.isFile()) {

**try** {

System.out.println(dirFile.getCanonicalFile());

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

}

**return** ;

}

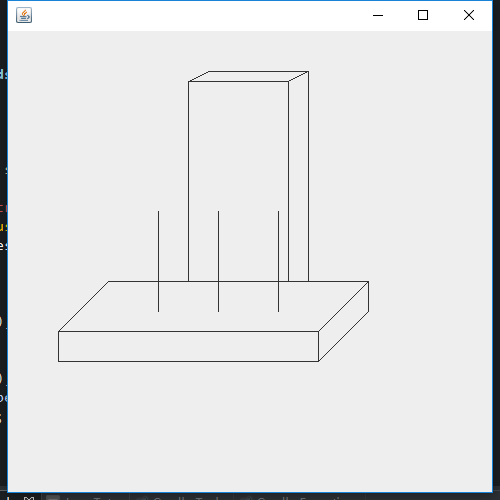
String[] fileList = dirFile.list();

Question03:

GUI 和清明节有关的图

利用Graphic进行绘图，绘制如下图所示的清明节有关图像

（貌似盗用了某位同学的创意），利用坐标变换画直线即可



Question04:

制作一个简单的聊天小程序

这个问题分为两个部分，一个是服务器（sever）部分，另一个是客户端（client）部分

首先，服务器部分，顾名思义，就是向外面提供服务的部分，我选择了tcp协议进行传输，利用了SeverSocket 类，首先 new 一个中间态的SeverSocket，这个服务器处于接收的状态，当它接收到来自客户端的请求时，则服务器变成接收的状态

**try**{

//服务器端启动

**while** (started){

Socket s = ss.accept();

Client c = **new** Client(s);

System.out.println("一个客户端连了上来");

**new** Thread(c).start();

//向数组中添加一个客户端

clients.add(c);

}

}

**catch** (IOException e) { //其他错误直接打印

e.printStackTrace();

}**finally** { //最后一定执行关闭资源

**try** {

ss.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

客户端部分 ：

建立一个与服务器端口号相同的客户端，然后new 一个新的对象即可与服务器进行链接，然后将内容放在输入输出流里面，

**public** **void** connect(){

**try** {

s = **new** Socket("127.0.0.1", 8888);

dos = **new** DataOutputStream(s.getOutputStream());

dis = **new** DataInputStream(s.getInputStream());

System.out.println("已连接！");

bConnected = **true**;

//启动对客户端的接收线程

// new Thread(new RecvThread()).start();

}**catch** (UnknownHostException e){

e.printStackTrace();

}**catch**(IOException e){

e.printStackTrace();

}

}

然后将所有的内容用GUI 的形式进行展示即可

Question 05:

多线程问题：一方面进行阶乘和的运算，另一方面每隔一段时间输出一次阶乘和的结果

阶乘和 1！+2！+3！+4！......

定义两个额外的类，其中一个类负责进行阶乘和的运算，另一个类用来输出阶乘和的结果

**class** ComputeThread **extends** Thread {// 计算阶乘

**double** sum = 0;// 存储阶乘和

**static** **int** i = 0;// 存储阶乘和进度

**static** String stringSum = "";// 存储阶乘和的字符串

**static** String stringResult = "";// 存储阶乘结果的字符串

**double** method(**int** n) {// 计算阶乘

**double** result = 1;// 阶乘结果

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

result \*= i;

}

**return** result;

}

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**while** (i < 30) {// 计算阶乘和

i++;

sum += method(i);

stringResult = String.valueOf(sum);// 将阶乘和存储到字符串中

**try** {// 0.5-1秒读取一次线程

Thread.sleep((**int**) (Math.random() \* 500 + 500));

} **catch** (InterruptedException ex) {

}

}

}

}

由于普通的sum变量没有办法在外部类里面访问（不是static类型）.而设置成static类型则会在计算阶乘和的时候出现错误，所以必须想另外一种方法，那就是将sum转换成字符串的形式

**static** String stringSum = "";

stringResult = String.valueOf(sum);

这样子就可以在另外一个类里面直接访问ComputeThread .stringResult 了

**class** ReadThread **extends** Thread {// 读取阶乘

**public** ReadThread() {// 初始化

}

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**while** (**true**) {

System.out.println(ComputeThread.stringResult);

**try** {

Thread.sleep(1000);// 1.0秒读取一次线程

} **catch** (Exception e) {

// **TODO**: handle exception

}

}

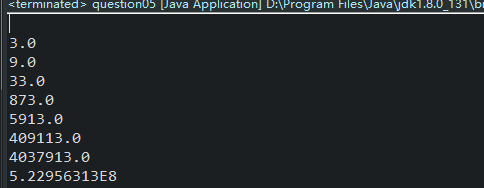
}

}

这个类用来输出，每个一秒输出一次

需要注意的是，因为阶乘和计算得速度太快，所以我加了一个random方法生成随机数来减慢运行的时间

Thread.sleep((**int**) (Math.random() \* 500 + 500));



运行结果

第一个位置为空是因为还未运行完就输出，所以输出了空串null