**《Java语言程序设计B》上机指导**

## 实验一Java 运行环境的配置与使用

## 实验目的

1．掌握安装Java SDK 软件包、Eclipse软件的方法。

2．掌握设置Java 程序运行环境的方法。

3．掌握编写与运行Java 程序的方法。

4．了解Java 语言的概貌。

5. 学习Java 语言和进行Java 程序的实验做好准备工作。

## 实验要求

1．下载、安装并设置Java SDK 软件包。

2．下载并安装Eclipse软件。

3．编写简单的Java 程序。

4．掌握运行Java 程序和Applet的步骤。

## 三、实验内容

### （一）SDK 的下载与安装

1. 机器要求

Java 对机器运行要求不高。下面给出的是基于Windows 平台的机器要求。

硬件要求：CPU PII 以上，64M 内存，1OOM 硬盘空间即可。

软件要求：Windows98/Me/XP/NT/2000，IE 5 以上。

1. 下载SDK

为了建立基于SDK 的Java 运行环境，需要先下载Sun 的免费SDK 软件包。SDK 包含了一整套开发工具，其中包含对编程最有用的是Java 编译器、Applet 查看器和Java 解释器。

在浏览器中输入http://www.oracle.com/，在主页上点击的Downloads一栏，找到java SE,选择java SE 8u101/8u102的SDK，选择对应的平台进行下载。

1. 安装SDK

运行下载的“jdk-8u101-windows-i586.exe”软件包，可安装SDK，在安装过程中可以设置安装路径及选择组件，系统默认的安装路径为C:\jdk1.8.0\_101，默认的组件选择是全部安装。

1. SDK 中的主要文件夹和文件

安装成功后，E:\ j2sdk1.4.2\_01 中的文件和子目录其中，bin 文件夹中包含编译器（javac.exe）、解释器（javac.exe）、Applet 查看器（appletviewer.exe）等可执行文件。

### （二）设置环境变量

SDK 中的工具都是命令行工具，需要从命令行即MS－DOS 提示符下运行它们。

为了能正确方便的使用SDK，可手工配置一些Windows 的环境变量，下面以

Windows/xp/NT/2000为例说明相关配置。

1. 设置Windows XP/NT/2000

* 在桌面“我的电脑”图标上单击右键，选择“属性”菜单，将出现系统特性设置界面
* 在“高级”选项卡中单击“环境变量”按钮。将出现“环境变量”设置界面
* 在“系统变量”框中选择“Path”，然后单击“编辑”按钮，在出现的“编辑系统变量”对话框中，
* 在“变量值”栏的命令前添加“jdk1.8.0\_101；”命令;
* 用同样方法设置环境变量“classpath”，其值为“E:\j2sdk1.4.2\_01\lib;”命令。
* 设置完成后，单击“开始”→“所有程序”→“附件”→“命令提示符”打开DOS 窗口，在命令提示符下输入“java”或“javac”，回车后，如果出现其用法参数提示信息，则安装正确
  1. 设置DOS 起始位置

为了在打开DOS 窗口时可以直接进入包含你开发的java 程序的路径下，下面以Windows XP/2000为例说明设置过程。

* 首先建立一个存放java 源程序的路径和文件夹，如将实验中建立的java 源程序都存放在为“e:\java\程序”文件夹中。
* 单击“开始”→“所有程序”→“附件”→“命令提示符”，选择“命令提示符”选项右击，
* 在出现的下拉菜单中选择“属性”菜单，将打开“命令提示符属性”对话框，如图1.11 所示。
* 在“命令提示符属性”对话框中选择“快捷方式”选项，在“起始位置”栏输入“e:\java\程序”，单击“确定”按钮。

进行了上面的设置后，打开DOS 窗口时将直接进入“e:\java\程序>”路径下。

如果没有进行上述设置，可使用DOS 命令，直接输入“e:”按回车进入到e 盘，再输入“cd java”到e 盘的java 子目录下，再输入“cd 程序”即可到“e:\java\程序>”路径下。

### （三）eclipse的安装与使用

### 1.打开http://www.eclipse.org/downloads/，点击“Eclipse Neon”，选择“[Eclipse IDE for Java EE Developers](http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/neonr" \o "Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others.)”下载，并step by step安装。

### 2.使用eclipse。

a．打开eclipse软件，编辑一java程序并保存；

b.使用编译工具,编译java程序，如显示“Normal Termination”表示编译正确；

c．使用运行工具,运行java字节码文件。

**（四）开发java简单程序**

**1．一个简单的应用程序**

* **实验要求：**

编写一个简单的Java应用程序，该程序在命令行窗口输出两行文字：“你好，很高兴学习Java”和“We are students”。

* **程序运行效果示例：**

程序运行效果如下图所示：

* **程序模板：**

**Hello.java**

public class Hello

{

public static void main (String args[ ])

{

【代码1】 //命令行窗口输出"你好，很高兴学习Java"

A a=new A();

a.fA();

}

}

class A

{

void fA()

{

【代码2】 //命令行窗口输出"We are students"

}

}

* **实验后的练习：**

1. 编译器怎样提示丢失大括号的错误？
2. 编译器怎样提示语句丢失分号的错误？
3. 编译器怎样提示将System写成system这一错误？
4. 编译器怎样提示将String写成string这一错误？

**2．一个简单的Java Applet程序**

* **实验要求：**

编写一个简单的Java Applet程序，并在Java Applet中写两行文字：“这是一个Java Applet程序”和“我改变了字体”。

* **程序模板：**

**FirstApplet.java**

import java.applet.\*;

import java.awt.\*;

public class FirstApplet extends Applet

{

public void paint(Graphics g)

{

g.setColor(Color.blue);

【代码1】//在Java Applet中绘制一行文字：“这是一个Java Applet 程序”

g.setColor(Color.red);

g.setFont(new Font("宋体",Font.BOLD,36));

【代码2】//在Java Applet中绘制一行文字：“我改变了字体”

}

}

* **实验后的练习：**

1. 程序中的主类如果不用public修饰，编译能通过吗？
2. 程序中的主类如果不用public修饰，程序能正确运行吗？
3. 程序将paint方法误写成Paint，编译能通过么？
4. 程序将paint方法误写成Paint，运行时能看到有关的输出信息吗？

**3．联合编译**

* **实验要求：**

编写4个源文件：Hello.java、A.java、B.java和C.java，每个源文件只有一个类，Hello.java是一个应用程序（含有main方法），使用了A、B和C类。将4个源文件保存到同一目录中，例如：C:\100，然后编译Hello.java。

* **程序运行效果示例：**

程序运行效果如下图所示：

* **程序模板：**

**模板1：Hello.java**

public class MainClass

{

public static void main (String args[ ])

{

【代码1】 //命令行窗口输出"你好，只需编译我"

A a=new A();

a.fA();

B b=new B();

b.fB();

}

}

**模板2 ：A.java**

publicclass A

{

void fA()

{

【代码2】 //命令行窗口输出"I am A"

}

}

**模板3 ：B.java**

publicclass B

{

void fB()

{

【代码3】 //命令行窗口输出"I am B"

}

}

**模板4 ：C.java**

publicclass C

{

void fC()

{

【代码4】 //命令行窗口输出"I am C"

}

}

* **实验后的练习：**

1. 将Hello.java编译通过后，不断修改A.java源文件中的代码，比如，在命令行窗口输出“我是A类”或“我被修改了”。要求每次修改A.java源文件后，单独编译A.java，然后直接运行应用程序Hello。
2. 如果需要编译某个目录下的全部Java源文件，比如C:\1000目录，可以使用如下命令：c:\1000\javac \*.java

【结果】

将本次上机实验结果按照以上内容依次填写实验目标、过程、结果。

【注意事项】

## 实验二类与对象

## 一、实验目的

通过编程和上机实验理解 Java 语言是如何体现面向对象编程基本思想，了解类的封装方法，以及如何创建类和对象，了解成员变量和成员方法的特性，掌握 OOP 方式进行程序设计的方法。

## 二、实验要求

1. 编写一个创建对象和使用对象的方法的程序。
2. 编写一个包含类成员和示例成员的程序。
3. 编写一个使用Java包的程序。

## 三、实验内容

（一）**三角形、梯形和圆形的类封装**

* **实验要求：**

编写一个Java应用程序，该程序中有3个类：Trangle、Leder和Circle，分别用来刻画“三角形”、“梯形”和“圆形”。具体要求如下：

a) Trangle类具有类型为double的三个边，以及周长、面积属性，Trangle类具有返回周长、面积以及修改三个边的功能。另外，Trangle类还具有一个boolean型的属性，该属性用来判断三个属能否构成一个三角形。

b) Lader类具有类型double的上底、下底、高、面积属性，具有返回面积的功能。

c) Circle类具有类型为double的半径、周长和面积属性，具有返回周长、面积的功能。

* **程序模板：**

**AreaAndLength.java**

class Trangle

{

double sideA,sideB,sideC,area,length;

boolean boo;

public Trangle(double a,double b,double c)

{

【代码1】 //参数a,b,c分别赋值给sideA,sideB,sideC

if(【代码2】) //a,b,c构成三角形的条件表达式

{

【代码3】 //给boo赋值。

}

else

{

【代码4】 //给boo赋值。

}

}

double getLength()

{

【代码5】 //方法体，要求计算出length的值并返回

}

public double getArea()

{

if(boo)

{

double p=(sideA+sideB+sideC)/2.0;

area=Math.sqrt(p\*(p-sideA)\*(p-sideB)\*(p-sideC)) ;

return area;

}

else

{

System.out.println("不是一个三角形,不能计算面积");

return 0;

}

}

public void setABC(double a,double b,double c)

{

【代码6】 //参数a,b,c分别赋值给sideA,sideB,sideC

if(【代码7】) //a,b,c构成三角形的条件表达式

{

【代码8】 //给boo赋值。

}

else

{

【代码9】 //给boo赋值。

}

}

}

class Lader

{

double above,bottom,height,area;

Lader(double a,double b,double h)

{

【代码10】 //方法体，将参数a,b,c分别赋值给above,bottom,height

}

double getArea()

{

【代码11】 //方法体，,要求计算出area返回

}

}

class Circle

{

double radius,area;

Circle(double r)

{

【代码12】 //方法体

}

double getArea()

{

【代码13】 //方法体，要求计算出area返回

}

double getLength()

{

【代码14】 //getArea方法体的代码,要求计算出length返回

}

void setRadius(double newRadius)

{

radius=newRadius;

}

double getRadius()

{

return radius;

}

}

public class AreaAndLength

{

public static void main(String args[])

{

double length,area;

Circle circle=null;

Trangle trangle;

Lader lader;

【代码15】 //创建对象circle

【代码16】 //创建对象trangle。

【代码17】 //创建对象lader

【代码18】 // circle调用方法返回周长并赋值给length

System.out.println("圆的周长:"+length);

【代码19】 // circle调用方法返回面积并赋值给area

System.out.println("圆的面积:"+area);

【代码20】 // trangle调用方法返回周长并赋值给length

System.out.println("三角形的周长:"+length);

【代码21】 // trangle调用方法返回面积并赋值给area

System.out.println("三角形的面积:"+area);

【代码22】 // lader调用方法返回面积并赋值给area

System.out.println("梯形的面积:"+area);

【代码23】 // trangle调用方法设置三个边，要求将三个边修改为12,34,1。

【代码24】 // trangle调用方法返回面积并赋值给area

System.out.println("三角形的面积:"+area);

【代码25】 // trangle调用方法返回周长并赋值给length

System.out.println("三角形的周长:"+length);

}

}

* **实验后的练习：**

1. 程序中仅仅省略【代码15】，编译能通过吗？

2. 程序中仅仅省略【代码16】，编译能通过吗？

3. 程序中仅仅省略【代码15】，运行时出现怎样的异常提示？

1. 给Trangle类增加3个方法，分别用来返回3个边：sideA、sideB和sideC。
2. 让AreaAndLength类中的circle对象调用方法修改半径，然后输出修改后的半径以及修改半径后的圆的面积和周长。

（二）**实例成员与类成员。**

* **实验要求：**

按程序模板的要求编写源文件，要特别注意程序的输出结果，并能正确解释输出的结果。

* **程序运行效果示例：**

程序运行效果如下图所示：

* **程序模板：**

**Example.java**

class A

{

【代码1】 //声明一个float型实例变量a

【代码2】 //声明一个float型类变量b,

void setA(float a)

{

【代码3】 //将参数a的值赋值给成员变量a

}

void setB(float b)

{

【代码4】 //将参数b的值赋值给成员变量b

}

float getA()

{

return a;

}

float getB()

{

return b;

}

void inputA()

{

System.out.println(a);

}

static void inputB()

{

System.out.println(b);

}

}

public class Example

{

public static void main(String args[])

{

【代码5】 //通过类名操作类变量b,并赋值100

【代码6】 //通过类名调用方法inputB()

A cat=new A();

A dog=new A();

【代码7】 //cat调用方法setA(int a)将cat的成员a的值设置为200

【代码8】 //cat调用方法setB(int b)将cat的成员b的值设置为400

【代码9】 //dog调用方法setA(int a)将dog的成员a的值设置为150

【代码10】 //dog调用方法setB(int b)将dog的成员b的值设置为300

【代码11】//cat调用inputA()。

【代码12】//cat调用inputB()。

【代码13】//dog调用inputA()。

【代码14】//dog调用inputB()。

}

}

* **实验后的练习：**

1. 将inputA()方法中的

System.out.println(a);

改写为

System.out.println(a+b);

编译是否出错？为什么？

2. 将inputB()方法中的

System.out.println(b);

改写为

System.out.println(a+b);

编译是否报错？为什么？

**（三）使用package语句与import语句。**

* **实验要求：**

按实验要求使用package语句，并用import语句使用Java平台提供的包中的类以及

自定义包中的类。掌握一些重要的操作步骤。

**程序模板：**

**模板1：**

将模板1给出的Java源文件命名为SquareEquation.java，将编译后得到的字节码文件复制到c:\1000\tom\jiafei目录中。

**SquareEquation.java**

package tom.jiafei;

public class SquareEquation

{

double a,b,c;

double root1,root2;

boolean boo;

public SquareEquation(double a,double b,double c)

{

this.a=a;

this.b=b;

this.c=c;

if(a!=0)

{

boo=true;

}

else

{

boo=false;

}

}

public void getRoots()

{

if(boo)

{

System.out.println("是一元2次方程");

double disk=b\*b-4\*a\*c;

if(disk>=0)

{

root1=(-b+Math.sqrt(disk))/(2\*a);

root2=(-b-Math.sqrt(disk))/(2\*a);

System.out.printf("方程的根:%f,%f\n",root1,root2);

}

else

{

System.out.printf("方程没有实根\n");

}

}

else

{

System.out.println("不是一元2次方程");

}

}

public void setCoefficient(double a,double b,double c)

{

this.a=a;

this.b=b;

this.c=c;

if(a!=0)

{

boo=true;

}

else

{

boo=false;

}

}

}

**模板2：**

将模板2给出的Java源程序SunRise.java保存到d:\2000中。在编译模板2给出的Java源文件之前，要重新设置classpath。假设本地机SDK的安装目录是E:\jdk1.8。

在命令行执行如下命令：

set classpath=e:\jdk1.8\jre\lib\rt.jar;.;c:\1000

或用鼠标右键点击“我的电脑”，弹出快捷菜单，然后选择“属性”命令，弹出“系统属性”对话框，再单击该对话框中的“高级”选项卡，然后单击“环境变量”按钮。将classpath的值修改为：

E:\jdk1.5\jre\lib\rt.jar;.;c:\1000

然后重新打开一个命令行窗口，编译模板2给出的Java源程序。

**SunRise.java**

import tom.jiafei.\*;

class SunRise

{

public static void main(String args[])

{

SquareEquation equation=new SquareEquation(4,5,1);

equation.getRoots();

equation.setCoefficient(-3,4,5);

equation.getRoots();

}

}

【结果】

将本次上机实验结果按照实验要求内容依次进行，填写实验目标、过程、结果。

【注意事项】

①实验报告要求每人提交一份，使用专门的报告纸，内容写全，包括实验名称、班级、姓名、同组人，试验日期、交报告日期等。

②报告不需要填写源代码，只要将完成情况和所有练习题结果写上就可以。

## 实验三继承与多态

## 一、实验目的

通过编程和上机实验理解 Java 语言的继承和多态特性，掌握变量的隐藏、方法的覆盖、重载，掌握抽象类和接口的使用。

## 二、实验要求

1.编写体现类的继承性（成员变量、成员方法、成员变量隐藏）的程序。

2.编写体现类的多态性（成员方法重载）的程序。

3.编写体现类的多态性（构造方法重载）的程序。

4.编写使用接口的程序。

## 三、实验内容

**（一）类的继承**

1.创建**公共类**Student.

（1）编写程序文件 Student.java，源代码如下:

**public** class Student

{

**protected**String name; //具有保护修饰符的成员变量

**protected** int number;

void setData(String m,int h) //设置数据的方法

{

name =m;

number= h;

}

public void print() //输出数据的方法

{

System.out.println(name+", "+number);

}

}

（2）编译 Student.java，产生类文件Student.class。

2．创建继承的类[Undergraduate](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);)

（1）程序功能：通过 Student 类产生子类[undergraduate](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);)，其不仅具有父类的成员变量

name（姓名）、number（学号），还定义了新成员变量[academy](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);)（学院）、department

（系）。在程序中调用父类的 print 方法。

1. 编写[Undergraduate](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);)程序:

class[Undergraduate](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);) extends Student

{

【代码1】 //定义成员变量[academy](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);)

【代码2】 //定义成员变量department

public static void main(String args[])

{

【代码3】 //创建一个学生对象s

【代码4】 //用父类的setData方法初始化对象s

【代码5】 //对象s调用print方法

【代码6】 //创建一个大学生对象u

【代码7】//调用父类的成员方法setData初始化对象u

【代码8】 //设置对象u的成员变量[academy](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);)

【代码9】 //设置对象u的成员变量department

System.out.print(u.name+", "+u.number+", "+u.[academy](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);)+", "+u.department);

}

}

（3）编译并运行程序

注意：公共类 Student 与[undergraduate](D:/Documents%20and%20Settings/Administrator/Local%20Settings/Application%20Data/Youdao/Dict/6.3.67.7016/resultui/frame/javascript:void(0);)类要在同一文件夹（路径）内。

**(二)方法的重载**

（1）程序功能：对不同的数进行排序输出。在IntSort 类中定义3 个同名的方法sort。

（2）编写Sort.java 文件，源代码如下。

importjava.awt.Graphics;

import java.applet.Applet;

class IntSort {

public String sort(int a, int b) {

if (a>b)

return a+""+b;

else

return b+""+a;

}

public String sort(int a, int b, int c) {

int swap;

if (a<b) {

swap=a;

a=b;

b=swap;

}

if (a<c) {

swap=a;

a=c;

c=swap;

}

54

if (b<c) {

swap=b;

b=c;

c=swap;

}

return a+""+b+""+c;

}

public String sort(int arr[]) {

String s="";

int swap;

for (int i=0; i<arr.length; i++)

for (int j=0; j<arr.length-1; j++)

if (arr[j]>arr[j+1]) {

swap=arr[j];

arr[j]=arr[j+1];

arr[j+1]=swap;

}

for (int i=0; i<arr.length; i++)

s=s+arr[i]+"";

return s;

}

}

public class Sort extends Applet {

IntSort s=new IntSort();

public void paint(Graphics g) {

int a=30, b=12, c=40;

int arr[]={34,8,12,67,44,98,52,23,16,16};

g.drawString("两个数的排序结果："+s.sort(a,b),30,30);

g.drawString("三个数的排序结果："+s.sort(a,b,c),30,60);

g.drawString("数组的排序结果："+s.sort(arr),30,90);

}

}

（3）编译并运行程序

（4）编写Sort类文件的页面文件sort.html

<html>

<applet codebase=e:/java/程序 code=KY3\_11.class width=420 height=380>

</applet >

</html>

**(三) 构造方法的重载**

（1）编写构造方法User 的重载程序文件RunDemo，源代码如下。

classUser{

private String userName, password;

【代码1】 //定义一个无参构造方法的头部

{

System.out.println("全部为空!");

}

【代码2】 //定义一个构造方法，参数为name，用name初始化成员变量userName

【代码3】 //定义一个构造方法的头部，参数为name和pwd

{

【代码4】 // 调用上个构造方法初始化userName

【代码5】 //初始化成员变量password

check();

}

void check() {

String s=null;

if (userName!=null)

s="用户名："+userName;

else

s="用户名不能为空！";

if (password!="12345678")

s=s+" 口令无效！";

else

s=s+" 口令：\*\*\*\*\*\*\*\*";

System.out.println(s);

}

}

public class RunDemo {

public static void main(String[] args) {

newUser();

new User("刘新宇");

new User(null,"邵丽萍");

new User("张驰","12345678");

}

}

（2）编译并运行程序

**（四）实现 MouseListener 和 MouseMotionListener 两个接口**

（1）编写实现接口的程序文件 Interface.java，源代码如下。

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

【代码1】 //定义一个实现接口MouseListener, MouseMotionListener并继承Applet的一个类的头部，类名为Interface

{

int x1, y1, x2, y2;

public void init() {

addMouseListener(this);

addMouseMotionListener(this);

}

public void paint(Graphics g) {

【代码2】//画一条直线

}

public void mousePressed(MouseEvent e) { // 记录起点坐标

x1=e.getX();

y1=e.getY();

}

public void mouseClicked(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseMoved(MouseEvent e) {}

public void mouseDragged(MouseEvent e) { // 记录终点坐标

x2=e.getX();

y2=e.getY();

repaint();

}

}

* + 编译 Interface.java 文件。
  + 编写 Interface.html 文件，源代码如下。

<html>

<applet codebase=e:/java/程序 code=Interface.classwidth=320 height=180>

</applet >

</html>

（5）在浏览器中打开 Interface.html 文件，在窗口中拖动鼠标可以随意画出一条线

## 四、思考题

1. 使用变量之前是不是都要先声明变量。使用变量之前是不是都要先赋值，为什么？
2. 什么是构造方法。
3. 类的继承和多态有什么作用，在使用上应该注意什么问题。

【结果】

将本次上机实验结果按照实验要求内容依次进行，填写实验目标、过程、结果。

【注意事项】

## 实验四数组与字符串

## 一、实验目的

掌握Java中的数组定义、引用，掌握String类和StringBuffer类的使用。

## 二、实验要求

1. 编写一个使用 Java 数组的的程序。
2. 编写一个使用String类的程序。
3. 编写一个使用StringBuffer类的程序。

## 三、实验内容

**(一) 使用数组求Fibonacci数列前10项之和**

1.编写程序。（代码附到实验报告中）

2.编译并运行程序。

**（二）练习String类的常用方法**

* **实验要求：**

编写一个Java应用程序，判断两个字符串是否相同，判断字符串的前缀、后缀是否和某个字符串相同，按字典顺序比较两个字符串的大小关系，检索字符串，创建字符串，将数字型字符串转换为数字，将字符串存放到数组中，用字符数组创建字符串。。

* **程序模板：**

**StringExample.java**

class StringExample

{ public static void main(String args[])

{ String s1=new String("you are a student"),s2=new String("how are you");

if(【代码1】) // 使用equals方法判断s1与s2是否相同

{

System.out.println("s1与s2相同");

}

else

{

System.out.println("s1与s2不相同");

}

String s3=new String("22030219851022024");

if(【代码2】) //判断s3的前缀是否是“220302”。

{

System.out.println("吉林省的身份证");

}

String s4=new String("你"),

s5=new String("我");

if(【代码3】)//按着字典序s4大于s5的表达式。

{

System.out.println("按字典序s4大于s5");

}

else

{

System.out.println("按字典序s4小于s5");

}

int position=0;

String path="c:\\java\\jsp\\A.java";

position=【代码5】 //获取path中最后出现目录分隔符号的位置

System.out.println("c:\\java\\jsp\\A.java中最后出现\\的位置:"+position);

String fileName=【代码6】//获取path中“A.java”子字符串。

System.out.println("c:\\java\\jsp\\A.java中含有的文件名:"+fileName);

String s6=new String("100"), s7=new String("123.678");

int n1=【代码7】 //将s6转化成int型数据。

double n2=【代码8】 //将s7转化成double型数据。

double m=n1+n2;

System.out.println(m);

String s8=【代码9】 //String调用valuOf(int n)方法将m转化为字符串对象

position=s8.indexOf(".");

String temp=s8.substring(position+1);

System.out.println("数字"+m+"有"+temp.length()+"位小数") ;

String s9=new String("ABCDEF");

char a[]=【代码10】 //将s8存放到数组a中。

for(int i=a.length-1;i>=0;i--)

{

System.out.print(" "+a[i]);

}

}

}

* **实验后的练习：**

1. 程序中的s6改写成

String s6=new String(“1a12b”);

运行时提示怎样的错误？为什么？

2. 请用数组a的前3个单元创建一个字符串并输出该串。

3. 请给出获取path中“jsp”子字符串的代码。

4. 在程序的适当位置增加如下代码，注意输出的结果。

String str1=new String(“ABCABC”),

str2=null,

str3=null,

str4=null;

str2=str1.replaceAll(“A”,”First”);

str3=str2.replaceAll(“B”,”Second”);

str4=str3.replaceAll(“C”,”Third”);

System.out.println(str1);

System.out.println(str2);

System.out.println(str3);

System.out.println(str4);

5. 可以使用Long类中的下列static方法得到整数各种进制的字符串表示：

Public static String toBinaryString(long i)

Public static String toOctalString(long i)

Public static String toHexString(long i)

Public static String toString(long i,int p)

其中的toString(long i, int p)返回整数i的p进制表示。请在适当位置添加代码输出12345的二进制、八进制和十六进制表示。

6. 在适当位置添加代码，分别输出数字m的整数部分和小数部分。

1. **使用StringBuffer类。**

* **程序模板：**

import java.applet.Applet;   
 import java.awt.Graphics;

public class LX5\_2 extends Applet {

public void paint(Graphics g) {  
 String str="这是一个字符串This is a test string";

g.drawString("正常字符串："+str, 30, 30);   
 g.drawString("翻转字符串："+reverse(str), 30, 60);

}  
 public String reverse(String s) {

int len=s.length();   
 StringBuffer buffer=【代码1】//创建一个大小为len的StringBuffer对象，并赋给buffer变量  
 【代码2】 // 将字符串s的反转字符串存入buffer.

return buffer.toString();

}

}

## 实验五Java基本类

## 一、实验目的

了解Java 常用的系统类，包括字符串类、System类、数学函数类、日期类、随机数类等的基本使用方法。

## 二、实验要求

1. 编写一个使用日期类的的程序。

1. 编写一个使用日历类和包装类的程序。
2. 编写一个使用数学函数的程序。
3. 编写一个使用集合框架的程序。

## 实验内容

## （一）使用日期类的Applet程序

（1）程序功能：说明3 个日期类Date、Calendar和DateFormat 的使用方式及显示的样式。

（2）编写LX5\_1java 程序文件，源代码如下。

import java.text.\*;   
 import java.util.\*;   
 import java.awt.\*;   
 import java.applet.\*;

public class LX5\_1 extends Applet {

public void paint(Graphics g) {   
 Date today;   
 Calendar now;   
 DateFormat f1,f2;   
 String s1,s2;

【代码1】; // 获取系统当前日期，存放到today变量

g.drawString("字符串格式："+today.toString(),20,20);

【代码2】; // 以默认格式生成格式化器，存放到f1变量  
 【代码3】; // 将日期转换为字符串，存放到s1变量  
 g.drawString("系统格式："+s1,20,40); // 生成长格式的中国日期格式化器

f1=DateFormat.getDateInstance(DateFormat.LONG, Locale.CHINA);

f2=DateFormat.getTimeInstance(DateFormat.LONG, Locale.CHINA);

s1=f1.format(today); // 将日期转换为日期字符串

s2=f2.format(today); // 将日期转换为时间字符串

g.drawString("中国格式："+s1+" "+s2,20,60);

【代码4】// 通过Calendar类获取系统时间，存放到now变量

s1=now.get(now.HOUR)+" 时"+now.get(now.MINUTE)+" 分"+now.get(now.SECOND)+"秒";

g.drawString("调整前时间："+s1,20,80);

【代码5】 //将now中的时间修改为2016年12月31日

today=now.getTime();

g.drawString("调整后时间："+today.toString(),20,100);

}  
}

（3）编译程序文件。

（4）运行程序。

## （二）使用日历类和包装类的程序

* **实验要求：**

编写一个Java应用程序，用户从输入对话框输入了两个日期，程序将判断两个日期的大小关系，以及两个日期之间的间隔天数。

* **程序模板：**

**DateExample**

import java.util.\*;

import javax.swing.JOptionPane;

public class DateExample

{

public static void main(String args[ ])

{

String str=JOptionPane.showInputDialog("输入第一个日期的年份:");

int yearOne=Integer.parseInt(str);

str=JOptionPane.showInputDialog("输入该年的月份:");

int monthOne=Integer.parseInt(str);

str=JOptionPane.showInputDialog("输入该月份的日期:");

int dayOne=Integer.parseInt(str);

str=JOptionPane.showInputDialog("输入第二个日期的年份:");

int yearTwo=Integer.parseInt(str);

str=JOptionPane.showInputDialog("输入该年的月份:");

int monthTwo=Integer.parseInt(str);

str=JOptionPane.showInputDialog("输入该月份的日期:");

int dayTwo=Integer.parseInt(str);

Calendar calendar=【代码1】 //初始化日历对象

【代码2】 //将calendar的时间设置为yearOne年monthOne月dayOne日

long timeOne=【代码3】 //calendar表示的时间转换成毫秒

【代码4】 //将calendar的时间设置为yearTwo年monthTwo月dayTwo日

long timeTwo=【代码5】 //calendar表示的时间转换成毫秒。

Date date1=【代码6】 //用timeOne做参数构造date1

Date date2=【代码7】 //用timeTwo做参数构造date2

if(date2.equals(date1))

{

System.out.println("两个日期的年、月、日完全相同");

}

else if(date2.after(date1))

{

System.out.println("您输入的第二个日期大于第一个日期");

}

else if(date2.before(date1))

{

System.out.println("您输入的第二个日期小于第一个日期");

}

long days=【代码8】//计算两个日期相隔天数

System.out.println(yearOne+"年"+monthOne+"月"+dayOne+"日和"

+yearTwo+"年"+monthTwo+"月"+dayTwo+"相隔"+days+"天");

}

}

* **实验后的练习：**

1. Calendar对象可以将时间设置到年、月、日、时、分、秒。请改进上面的程序，使用户输入的两个日期包括时、分、秒。

2. 根据本程序中的一些知识，编写一个计算利息（按天计算）的程序。从输入对话框输入存款的数目和起止时间。

**（三）Math类的使用。**

Math 是一个最终类，含有基本数学运算函数，如指数运算、对数运算、求平方根、三角函数、随机数等，可以直接在程序中加Math.前缀调用。

1. 创建使用Math 类的应用程序KY5\_2.java，源程序如下。

import java.util.\*;

class KY5\_2 {

public static void main(String args[]) {

Random r1=new Random(1234567890L);

Random r2=new Random(1234567890L);

boolean b=r1.nextBoolean(); // 随机数不为0 时取真值

int i1=r1.nextInt(100); // 产生大于等于0 小于100 的随机数

int i2=r2.nextInt(100); // 同上

double i3=r1.nextDouble(); // 产生大于等于0.0 小于1.0 的随机数

double i4=r2.nextDouble(); // 同上

double d1=Math.sin(Math.toRadians(30.0));

double d2=Math.log(Math.E);

double d3=Math.pow(2.0, 3.0);

int r=Math.round(33.6F);

System.out.println("b 的随机数不为0 时 "+b);

System.out.println("i1 的随机数为 "+i1);

System.out.println("i2 的随机数为 "+i2);

System.out.println("d1 的随机数为 "+i3);

System.out.println("d2 的随机数为 "+i4);

System.out.println("30 弧度的正弦值：Math.sin(Math.toRadians(30.0))＝ "+d1);

System.out.println("E 的对数值：Math.log(Math.E)＝ "+d2);

System.out.println("2 的3 次方：Math.pow(2.0, 3.0)＝ "+d3);

System.out.println("33.6F 四舍五入：Math.round(33.6F)＝ "+r);

}

}

1. 编译 KY5\_2.java 文件。
2. 运行 KY5\_2.class 文件，并输入 1234

**（四）使用ArrayList线性表**

* **实验要求：**

编写程序，在ArrayList线性表中进行定位、查找和迭代。

* **程序模板：**

**ArrayListDemo**

import java.util.\*;

class ArrayListDemo {

public static void main(String[] args) {

ArrayList myWeekday =【代码1】; //创建一个ArrayList

【代码2】//将"Sunday"、"Monday"、"Wednesday"、"Friday"逐个加入线性表 System.out.println(myWeekday);

【代码3】//插入"Tuesday"，作为第二个元素

【代码4】 //将第4个元素替换为"Thursday"

【代码5】 //删除第0个元素

【代码6】 //获取线性表的迭代器

【代码7】 //迭代获得线性表的每个元素，并打印

}

}

【结果】

将本次上机实验结果按照实验要求内容依次进行，填写实验目标、过程、结果。

【注意事项】

①实验报告要求每人提交一份，使用专门的报告纸，内容写全，包括实验名称、班级、姓名、同组人，试验日期、交报告日期等。

②报告不需要填写源代码，只要将完成情况和所有练习题结果写上就可以。

## 实验六 图形用户界面

## 一、实验目的

了解图形用户界面基本组件窗口、按钮、文本框、选择框、滚动条等的使用方法，了解如何使用布局管理器对组件进行管理，以及如何使用Java 的事件处理机制。

## 二、实验要求

1. 掌握使用布局管理器对组件进行管理的方法。

2. 理解 Java 的事件处理机制，掌握为不同组件编写事件处理程序的方法。

3. 掌握编写独立运行的窗口界面的方法。

4. 掌握组件的使用方法。

5. 了解对话框的使用方法。

## 三、实验内容

**（一）算术测试。**

* **实验要求：**

编写一个算术测试小软件，用来训练小学生的算术能力。程序由3个类组成，其中Teacher类对象负责给出算术题目，并判断回答者的答案是否正确；ComputerFrame类对象负责为算术题目提供视图，比如用户可以通过ComputerFrame类对象提供的GUI界面看到题目，并通过该GUI界面给出题目的答案；MainClass是软件的主类。

* **程序模板：**

**Teacher.java**

public class Teacher

{ int numberOne,numberTwo;

String operator="";

boolean right;

public int giveNumberOne(int n)

{ numberOne=(int)(Math.random()\*n)+1;

return numberOne;

}

public int giveNumberTwo(int n)

{ numberTwo=(int)(Math.random()\*n)+1;

return numberTwo;

}

public String giveOperator()

{ double d=Math.random();

if(d>=0.5)

operator="+";

else

operator="-";

return operator;

}

public boolean getRight(int answer)

{ if(operator.equals("+"))

{ if(answer==numberOne+numberTwo)

right=true;

else

right=false;

}

else if(operator.equals("-"))

{ if(answer==numberOne-numberTwo)

right=true;

else

right=false;

}

return right;

}

}

**ComputerFrame.java**

import java.awt.\*;

import java.awt.event.\*;

public class ComputerFrame extends Frame implements ActionListener

{ TextField textOne,textTwo,textResult;

Button getProblem,giveAnwser;

Label operatorLabel,message;

Teacher teacher;

ComputerFrame(String s)

{ super(s);

teacher=new Teacher();

setLayout(new FlowLayout());

textOne=【代码1】 //创建textOne,其可见字符长是10

textTwo=【代码2】 //创建textTwo,其可见字符长是10

textResult=【代码3】 //创建textResult,其可见字符长是10

operatorLabel=new Label("+");

message=new Label("你还没有回答呢");

getProblem=new Button("获取题目");

giveAnwser=new Button("确认答案");

add(getProblem);

add(textOne);

add(operatorLabel);

add(textTwo);

add(new Label("="));

add(textResult);

add(giveAnwser);

add(message);

textResult.requestFocus();

textOne.setEditable(false);

textTwo.setEditable(false);

【代码4】//将当前窗口注册为getProblem的ActionEvent事件监视器

【代码5】//将当前窗口注册为giveAnwser的ActionEvent事件监视器

【代码6】//将当前窗口注册为textResult的ActionEvent事件监视器

setBounds(100,100,450,100);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

}

);

}

public void actionPerformed(ActionEvent e)

{ if(【代码7】) //判断事件源是否是getProblem

{ int number1=teacher.giveNumberOne(100);

int number2=teacher.giveNumberTwo(100);

String operator=teacher.givetOperator();

textOne.setText(""+number1);

textTwo.setText(""+number2);

operatorLabel.setText(operator);

message.setText("请回答");

textResult.setText(null);

}

if(【代码8】) //判断事件源是否是giveAnwser

{ String answer=textResult.getText();

try{

int result=Integer.parseInt(answer);

if(teacher.getRight(result)==true)

{ message.setText("你回答正确");

}

else

{ message.setText("你回答错误");

}

}

catch(NumberFormatException ex)

{ message.setText("请输入数字字符");

}

}

textResult.requestFocus();

validate();

}

}

**MainClass.java**

public class MainClass

{ public static void main(String args[])

{ ComputerFrame frame;

frame=【代码9】//创建窗口，其标题为：算术测试

}

}

* **实验后的练习：**

1. 给上述程序增加测试乘、除的功能。

**（二）信号灯。**

* **实验要求：**

编写一个带有窗口的应用程序。在窗口的北面添加一个下拉列表，该下拉列表有“”、“”和“”三个选项。在窗口的中心添加一个画布，当用户在下拉列表选择某项后，画布上绘制相应的信号灯。

* **程序模板：**

**SignalCanvas.java**

import java.awt.\*;

public class SignalCanvas extends Canvas

{ int red,green,yellow,x,y,r;

SignalCanvas()

{ setBackground(Color.white);

}

public void setRed(int r)

{ red=r;

}

public void setGreen(int g)

{ green=g;

}

public void setYellow(int y)

{ yellow=y;

}

public void setPosition(int x,int y)

{ this.x=x;

this.y=y;

}

public void setRadius(int r)

{ this.r=r;

}

public void paint(Graphics g)

{ if(red==1)

{ g.setColor(Color.red);

}

else if(green==1)

{ g.setColor(Color.green);

}

else if(yellow==1)

{ g.setColor(Color.yellow);

}

g.fillOval(x,y,2\*r,2\*r);

}

}

**SignalFrame.java**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

public class SignalFrame extends Frame implements ItemListener

{ Choice choice;

SignalCanvas signal=null;

String itemRed="红灯",itemYellow="黄灯",itemGreen="绿灯";

public SignalFrame()

{ choice=【代码1】 //创建choice

【代码2】 //创建choice添加itemRed

【代码3】 //创建choice添加itemYellow

【代码4】 //创建choice添加itemGreen

【代码5】 //将当前窗口注册为choice的ItemEvent事件监视器

add(choice,BorderLayout.NORTH);

try{ Class cs=Class.forName("SignalCanvas");

signal=(SignalCanvas)cs.newInstance();

add(signal,BorderLayout.CENTER);

}

catch(Exception e)

{ add(new Label("您还没有编写SignalCanvas类"),BorderLayout.CENTER);

}

setBounds(100,100,360,300);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

}

);

}

public void itemStateChanged(ItemEvent e)

{ String item= 【代码6】 // choice返回被选中的条目

int w=signal.getBounds().width;

int h=signal.getBounds().height;

int m=Math.min(w,h);

signal.setRadius(m/6);

if(item.equals(itemRed))

{ if(signal!=null)

{ signal.setRed(1);

signal.setYellow(0);

signal.setGreen(0);

signal.setPosition(w/3,0);

signal.repaint();

}

}

else if(item.equals(itemYellow))

{ if(signal!=null)

{ signal.setRed(0);

signal.setYellow(1);

signal.setGreen(0);

signal.setPosition(w/3,h/3);

signal.repaint();

}

}

else if(item.equals(itemGreen))

{ if(signal!=null)

{ signal.setRed(0);

signal.setYellow(0);

signal.setGreen(1);

signal.setPosition(w/3,2\*h/3);

signal.repaint();

}

}

}

}

**SignalMainClass.java**

public class SignalMainClass

{ public static void main(String args[])

{ SignalFrame frame;

frame=new SignalFrame() ;

frame.setTitle("信号灯");

}

}

* **实验后的练习：**

1. 改进上述程序，在下拉列表中增加“熄灭所有灯”选项，当选中该项时，画布上绘制一个半径为0，位置是（0,0）的圆。

**（三）布局与日历。**

* **实验要求：**

编写一个应用程序，有一个窗口，该窗口为BorderLayout布局。窗口的中心添加一个Panel容器：pCenter，pCenter的布局是7行7列的GriderLayout布局，pCenter中放置49个标签，用来显示日历。窗口的北面添加一个Panel容器pNorth，其布局是FlowLayout布局，pNorth放置两个按钮：nextMonth和previousMonth，单击nextMonth按钮，可以显示当前月的下一月的日历；单击previousMonth按钮，可以显示当前月的上一月的日历。窗口的南面添加一个Panel容器pSouth，其布局是FlowLayout布局，pSouth中放置一个标签用来显示一些信息。

* **程序模板：**

CalendarBean.java

import java.util.Calendar;

public class CalendarBean

{

String day[];

int year=2005,month=0;

public void setYear(int year)

{ this.year=year;

}

public int getYear()

{ return year;

}

public void setMonth(int month)

{ this.month=month;

}

public int getMonth()

{ return month;

}

public String[] getCalendar()

{ String a[]=new String[42];

Calendar 日历=Calendar.getInstance();

日历.set(year,month-1,1);

int 星期几=日历.get(Calendar.DAY\_OF\_WEEK)-1;

int day=0;

if(month==1||month==3||month==5||month==7||month==8||month==10||month==12)

{ day=31;

}

if(month==4||month==6||month==9||month==11)

{ day=30;

}

if(month==2)

{ if(((year%4==0)&&(year%100!=0))||(year%400==0))

{ day=29;

}

else

{ day=28;

}

}

for(int i=星期几,n=1;i<星期几+day;i++)

{

a[i]=String.valueOf(n) ;

n++;

}

return a;

}

}

**CalendarFrame.java**

import java.util.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class CalendarFrame extends Frame implements ActionListener

{ Label labelDay[]=new Label[42];

Button titleName[]=new Button[7];

String name[]={"日","一","二","三", "四","五","六"};

Button nextMonth,previousMonth;

int year=2006,month=10;

CalendarBean calendar;

Label showMessage=new Label("",Label.CENTER);

public CalendarFrame()

{ Panel pCenter=new Panel();

【代码1】 //将pCenter的布局设置为7行7列的GridLayout 布局。

for(int i=0;i<7;i++)

{ titleName[i]=new Button(name[i]);

【代码2】//pCenter添加组件titleName[i]。

}

for(int i=0;i<42;i++)

{

labelDay[i]=new Label("",Label.CENTER);

【代码3】//pCenter添加组件labelDay[i]。

}

calendar=new CalendarBean();

calendar.setYear(year);

calendar.setMonth(month);

String day[]=calendar.getCalendar();

for(int i=0;i<42;i++)

{ labelDay[i].setText(day[i]);

}

nextMonth=new Button("下月");

previousMonth=new Button("上月");

nextMonth.addActionListener(this);

previousMonth.addActionListener(this);

Panel pNorth=new Panel(),

pSouth=new Panel();

pNorth.add(previousMonth);

pNorth.add(nextMonth);

pSouth.add(showMessage);

showMessage.setText("日历："+calendar.getYear()+"年"+ calendar.getMonth()+"月" );

ScrollPane scrollPane=new ScrollPane();

scrollPane.add(pCenter);

【代码4】// 窗口添加scrollPane在中心区域

【代码5】// 窗口添加pNorth 在北面区域

【代码6】// 窗口添加pSouth 在南区域。

}

public void actionPerformed(ActionEvent e)

{ if(e.getSource()==nextMonth)

{ month=month+1;

if(month>12)

month=1;

calendar.setMonth(month);

String day[]=calendar.getCalendar();

for(int i=0;i<42;i++)

{ labelDay[i].setText(day[i]);

}

}

else if(e.getSource()==previousMonth)

{ month=month-1;

if(month<1)

month=12;

calendar.setMonth(month);

String day[]=calendar.getCalendar();

for(int i=0;i<42;i++)

{ labelDay[i].setText(day[i]);

}

}

showMessage.setText("日历："+calendar.getYear()+"年"+calendar.getMonth()+"月" );

}

}

**CalendarMainClass.java**

public class CalendarMainClass

{ public static void main(String args[])

{ CalendarFrame frame=new CalendarFrame();

frame.setBounds(100,100,360,300);

frame.setVisible(true);

frame.validate();

frame.addWindowListener(new java.awt.event.WindowAdapter()

{ public void windowClosing(java.awt.event.WindowEvent e)

{ System.exit(0);

}

}

);

}

}

* **实验后的练习：**
  + - 1. 请在CalendarFrame类中增加一个TextField文本框，用户可以通过在文本框中输入年份来修改calendar对象的int成员year。

**（四）选择复选框和单选框按钮的事件处理程序。**

（1）程序功能：在Applte 上创建复选框、单选框、文本区域、单行文本框等组件，并实现根据用户输入的10 进制数，选择不同选项可转换为2、8、16进制数。

（2）编写LX6\_4.java 程序文件，源代码如下。

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

public class LX6\_1 extends Applet implements ItemListener {

TextArea area=new TextArea(6,30);//创建文本区   
String Item[]={"2 进制","8进制","16进制","10进制"};   
Checkbox cb[]=new Checkbox[5];   
Checkbox radio[]=new Checkbox[5];   
Label l=new Label("输入10 进制数");   
TextField TF=new TextField(6);// 创建单行文本框   
public void init() {

add(l);add(TF);

add(area); add(new Label(" 请选择进制："));   
for(int i=0; i<4; i++) {

cb[i]=new Checkbox(Item[i]); add(cb[i]);

cb[i].addItemListener(this);

}

CheckboxGroup cbGroup=new CheckboxGroup();//创建单选框

add(new Label("请选择进制："));

for(int i=0; i<4; i++) {

radio[i]=new

Checkbox(Item[i],cbGroup,false); add(radio[i]);

radio[i].addItemListener(this);

}   
}

public void itemStateChanged(ItemEvent e) {

int x=Integer.parseInt(TF.getText());

if (e.getItem ()=="2 进制")

area.append ("你选择的是"+e.getItem ()+ Integer.toBinaryString(x)+"\n");

if (e.getItem ()=="8 进制")

area.append ("你选择的是"+e.getItem ()+ Integer.toOctalString(x)+"\n");

if (e.getItem ()=="16 进制")

area.append ("你选择的是"+e.getItem ()+Integer.toHexString(x)+"\n");

if (e.getItem ()=="10 进制")

area.append ("你选择的是"+e.getItem ()+x+"\n");

}   
}

1. 编译程序LX6\_1.java。
2. 运行程序LX6\_1。

【结果】

将本次上机实验结果按照实验要求内容依次进行，填写实验目标、过程、结果。

【注意事项】

①实验报告要求每人提交一份，使用专门的报告纸，内容写全，包括实验名称、班级、姓名、同组人，试验日期、交报告日期等。

②报告不需要填写源代码，只要将完成情况和所有练习题结果写上就可以。

## 实验七 多线程

## 一、实验目的

1. 线程的概念、线程的生命周期。
2. 多线程的编程：继承Thread类与使用Runnable接口。
3. 使用多线程机制实现动画。

## 二、实验要求

1. 掌握利用JAVA语言编写多线程程序的方法。

2. 掌握线程的调度方法。

3. 掌握多线程环境中GUI程序的编写方法。

## 实验内容

**（一）汉字打字练习。**

* **实验要求：**

编写一个Java应用程序，在主线程中再创建一个Frame类型的窗口，在该窗口中再创建1个线程giveWord。线程giveWord每隔2秒钟给出一个汉字，用户使用一种汉字输入法将该汉字输入到文本框中。

* **程序模板：**

**WordThread.java**

import java.awt.\*;

public class WordThread extends Thread

{ char word;

int k=19968;

Label com;

WordThread(Label com)

{ this.com=com;

}

public void run()

{ k=19968;

while(true)

{

word=(char)k;

com.setText(""+word);

try{ 【代码1】//调用sleep方法使得线程中断6000豪秒

}

catch(InterruptedException e){}

k++;

if(k>=29968) k=19968;

}

}

}

**ThreadFrame.java**

import java.awt.\*;

import java.awt.event.\*;

public class ThreadFrame extends Frame implements ActionListener

{

Label wordLabel;

Button button;

TextField inputText,scoreText;

【代码2】//用WordThread声明一个giveWord对象

int score=0;

ThreadFrame()

{ wordLabel=new Label(" ",Label.CENTER);

wordLabel.setFont(new Font("",Font.BOLD,72));

button=new Button("开始");

inputText=new TextField(3);

scoreText=new TextField(5);

scoreText.setEditable(false);

【代码3】//创建giveWord，将wordLabel传递给WordThread构造方法的参数

button.addActionListener(this);

inputText.addActionListener(this);

add(button,BorderLayout.NORTH);

add(wordLabel,BorderLayout.CENTER);

Panel southP=new Panel();

southP.add(new Label("输入标签所显示的汉字后回车:"));

southP.add(inputText);

southP.add(scoreText);

add(southP,BorderLayout.SOUTH);

setBounds(100,100,350,180);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

}

);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==button)

{ if(!(【代码4】)) //giveWord调用方法isAlive()

{ giveWord=new WordThread(wordLabel);

}

try

{ 【代码5】//giveWord调用方法start()

}

catch(Exception exe){}

}

else if(e.getSource()==inputText)

{ if(inputText.getText().equals(wordLabel.getText()))

{ score++;

}

scoreText.setText("得分:"+score);

inputText.setText(null);

}

}

}

**WordThread.java**

public class ThreadWordMainClass

{ public static void main(String args[])

{ new ThreadFrame();

}

* **实验后的练习：**

1. 在WordThread类中增加int型的成员time，其初值为6000，将其中的【代码1】更改为线程中断time毫秒。在WordThread类增加public void setTime(int n)方法，使得WordThread线程对象可以调用该方法修改time的值。

**（二）旋转的行星。**

* **实验要求：**

编写一个应用程序，模拟月亮围绕地球旋转、地球围绕太阳旋转。

* **程序模板：**

**Mycanvas.java**

import java.awt.\*;

public class Mycanvas extends Canvas

{ int r;

Color c;

public void setColor(Color c)

{ this.c=c;

}

public void setR(int r)

{ this.r=r;

}

public void paint(Graphics g)

{ g.setColor(c);

g.fillOval(0,0,2\*r,2\*r);

}

public int getR()

{ return r;

}

}

**Planet.java**

import java.awt.\*;

public class Planet extends Panel implements Runnable

{ 【代码1】 //用Thread类声明一个moon对象

Mycanvas yellowBall;

double pointX[]=new double[360],

pointY[]=new double[360]; //用来表示画布左上角端点坐标的数组

int w=100,h=100;

int radius=30;

Planet()

{ setSize(w,h);

setLayout(null);

yellowBall=new Mycanvas();

yellowBall.setColor(Color.yellow);

add(yellowBall);

yellowBall.setSize(12,12);

yellowBall.setR(12/2);

pointX[0]=0;

pointY[0]=-radius;

double angle=1\*Math.PI/180; //刻度为1度

for(int i=0;i<359;i++) //计算出数组中各个元素的值

{ pointX[i+1]=pointX[i]\*Math.cos(angle)-Math.sin(angle)\*pointY[i];

pointY[i+1]=pointY[i]\*Math.cos(angle)+pointX[i]\*Math.sin(angle);

}

for(int i=0;i<360;i++)

{ pointX[i]=pointX[i]+w/2; //坐标平移

pointY[i]=pointY[i]+h/2;

}

yellowBall.setLocation((int)pointX[0]-yellowBall.getR(),

(int)pointY[0]-yellowBall.getR());

【代码2】 //创建 moon线程，当前面板做为该线程的目标对象

}

public void start()

{ try{ moon .start();

}

catch(Exception exe){}

}

public void paint(Graphics g)

{ g.setColor(Color.blue);

g.fillOval(w/2-9,h/2-9,18,18);

}

public void run()

{ int i=0;

while(true)

{ i=(i+1)%360;

yellowBall.setLocation((int)pointX[i]-yellowBall.getR(),

(int)pointY[i]-yellowBall.getR());

try{ 【代码3】 // Thread类调用类方法sleep使得线程中断10豪秒

}

catch(InterruptedException e){}

}

}

}

**HaveThreadFrame.java**

import java.awt.\*;

import java.awt.event.\*;

public class HaveThreadFrame extends Frame implements Runnable

{ 【代码4】 //用Thread类声明一个rotate对象

Planet earth;

double pointX[]=new double[360],

pointY[]=new double[360];

int width,height;

int radius=120;

HaveThreadFrame()

{ rotate=new Thread(this);

earth=new Planet();

setBounds(0,0,360,400);

width=getBounds().width;

height=getBounds().height;

pointX[0]=0;

pointY[0]=-radius;

double angle=1\*Math.PI/180;

for(int i=0;i<359;i++)

{ pointX[i+1]=pointX[i]\*Math.cos(angle)-Math.sin(angle)\*pointY[i];

pointY[i+1]=pointY[i]\*Math.cos(angle)+pointX[i]\*Math.sin(angle);

}

for(int i=0;i<360;i++)

{ pointX[i]=pointX[i]+width/2;

pointY[i]=pointY[i]+height/2;

}

setLayout(null);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

}

);

add(earth);

earth.setLocation((int)pointX[0]-earth.getSize().width/2,

(int)pointY[0]-earth.getSize().height/2);

earth.start();

【代码5】 //用rotate调用start方法

}

public void run()

{ int i=0;

while(true)

{ i=(i+1)%360;

earth.setLocation((int)pointX[i]-earth.getSize().width/2,

(int)pointY[i]-earth.getSize().height/2);

try{ Thread.sleep(100);

}

catch(InterruptedException e){}

}

}

public void paint(Graphics g)

{ g.setColor(Color.red);

g.fillOval(width/2-15,height/2-15,30,30);

}

}

**HaveThreadFrame.java**

public class ThreadRotateMainClass

{ public static void main(String args[])

{ new HaveThreadFrame();

}

}

* **实验后的练习：**

1. 在Planet类中再增加一个Mycanvas对象greenBall和一个Thread对象Satellite，线程Satellite占有CPU资源期间可以让greenBall画布旋转。

**（三）双线程接力。**

* **实验要求：**

编写一个应用程序，除了主线程外，还有两个线程：first和second。first负责模拟一个红色的按钮从坐标(10，60)运动到(100，60)；second负责模拟一个绿色的按钮从坐标(100，60)运动到(200，60)。

* **程序运行效果示例：**

程序运行效果如下图所示：

* **程序模板：**

**MoveButton.java**

import java.awt.\*;

import java.awt.event.\*;

public class MoveButton extends Frame implements Runnable,ActionListener

{ 【代码1】//用Thread类声明first,second两个线程对象

Button redButton,greenButton,startButton;

int distance=10;

MoveButton()

{ 【代码2】 //创建first线程，当前窗口做为该线程的目标对象

【代码3】 //创建first线程，当前窗口做为该线程的目标对象

redButton=new Button();

greenButton=new Button();

redButton.setBackground(Color.red);

greenButton.setBackground(Color.green);

startButton=new Button("start");

startButton.addActionListener(this);

setLayout(null);

add(redButton);

redButton.setBounds(10,60,15,15);

add(greenButton);

greenButton.setBounds(100,60,15,15);

add(startButton);

startButton.setBounds(10,100,30,30);

setBounds(0,0,300,200);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

}

);

}

public void actionPerformed(ActionEvent e)

{ try{ first.start();

second.start();

}

catch(Exception exp){}

}

public void run()

{ while(true)

{ if(【代码4】) //判断当前占有CPU资源的线程是否是first

{ moveComponent(redButton);

try{ Thread.sleep(20);

}

catch(Exception exp){}

}

if(【代码5】) //判断当前占有CPU资源的线程是否是second

{ moveComponent(greenButton);

try{ Thread.sleep(10);

}

catch(Exception exp){}

}

}

}

public synchronized void moveComponent(Component b)

{

if(Thread.currentThread()==first)

{ while(distance>100&&distance<=200)

try{ wait();

}

catch(Exception exp){}

distance=distance+1;

b.setLocation(distance,60);

if(distance>=100)

{ b.setLocation(10,60);

notifyAll();

}

}

if(Thread.currentThread()==second)

{ while(distance>=10&&distance<100)

try{ wait();

}

catch(Exception exp){}

distance=distance+1;

b.setLocation(distance,60);

if(distance>200)

{ distance=10;

b.setLocation(100,60);

notifyAll();

}

}

}

}

**MoveButtonMainClass.java**

public class MoveButtonMainClass

{ public static void main(String args[])

{ new MoveButton();

}

* **实验后的练习：**

1. 在MoveButton类中再增加一个蓝色的按钮和一个third线程，third线程负责将这个蓝色的按钮从(200，60)运动到(300，60)。

【结果】

将本次上机实验结果按照实验要求内容依次进行，填写实验目标、过程、结果。

【注意事项】

①实验报告要求每人提交一份，使用专门的报告纸，内容写全，包括实验名称、班级、姓名、同组人，试验日期、交报告日期等。

②报告不需要填写源代码，只要将完成情况和所有练习题结果写上就可以。

## 实验八 输入输出流

## 一、实验目的

1. 理解数据流的概念
2. 理解Java流的层次结构
3. 理解文件的概念

## 二、实验要求

1. 掌握字符输入、输出流的用法；

2. 掌握RandomAccessFil类的使用；

* + - 1. 掌握ZipInputStream流的使用。

## 三、实验内容

**（一）学读汉字。**

* **实验要求：**

编写一个Java应用程序，要求如下：

（1）可以将一个由汉字字符组成的文本文件读入到程序中；

（2）单击名为“下一个汉字”的按钮，可以在一个标签中显示程序读入的一个汉字；

（3）单击名为“发音”的按钮，可以听到标签上显示的汉字的读音。

（4）用户可以使用文本编辑器编辑程序中用到的3个由汉字字符组成的文本文件：training1.txt、training2.txt和training.txt，这些文本文件中的汉字需要用空格、逗号或回车分隔。

（）需要自己制作相应的声音文件，比如：training1.txt文件包含汉字“你”，那么在当前应用程序的运行目录中需要有“你.wav”格式的声音文件。

（）用户选择“帮助”菜单，可以查看软件的帮助信息。

* **程序模板：**

**ChineseCharacters.java**

import java.io.\*;

import java.util.StringTokenizer;

public class ChineseCharacters

{ public StringBuffer getChinesecharacters(File file)

{ StringBuffer hanzi=new StringBuffer();

try{ FileReader inOne=【代码1】 //创建指向文件f的inOne 的对象

BufferedReader inTwo=【代码2】//创建指向文件inOne的inTwo的对象

String s=null;

int i=0;

while((s=【代码3】)!=null) //inTwo读取一行

{ StringTokenizer tokenizer=new StringTokenizer(s," ,'\n' ");

while(tokenizer.hasMoreTokens())

{ hanzi.append(tokenizer.nextToken());

}

}

}

catch(Exception e) {}

return hanzi;

}

}

**StudyFrame.java**

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import javax.sound.sampled.\*;

public class StudyFrame extends Frame implements ItemListener,ActionListener,Runnable

{ ChineseCharacters chinese;

Choice choice;

Button getCharacters,voiceCharacters;

Label showCharacters;

StringBuffer trainedChinese=null;

Clip clip=null;

Thread voiceThread;

int k=0;

Panel pCenter;

CardLayout mycard;

TextArea textHelp;

MenuBar menubar;

Menu menu;

MenuItem help;

public StudyFrame()

{ chinese=new ChineseCharacters();

choice=new Choice();

choice.add("training1.txt");

choice.add("training2.txt");

choice.add("training3.txt");

showCharacters=new Label("",Label.CENTER);

showCharacters.setFont(new Font("宋体",Font.BOLD,72));

showCharacters.setBackground(Color.green);

getCharacters=new Button("下一个汉字");

voiceCharacters=new Button("发音");

voiceThread=new Thread(this);

choice.addItemListener(this);

voiceCharacters.addActionListener(this);

getCharacters.addActionListener(this);

Panel pNorth=new Panel();

pNorth.add(new Label("选择一个汉字字符组成的文件"));

pNorth.add(choice);

add(pNorth,BorderLayout.NORTH);

Panel pSouth=new Panel();

pSouth.add(getCharacters);

pSouth.add(voiceCharacters);

add(pSouth,BorderLayout.SOUTH);

pCenter=new Panel();

mycard=new CardLayout();

pCenter.setLayout(mycard);

textHelp=new TextArea();

pCenter.add("hanzi",showCharacters);

pCenter.add("help",textHelp);

add(pCenter,BorderLayout.CENTER);

menubar=new MenuBar();

menu=new Menu("帮助");

help=new MenuItem("关于学汉字");

help.addActionListener(this);

menu.add(help);

menubar.add(menu);

setMenuBar(menubar);

setSize(350,220);

setVisible(true);

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

});

validate();

}

public void itemStateChanged(ItemEvent e)

{ String fileName=choice.getSelectedItem();

File file=new File(fileName);

trainedChinese=chinese.getChinesecharacters(file);

k=0;

mycard.show(pCenter,"hanzi") ;

}

public void actionPerformed(ActionEvent e)

{ if(e.getSource()==getCharacters)

{ if(trainedChinese!=null)

{ char c=trainedChinese.charAt(k);

k++;

if(k>=trainedChinese.length())

k=0;

showCharacters.setText(""+c);

}

else

{ showCharacters.setText("请选择一个汉字字符文件");

}

}

if(e.getSource()==voiceCharacters)

{ if(!(voiceThread.isAlive()))

{ voiceThread=new Thread(this);

}

try{ voiceThread.start();

}

catch(Exception exp){}

}

if(e.getSource()==help)

{ mycard.show(pCenter,"help") ;

try{ File helpFile=new File("help.txt");

FileReader inOne=【代码4】 //创建指向文件helpFile的inOne 的对象

BufferedReader inTwo=【代码5】 //创建指向文件inOne的inTwo 的对象

String s=null;

while((s=inTwo.readLine())!=null)

{ textHelp.append(s+"\n");

}

inOne.close();

inTwo.close();

}

catch(IOException exp){}

}

}

public void run()

{ voiceCharacters.setEnabled(false);

try{ if(clip!=null)

{ clip.close()

}

clip=AudioSystem.getClip();

File voiceFile=new File(showCharacters.getText().trim()+".wav");

clip.open(AudioSystem.getAudioInputStream(voiceFile));

}

catch(Exception exp){}

clip.start();

voiceCharacters.setEnabled(true);

}

}

**StudyMainClass.java**

public class StudyMainClass

{ public static void main(String args[])

{ new StudyFrame();

}

}

* **实验后的练习：**

1. 在StudyFrame类中增加一个按钮previousButton，单击该按钮可以读取前一个汉字。

**（二）统计英文单词。**

* **实验要求：**

使用RandomAccessFile流统计一篇英文中的单词，要求如下：

（1）一共出现了多少个单词；

（2）有多少个互不相同的单词；

（3）给出每个单词出现的频率，并将这些单词按频率大小顺序显示在一个TextArea中。

* **程序模板：**

**WordStatistic.java**

import java.io.\*;

import java.util.Vector;

public class WordStatistic

{ Vector allWorsd,noSameWord;

WordStatistic()

{ allWorsd=new Vector();

noSameWord=new Vector();

}

public void wordStatistic(File file)

{ try{ RandomAccessFile inOne=【代码1】 //创建指向文件file的inOne 的对象

RandomAccessFile inTwo=【代码2】 //创建指向文件file的inTwo 的对象

long wordStarPostion=0,wordEndPostion=0;

long length=inOne.length();

int flag=1;

int c=-1;

for(int k=0;k<=length;k++)

{ c=【代码3】 // inOne调用read()方法

boolean boo=(c<='Z'&&c>='A')||(c<='z'&&c>='a');

if(boo)

{ if(flag==1)

{ wordStarPostion=inOne.getFilePointer()-1;

flag=0;

}

}

else

{ if(flag==0)

{

if(c==-1)

wordEndPostion=inOne.getFilePointer();

else

wordEndPostion=inOne.getFilePointer()-1;

【代码4】// inTwo调用seek方法将读写位置移动到wordStarPostion

byte cc[]=new byte[(int)wordEndPostion-(int)wordStarPostion];

【代码5】// inTwo调用readFully(byte a)方法，向a传递cc

String word=new String(cc);

allWorsd.add(word);

if(!(noSameWord.contains(word)))

noSameWord.add(word);

}

flag=1;

}

}

inOne.close();

inTwo.close();

}

catch(Exception e){}

}

public Vector getAllWorsd()

{ return allWorsd;

}

public Vector getNoSameWord()

{ return noSameWord;

}

}

**RandomExample.java**

import java.awt.\*;

import java.awt.event.\*;

import java.util.Vector;

import java.io.File;

public class StatisticFrame extends Frame implements ActionListener

{ WordStatistic statistic;

TextArea showMessage;

Button openFile;

FileDialog openFileDialog;

Vector allWord,noSameWord;

public StatisticFrame()

{ statistic=new WordStatistic();

showMessage=new TextArea();

openFile=new Button("Open File");

openFile.addActionListener(this);

add(openFile,BorderLayout.NORTH);

add(showMessage,BorderLayout.CENTER);

openFileDialog=new FileDialog(this,"打开文件话框",FileDialog.LOAD);

allWord=new Vector();

noSameWord=new Vector();

setSize(350,300);

setVisible(true);

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

});

validate();

}

public void actionPerformed(ActionEvent e)

{ noSameWord.clear();

allWord.clear();

showMessage.setText(null);

openFileDialog.setVisible(true);

String fileName=openFileDialog.getFile();

if(fileName!=null)

{ statistic.wordStatistic(new File(fileName));

allWord=statistic.getAllWorsd();

noSameWord=statistic.getNoSameWord();

showMessage.append("\n"+fileName+"中有"+allWord.size()+"个英文单词");

showMessage.append("\n其中有"+noSameWord.size()+"个互不相同英文单词");

showMessage.append("\n按使用频率排列：\n");

int count[]=new int[noSameWord.size()];

for(int i=0;i<noSameWord.size();i++)

{ String s1=(String)noSameWord.elementAt(i);

for(int j=0;j<allWord.size();j++)

{ String s2=(String)allWord.elementAt(j);

if(s1.equals(s2))

count[i]++;

}

}

for(int m=0;m<noSameWord.size();m++)

{ for(int n=m+1;n<noSameWord.size();n++)

{ if(count[n]>count[m])

{ String temp=(String)noSameWord.elementAt(m);

noSameWord.setElementAt((String)noSameWord.elementAt(n),m);

noSameWord.setElementAt(temp,n);

int t=count[m];

count[m]=count[n];

count[n]=t;

}

}

}

for(int m=0;m<noSameWord.size();m++)

{ showMessage.append("\n"+(String)noSameWord.elementAt(m)+

":"+count[m]+"/"+allWord.size()+

"="+(1.0\*count[m])/allWord.size());

}

}

}

}

**RandomExample.java**

public class StatisticMainClass

{ public static void main(String args[])

{ new StatisticFrame();

}

}

* **实验后的练习：**

1. 在StatisticFrame的showMessage中增加单词按字典序排序输出的信息。

**（三）读取Zip文件。**

* **实验要求：**

读取book.zip，并将book.zip中含有的文件重新存放到当前目录中的book文件夹中，即将book.zip的内容解压到book文件夹中。

* **程序模板：**

**ReadZipFile.java**

import java.io.\*;

import java.util.zip.\*;

public class ReadZipFile

{ public static void main(String args[])

{ File f=new File("book.zip");

File dir=new File("Book");

byte b[]=new byte[100];

dir.mkdir();

try

{ ZipInputStream in=new ZipInputStream(new FileInputStream(f));

ZipEntry zipEntry=null;

while((zipEntry=in.getNextEntry())!=null)

{ File file=new File(dir,zipEntry.getName());

FileOutputStream out=new FileOutputStream(file);

int n=-1;

System.out.println(file.getAbsolutePath()+"的内容：");

while((n=in.read(b,0,100))!=-1)

{ String str=new String(b,0,n);

System.out.println(str);

out.write(b,0,n);

}

out.close();

}

in.close();

}

catch(IOException ee)

{

System.out.println(ee);

}

}

}

* **实验后的练习：**

1. 编写一个GUI程序，提供1个对话框，用户可以使用这个对话框选择要解压缩的zip文件，设置解压后所得到的文件的存放目录。

【结果】

将本次上机实验结果按照实验要求内容依次进行，填写实验目标、过程、结果。

【注意事项】

①实验报告要求每人提交一份，使用专门的报告纸，内容写全，包括实验名称、班级、姓名、同组人，试验日期、交报告日期等。

②报告不需要填写源代码，只要将完成情况和所有练习题结果写上就可以。