Java面向对象程序设计

（编著 耿祥义 张跃平）

习题1

1．James Gosling

2．

（1）使用一个文本编辑器编写源文件。

（2）使用Java编译器（javac.exe）编译Java源程序，得到字节码文件。

（3）使用Java解释器（java.exe）运行Java程序

3．Java的源文件是由若干个书写形式互相独立的类组成的。

应用程序中可以没有public类，若有的话至多可以有一个public类。

4．系统环境path D\jdk\bin;

系统环境classpath D\jdk\jre\lib\rt.jar;.;

5． B

6． Java源文件的扩展名是.java。Java字节码的扩展名是.class。

7． D

8．（1）Speak.java

（2）生成两个字节码文件，这些字节码文件的名字Speak.class 和 Xiti8.class

（3）java Xiti8

（4）执行java Speak的错误提示

Exception in thread "main" java.lang.NoSuchMethodError: main

执行java xiti8得到的错误提示

Exception in thread "main" java.lang.NoClassDefFoundError: xiti8 (wrong name: Xiti8)

执行java Xiti8.class得到的错误提示

Exception in thread "main" java.lang.NoClassDefFoundError: Xiti8/class

执行java Xiti8得到的输出结果

I'm glad to meet you

9．属于操作题，解答略。

习题2

1． D

2．【代码1】 【代码2】 错误 //【代码3】更正为 float z=6.89F;

3．float型常量后面必须要有后缀“f”或“F”。

对于double常量，后面可以有后缀“d”或“D”，但允许省略该后缀。

4．public class Xiti4{

public static void main (String args[ ]){

char ch1='你',ch2='我',ch3='他';

System.out.println("\""+ch1+"\"的位置:"+(int)ch1);

System.out.println("\""+ch2+"\"的位置:"+(int)ch2);

System.out.println("\""+ch3+"\"的位置:"+(int)ch3);

}

}

5．数组名字.length

6．数组名字.length

7． 【代码1】A,65

【代码2】-127

【代码3】 123456.783,123456.78312

8．

【代码1】false

【代码2】true

【代码3】false

【代码4】3

【代码5】4.4

【代码6】8.8

习题3

输出110

if-else语句书写的不够规范，复合语句缺少大括号“{}”，代码不够清晰。

2．你好好酷!!

3．

public class Xiti3\_3

{

public static void main (String args[ ]){

int startPosition=0,endPosition=0;

char cStart='а',cEnd='я';

startPosition=(int)cStart; //cStart做int型转换据运算，并将结果赋值给startPosition

endPosition=(int)cEnd ; //cEnd做int型转换运算，并将结果赋值给endPosition

System.out.println("俄文字母表：");

for(int i=startPosition;i<=endPosition;i++){

char c='\0';

c=(char)i; //i做char型转换运算，并将结果赋值给c

System.out.print(" "+c);

if((i-startPosition+1)%10==0)

System.out.println("");

}

}

}

4．

public class Xiti4

{ public static void main(String args[])

{ double sum=0,a=1;

int i=1;

while(i<=20)

{ sum=sum+a;

i++;

a=a\*i;

}

System.out.println("sum="+sum);

}

}

5．

class Xiti5

{ public static void main(String args[])

{ int i,j;

for(j=2;j<=100;j++)

{ for(i=2;i<=j/2;i++)

{ if(j%i==0)

break;

}

if(i>j/2)

{ System.out.print(" "+j);

}

}

}

}

6．

class Xiti6

{ public static void main(String args[])

{ double sum=0,a=1,i=1;

while(i<=20)

{ sum=sum+a;

i++;

a=(1.0/i)\*a;

}

System.out.println("使用while循环计算的sum="+sum);

for(sum=0,i=1,a=1;i<=20;i++)

{ a=a\*(1.0/i);

sum=sum+a;

}

System.out.println("使用for循环计算的sum="+sum);

}

}

7．

public class Xiti7

{ public static void main(String args[])

{ int sum=0,i,j;

for(i=1;i<=1000;i++)

{ for(j=1,sum=0;j<i;j++)

{ if(i%j==0)

sum=sum+j;

}

if(sum==i)

System.out.println("完数:"+i);

}

}

}

8．方法之一

import java.util.Scanner;

public class Xiti8

{ public static void main (String args[ ]){

System.out.println("请输入两个非零正整数，每输入一个数回车确认");

Scanner reader=new Scanner(System.in);

int m=0,n=0,temp=0,gy=0,gb=0,a,b;

a=m = reader.nextInt();

b=n = reader.nextInt();

if(m<n)

{ temp=m;

m=n;

n=temp;

}

int r=m%n;

while(r!=0)

{ n=m;

m=r;

r=m%n;

}

gy=n;

gb=a\*b/gy;

System.out.println("最大公约数 ："+gy);

System.out.println("最小公倍数 ："+gb);

}

}

8．方法之二

import java.util.Scanner;

public class Xiti8 {

public static void main (String args[ ]){

System.out.println("请输入两个非零正整数，每输入一个数回车确认");

Scanner reader=new Scanner(System.in);

int m=0,n=0,t=0,gy=0,gb=0;

m = reader.nextInt();

n = reader.nextInt();

if(m>n){

t=m;

m=n;

n=t;

}

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

if(m%i==0 && n%i==0){

gy=i;

}

}

gb=m\*n/gy;

System.out.println(m+","+n+"的最大公约数为 "+gy);

System.out.println(m+","+n+"的最小公倍数为 "+gb);

}

}

9．

public class Xiti9

{ public static void main(String args[])

{ int n=1;

long sum=0,t=1;

t=n\*t;

while(true)

{ sum=sum+t;

if(sum>9999)

break;

n++;

t=n\*t;

}

System.out.println("满足条件的最大整数:"+(n-1));

}

}// 1至7的阶乘和是sum=5913.0 // 1至8的阶乘和是sum=46233.0

习题4

1．用该类创建对象时。

2．所谓方法重载是在一个类中可以有多个方法具有相同的名字，但这些方法的参数必须不同，即或者是参数的个数不同，或者是参数的类型不同。构造方法可以重载。

3． 可以。不可以。

4．不可以。

5．成员变量又分为实例变量和类变量，用static修饰的变量是类变量。那么类变量和实例变量有什么区别呢？一个类通过使用new运算符可以创建多个不同的对象，不同的对象的实例变量将被分配不同的内存空间；如果类中的成员变量有类变量，那么所有对象的这个类变量都分配给相同的一处内存，改变其中一个对象的这个类变量会影响其它对象的这个类变量。也就是说对象共享类变量。

6．C，D

7．【代码1】，【代码4】

8．sum=-100

9．27

10. \*\*20

##100

习题5

1． 如果子类和父类在同一个包中，那么子类自然地继承了其父类中不是private的成员变量作为自己的成员变量，并且也自然地继承了父类中不是private的方法作为自己的方法。继承的成员或方法的访问权限保持不变。如果子类和父类不在同一个包中，那么子类继承了父类的protected、public成员变量做为子类的成员变量，并且继承了父类的protected、public方法为子类的方法，继承的成员或方法的访问权限保持不变。如果子类和父类不在同一个包里，子类不能继承父类的友好变量和友好方法。

只要子类中声明的成员变量和父类中的成员变量同名时，子类就隐藏了继承的成员变量。

子类中定义一个方法，这个方法的类型和父类的方法的类型一致或者是父类的方法的类型的子类型，并且这个方法的名字、参数个数、参数的类型和父类的方法完全相同，子类如此定义的方法称作子类重写的方法。子类通过方法的重写可以隐藏继承的方法。

2．不可以。

3．abstract类。

4．假设B类是A类子类或间接子类，当我们用子类B创建一个对象，并把这个对象的引用放到A类的对象中时，称这个A类对象是子类对象的上转型对象。

5．可以把实现某一接口的类创建的对象的引用赋给该接口声明的接口变量中。那么该接口变量就可以调用被类实现的接口中的方法。

6．A，C，D

7．15.0

8.0

8．98.0

12

9．

class A

{ public final void f()

{ char cStart='a',cEnd='z';

for(char c=cStart;c<=cEnd;c++)

{ System.out.print(" "+c);

}

}

}

class B extends A

{ public void g()

{ char cStart='α',cEnd='ω';

for(char c=cStart;c<=cEnd;c++)

{ System.out.print(" "+c);

}

}

}

public class Xiti9

{ public static void main (String args[ ])

{ B b=new B();

b.f();

b.g();

}

}

10．

class A

{ public int f(int a,int b){

if(b<a)

{ int temp=0;

temp=a;

a=b;

b=temp;

}

int r=b%a;

while(r!=0)

{ b=a;

a=r;

r=b%a;

}

return a;

}

}

class B extends A

{ public int f(int a,int b)

{ int division=super.f(a,b);

return (a\*b)/division;

}

}

public class Xiti10

{ public static void main (String args[ ])

{ A a=new A();

B b=new B();

System.out.println("最大公约数 ："+a.f(36,24));

System.out.println("最小公倍数 ："+b.f(36,24));

}

}

习题6

1．仍然有效。

2．可以。

3．不可以。

4．大家好，祝工作顺利！

5． 96

乘数超过99

习题7

1．

（1）Strategy是接口。

（2）Army不是抽象类。

（3）Army和Strategy是关联关系。

（4）StrategyA, StrategyB、StrategyC与Strategy是实现关系。

2．

3．

4．例子13的设计符合开-闭原则。

5．例子17的设计符合开-闭原则。

习题8

采用新增的策略为选手计算得分。

增加新的具体策略StrategyFour。StrategyFour类将double computeScore(double [] a)方法实现为去掉数组a的元素中的一个最大值和一个最小值，然后计算剩余元素的几何平均值。

import java.util.Arrays;

public class StrategyFour implements ComputableStrategy {

public double computeScore(double [] a) {

if(a.length<=2)

return 0;

double score=0,multi=1;

Arrays.sort(a);

int n=a.length-2;

for(int i=1;i<a.length-1;i++) {

multi=multi\*a[i];

}

score=Math.pow(multi,1.0/n);

return score;

}

}

2．

(1)策略（Strategy）PrintCharacter.java

public interface PrintCharacter{

public abstract void printTable(char [] a,char[] b);

}

(2) 具体策略

PrintStrategyOne.java

public class PrintStrategyOne implements PrintCharacter {

public void printTable(char [] a,char[] b) {

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

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

}

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

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

}

System.out.println("");

}

}

PrintStrategyTwo.java

public class PrintStrategyTwo implements PrintCharacter {

public void printTable(char [] a,char[] b) {

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

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

}

}

}

（3）上下文 PrintGame.java

public class PrintGame {

PrintCharacter strategy;

public void setStrategy(PrintCharacter strategy) {

this.strategy=strategy;

}

public void getPersonScore(char[] a,char[] b){

if(strategy==null)

System.out.println("sorry!");

else

strategy.printTable(a,b);

}

}

应用以上策略:

public class Application {

public static void main(String args[]) {

char [] a=new char[26];

char [] b=new char[26];

for(int i=0;i<=25;i++){

a[i]=(char)('a'+i);

}

for(int i=0;i<=25;i++){

b[i]=(char)('A'+i);

}

PrintGame game=new PrintGame(); //上下文对象

game.setStrategy(new PrintStrategyOne()); //上下文对象使用策略一

System.out.println("方案1:");

game.getPersonScore(a,b);

game.setStrategy(new PrintStrategyTwo()); //上下文对象使用策略二

System.out.println("方案2:");

game.getPersonScore(a,b);

}

}

3．参照本章8.3.3自主完成。

习题9

1．A，B，D

2． Love:Game

3．13

abc夏日

4．13579

5．9javaHello

6．

public class Xiti6 {

public static void main (String args[ ]) {

String s1,s2,s3,t1="ABCDabcd";

System.out.println("字符串原来是这个样子: "+t1);

s1=t1.toUpperCase();

System.out.println("字符串中的小写字母变成大写是这个样子: "+s1);

s2=t1.toLowerCase();

System.out.println("字符串中的大写字母变成小写是这个样子: "+s2);

s3=s1.concat(s2);

System.out.println("大写字符串连接小写字符串是这个样子: "+s3);

}

}

7．

class Xiti7

{ public static void main(String args[ ])

{ String s ="中华人民共和国";

char a=s.charAt(0);

char b=s.charAt(6);

System.out.println("第一个字符: "+a);

System.out.println("最后一个字符: "+b);

}

}

8．

import java.util.\*;

class Xiti8

{ public static void main(String args[]){

int year,month;

System.out.println("请输入年份和月份，每输入一个数回车确认");

Scanner reader=new Scanner(System.in);

year= reader.nextInt();

month= reader.nextInt();

String [] day=new String[42];

System.out.println(" 日 一 二 三 四 五 六");

Calendar rili=Calendar.getInstance();

rili.set(year,month-1,1);//将日历翻到year年month月1日,注意0表示一月...11表示十二月

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

int dayAmount=0;

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

dayAmount=31;

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

dayAmount=30;

if(month==2)

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

dayAmount=29;

else

dayAmount=28;

for(int i=0;i<星期几;i++)

day[i]="";

for(int i=星期几,n=1;i<星期几+dayAmount;i++){

if(n<=9)

day[i]=String.valueOf(n)+" " ;

else

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

n++;

}

for(int i=星期几+dayAmount;i<42;i++)

day[i]="";

for(int i=0;i<星期几;i++)

{ day[i]="\*\*";

}

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

{ if(i%7==0)

{ System.out.println("");

}

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

}

}

}

9．

import java.util.\*;

class Xiti9

{ public static void main(String args[]){

int year1,month1,day1,year2,month2,day2;

Scanner reader=new Scanner(System.in);

System.out.println("请输入第一个日期的年份 月份 日期 ，每输入一个数回车确认");

year1= reader.nextInt();

month1= reader.nextInt();

day1= reader.nextInt();

System.out.println("请输入第二个日期的年份 月份 日期 ，每输入一个数回车确认");

year2= reader.nextInt();

month2= reader.nextInt();

day2= reader.nextInt();

Calendar calendar=Calendar.getInstance();

calendar.set(year1,month1,day1);

long timeYear1=calendar.getTimeInMillis();

calendar.set(year2,month2,day2);

long timeYear2=calendar.getTimeInMillis();

long 相隔天数=Math.abs((timeYear1-timeYear2)/(1000\*60\*60\*24));

System.out.println(""+year1+"年"+month1+"月"+day1+"日和"+

year2+"年"+month2+"月"+day2+"日相隔"+相隔天数+"天");

}

}

10．

public class Xiti10

{ public static void main(String args[])

{ double a=0,b=0,c=0;

a=12;

b=24;

c=Math.max(a,b);

System.out.println(c);

c=Math.min(a,b);

System.out.println(c);

c=Math.pow(2,3);

System.out.println(c);

c=Math.abs(-0.123);

System.out.println(c);

c=Math.asin(0.56);

System.out.println(c);

c=Math.cos(3.14);

System.out.println(c);

c=Math.exp(1);

System.out.println(c);

c=Math.log(8);

System.out.println(c);

}

}

习题10

1．BorderLayout布局。

2．不可以。

3．A,C。

4．

import java.util.StringTokenizer;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Xiti4

{ public static void main(String args[])

{ ComputerFrame fr=new ComputerFrame();

fr.setTitle("计算的窗口");

}

}

class ComputerFrame extends JFrame implements TextListener

{ TextArea text1,text2;

int count=1;

double sum=0,aver=0;

public ComputerFrame()

{ setLayout(new FlowLayout());

text1=new TextArea(6,20);

text2=new TextArea(6,20);

add(text1);

add(text2);

text2.setEditable(false);

text1.addTextListener(this);

setSize(300,320);

setVisible(true);

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

});

validate();

}

public void textValueChanged(TextEvent e)

{ String s=text1.getText();

sum=0;

aver=0;

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

int n=fenxi.countTokens();

count=n;

double a[]=new double[n];

for(int i=0;i<=n-1;i++)

{ String temp=fenxi.nextToken();

try { a[i]=Double.parseDouble(temp);

sum=sum+a[i];

}

catch(Exception ee)

{ count--;

}

}

aver=sum/count;

text2.setText(null);

text2.append("\n和:"+sum);

text2.append("\n平均值:"+aver);

}

}

5．

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Xiti5

{ public static void main(String args[])

{ ComputerFrame fr=new ComputerFrame();

fr.setTitle("计算");

}

}

class ComputerFrame extends Frame implements ActionListener

{ TextField text1,text2,text3;

Button button1,button2,button3,button4;

Label label;

public ComputerFrame()

{setLayout(new FlowLayout());

text1=new TextField(10);

text2=new TextField(10);

text3=new TextField(10);

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

label.setBackground(Color.green);

add(text1);

add(label);

add(text2);

add(text3);

button1=new Button("加");

button2=new Button("减");

button3=new Button("乘");

button4=new Button("除");

add(button1);

add(button2);

add(button3);

add(button4);

button1.addActionListener(this);

button2.addActionListener(this);

button3.addActionListener(this);

button4.addActionListener(this);

setSize(300,320);

setVisible(true);

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

});

validate();

}

public void actionPerformed(ActionEvent e)

{ double n;

if(e.getSource()==button1)

{ double n1,n2;

try{ n1=Double.parseDouble(text1.getText());

n2=Double.parseDouble(text2.getText());

n=n1+n2;

text3.setText(String.valueOf(n));

label.setText("+");

}

catch(NumberFormatException ee)

{ text3.setText("请输入数字字符");

}

}

else if(e.getSource()==button2)

{ double n1,n2;

try{ n1=Double.parseDouble(text1.getText());

n2=Double.parseDouble(text2.getText());

n=n1-n2;

text3.setText(String.valueOf(n));

label.setText("-");

}

catch(NumberFormatException ee)

{ text3.setText("请输入数字字符");

}

}

else if(e.getSource()==button3)

{double n1,n2;

try{ n1=Double.parseDouble(text1.getText());

n2=Double.parseDouble(text2.getText());

n=n1\*n2;

text3.setText(String.valueOf(n));

label.setText("\*");

}

catch(NumberFormatException ee)

{ text3.setText("请输入数字字符");

}

}

else if(e.getSource()==button4)

{double n1,n2;

try{ n1=Double.parseDouble(text1.getText());

n2=Double.parseDouble(text2.getText());

n=n1/n2;

text3.setText(String.valueOf(n));

label.setText("/");

}

catch(NumberFormatException ee)

{ text3.setText("请输入数字字符");

}

}

validate();

}

}

6．

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Xiti6

{ public static void main(String args[])

{ new WindowPanel();

}

}

class Mypanel extends JPanel implements ActionListener

{ Button button;

TextField text;

Mypanel()

{ button=new Button(" ");

text=new TextField(12);

add(button);

add(text);

button.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{ String name=text.getText();

if(name.length()>0)

button.setLabel(name);

validate();

}

}

class WindowPanel extends Frame

{ Mypanel panel1,panel2;

WindowPanel()

{ panel1=new Mypanel();

panel2=new Mypanel();

panel1.setBackground(Color.red);

panel2.setBackground(Color.blue);

add(panel1,BorderLayout.SOUTH);

add(panel2,BorderLayout.NORTH);

setSize(300,320);

setVisible(true);

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

});

validate();

}

}

7．参见10.13, 参照本章例子10.21。

8．

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Xiti8

{ public static void main(String args[])

{ MoveFrame f=new MoveFrame();

f.setBounds(12,12,300,300);

f.setVisible(true);

f.setTitle("移动");

f.validate();

f. addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

}

);

}

}

class MoveFrame extends JFrame implements ActionListener

{ JButton controlButton,movedButton;

public MoveFrame()

{ controlButton=new JButton("单击我运动另一个按钮");

controlButton.addActionListener(this);

movedButton=new JButton();

movedButton.setBackground(new Color(12,200,34));

setLayout(null);

add(controlButton);

add(movedButton);

controlButton.setBounds(10,30,130,30);

movedButton.setBounds(100,100,10,10);

}

public void actionPerformed(ActionEvent e)

{ int x=movedButton.getBounds().x;

int y=movedButton.getBounds().y;

x=x+5;

y=y+1;

movedButton.setLocation(x,y);

if(x>200)

{ x=100;

y=100;

}

}

}

9．

import java.awt.\*;

import java.awt.event.\*;

public class Xiti9

{ public static void main(String args[])

{ Win win=new Win();

}

}

class Win extends Frame implements KeyListener

{ Button b[]=new Button[8];

int x,y;

Win()

{ setLayout(new FlowLayout());

for(int i=0;i<8;i++)

{ b[i]=new Button(""+i);

b[i].addKeyListener(this);

add(b[i]);

}

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e){

System.exit(0);

}

});

setBounds(10,10,300,300);

setVisible(true);

validate();

}

public void keyPressed(KeyEvent e)

{ int moveDistance=1;

Component com=(Component)e.getSource();

int x=(int)com.getBounds().x;

int y=(int)com.getBounds().y;

Component component[]=this.getComponents();

if(e.getKeyCode()==KeyEvent.VK\_UP)

{ y=y-moveDistance;

com.setLocation(x,y);

Rectangle comRect=com.getBounds();

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

{ Rectangle orthRect=component[k].getBounds();

if(comRect.intersects(orthRect)&&com!=component[k])

{ y=y+moveDistance;

com.setLocation(x,y);

break;

}

}

if(y<=0) y=10;

}

else if(e.getKeyCode()==KeyEvent.VK\_DOWN)

{ y=y+moveDistance;

com.setLocation(x,y);

Rectangle comRect=com.getBounds();

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

{ Rectangle orthRect=component[k].getBounds();

if(comRect.intersects(orthRect)&&com!=component[k])

{ y=y-moveDistance;

com.setLocation(x,y);

break;

}

}

if(y>=300) y=300;

}

else if(e.getKeyCode()==KeyEvent.VK\_LEFT)

{ x=x-moveDistance;

com.setLocation(x,y);

Rectangle comRect=com.getBounds();

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

{ Rectangle orthRect=component[k].getBounds();

if(comRect.intersects(orthRect)&&com!=component[k])

{ x=x+moveDistance;

com.setLocation(x,y);

break;

}

}

if(x<=0) x=0;

}

else if(e.getKeyCode()==KeyEvent.VK\_RIGHT)

{ x=x+moveDistance;

com.setLocation(x,y);

Rectangle comRect=com.getBounds();

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

{ Rectangle orthRect=component[k].getBounds();

if(comRect.intersects(orthRect)&&com!=component[k])

{ x=x-moveDistance;

com.setLocation(x,y);

break;

}

}

if(x>=300) x=300;

}

}

public void keyTyped(KeyEvent e) {}

public void keyReleased(KeyEvent e) {}

}

习题11

1．A

2．

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

class Dwindow extends Frame implements ActionListener

{ TextField inputNumber;

TextArea save;

Dwindow(String s)

{ super(s);

inputNumber=new TextField(22);

inputNumber.addActionListener(this);

save=new TextArea(12,16);

setLayout(new FlowLayout());

add(inputNumber);

add(save);

setBounds(60,60,300,300);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

}

);

}

public void actionPerformed(ActionEvent event)

{ String s=inputNumber.getText();

double n=0;

try{ n=Double.parseDouble(s);

if(n>1000)

{ int select=JOptionPane.showConfirmDialog(this,"已经超过1000确认正确吗？","确认对话框",

JOptionPane.YES\_NO\_OPTION );

if(select==JOptionPane.YES\_OPTION)

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

}

else

{ inputNumber.setText(null);

}

}

else

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

}

}

catch(NumberFormatException e)

{ JOptionPane.showMessageDialog(this,"您输入了非法字符","警告对话框",

JOptionPane.WARNING\_MESSAGE);

inputNumber.setText(null);

}

}

}

public class E

{ public static void main(String args[])

{ new Dwindow("带对话框的窗口");

}

}

3．参照以下例子完成

Xiti3.java

public class Xiti3 {

public static void main(String args[]) {

WindowColor win=new WindowColor();

win.setTitle("带颜色对话框的窗口");

}

}

WindowColor.java

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

public class WindowColor extends JFrame implements ActionListener {

JButton button;

WindowColor() {

button=new JButton("打开颜色对话框");

button.addActionListener(this);

setLayout(new FlowLayout());

add(button);

setBounds(60,60,300,300);

setVisible(true);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e) {

Color newColor=JColorChooser.showDialog(this,"调色板",button.getBackground());

if(newColor!=null) {

button.setBackground(newColor);

}

}

}

习题12

1．使用FileInputStream流。

2．FileInputStream按字节读取文件，FileReader按字符读取文件。

3．不能。

4．使用对象流写入或读入对象时，要保证对象是序列化的。

5．使用对象流很容易得获取一个序列化对象的克隆，只需将该对象写入到对象输出流，那么用对象输入流读回的对象一定是原对象的一个克隆。

6．

import java.io.\*;

public class Xiti6

{ public static void main(String args[])

{ File f=new File("E.java");;

try{ RandomAccessFile random=new RandomAccessFile(f,"rw");

random.seek(0);

long m=random.length();

while(m>=0)

{ m=m-1;

random.seek(m);

int c=random.readByte();

if(c<=255&&c>=0)

{ System.out.print((char)c);

}

else

{ m=m-1;

random.seek(m);

byte cc[]=new byte[2];

random.readFully(cc);

System.out.print(new String(cc));

}

}

}

catch(Exception exp){}

}

}

7．

import java.io.\*;

public class Xiti7

{ public static void main(String args[ ])

{ File file=new File("E.java");

File tempFile=new File("temp.txt");

try{ FileReader inOne=new FileReader(file);

BufferedReader inTwo= new BufferedReader(inOne);

FileWriter tofile=new FileWriter(tempFile);

BufferedWriter out= new BufferedWriter(tofile);

String s=null;

int i=0;

s=inTwo.readLine();

while(s!=null)

{ i++;

out.write(i+" "+s);

out.newLine();

s=inTwo.readLine();

}

inOne.close();

inTwo.close();

out.flush();

out.close();

tofile.close();

}

catch(IOException e)

{ System.out.println(e);

}

}

}

8．属于操作题目，解答略。

9．

import java.util.\*;

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Xiti9

{ public static void main(String args[])

{ EWindow w=new EWindow();

w.validate();

}

}

class EWindow extends Frame implements ActionListener,ItemListener

{ String str[]=new String[7],s;

FileReader file;

BufferedReader in;

Button start,next;

Checkbox checkbox[];

TextField 题目,分数;

int score=0;

CheckboxGroup age=new CheckboxGroup();

EWindow()

{ super("英语单词学习");

分数=new TextField(10);题目=new TextField(70);

start=new Button("重新练习");

start.addActionListener(this);

next=new Button("下一题目");

next.addActionListener(this);

checkbox=new Checkbox[4];

for(int i=0;i<=3;i++)

{ checkbox[i]=new Checkbox("",false,age);

checkbox[i].addItemListener(this);

}

try { file=new FileReader("English.txt");

in=new BufferedReader(file);

}

catch(IOException e){}

setBounds(20,100,660,300); setVisible(true);

Box box=Box.createVerticalBox();

Panel p1=new Panel(),p2=new Panel(),

p3=new Panel() ,p4=new Panel(),p5=new Panel();

p1.add(new Label("题目:"));p1.add(题目);

p2.add(new Label("选择答案："));

for(int i=0;i<=3;i++)

{ p2.add(checkbox[i]);

}

p3.add(new Label("您的得分:"));p3.add(分数);

p4.add(start); p4.add(next);

box.add(p1);box.add(p2);box.add(p3);box.add(p4);

addWindowListener(new WindowAdapter()

{public void windowClosing(WindowEvent e)

{ System.exit(0);

}

});

add(box,BorderLayout.CENTER);

reading();

}

public void reading()

{ int i=0;

try { s=in.readLine();

if(!(s.startsWith("endend")))

{ StringTokenizer tokenizer=new StringTokenizer(s,"#");

while(tokenizer.hasMoreTokens())

{ str[i]=tokenizer.nextToken();

i++;

}

题目.setText(str[0]);

for(int j=1;j<=4;j++)

{ checkbox[j-1].setLabel(str[j]);

}

}

else if(s.startsWith("endend"))

{ 题目.setText("学习完毕");

for(int j=0;j<4;j++)

{ checkbox[j].setLabel("end");

in.close();file.close();

}

}

}

catch(Exception exp){ 题目.setText("无试题文件") ; }

}

public void actionPerformed(ActionEvent event)

{ if(event.getSource()==start)

{ score=0;

分数.setText("得分： "+score);

try { file=new FileReader("English.txt");

in=new BufferedReader(file);

}

catch(IOException e){}

reading();

}

if(event.getSource()==next)

{ reading();

for(int j=0;j<4;j++)

{ checkbox[j].setEnabled(true);

}

}

}

public void itemStateChanged(ItemEvent e)

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

{ if(checkbox[j].getLabel().equals(str[5])&&checkbox[j].getState())

{ score++;

分数.setText("得分： "+score);

}

checkbox[j].setEnabled(false);

}

}

}

习题13

1．一个使用链式结构，一个使用顺序结构。

2．8。

3．ABCD。

4．选用HashMap<K,V>来存储。

5．

import java.util.\*;

class UFlashKey implements Comparable {

double d=0;

UFlashKey (double d) {

this.d=d;

}

public int compareTo(Object b) {

UFlashKey st=(UFlashKey)b;

if((this.d-st.d)==0)

return -1;

else

return (int)((this.d-st.d)\*1000);

}

}

class UFlash {

String name=null;

double capacity,price;

UFlash(String s,double m,double e) {

name=s;

capacity=m;

price=e;

}

}

public class Xiti5 {

public static void main(String args[ ]) {

TreeMap<UFlashKey,UFlash> treemap= new TreeMap<UFlashKey,UFlash>();

String str[]={"U1","U2","U3","U4","U5","U6","U7","U8","U9","U10"};

double capacity[]={1,2,2,4,0.5,10,8,4,4,2};

double price[]={30,66,90,56,50,149,120,80,85,65};

UFlash UFlash[]=new UFlash[10];

for(int k=0;k<UFlash.length;k++) {

UFlash[k]=new UFlash(str[k],capacity[k],price[k]);

}

UFlashKey key[]=new UFlashKey[10] ;

for(int k=0;k<key.length;k++) {

key[k]=new UFlashKey(UFlash[k].capacity); //关键字按容量成绩排列大小

}

for(int k=0;k<UFlash.length;k++) {

treemap.put(key[k],UFlash[k]);

}

int number=treemap.size();

System.out.println("树映射中有"+number+"个对象,按容量成绩排序:");

Collection<UFlash> collection=treemap.values();

Iterator<UFlash> iter=collection.iterator();

while(iter.hasNext()) {

UFlash stu=iter.next();

System.out.println("U盘 "+stu.name+" 容量 "+stu.capacity);

}

treemap.clear();

for(int k=0;k<key.length;k++) {

key[k]=new UFlashKey(UFlash[k].price);//关键字按价格成绩排列大小

}

for(int k=0;k<UFlash.length;k++) {

treemap.put(key[k],UFlash[k]);

}

number=treemap.size();

System.out.println("树映射中有"+number+"个对象:按价格成绩排序:");

collection=treemap.values();

iter=collection.iterator();

while(iter.hasNext()) {

UFlash stu=(UFlash)iter.next();

System.out.println("U盘 "+stu.name+" 价格 "+stu.price);

}

}

}

习题14

1．

(1)创建数据源

选择“控制面板”→“管理工具”→“ODBC数据源”（某些window/xp系统，需选择“控制面板”→“性能和维护”→“管理工具”→“ODBC数据源”）。双击ODBC数据源图标，选择“系统DSN”或“用户DSN”，单击“添加”按钮，可以创建新的数据源。

(2) 数据源选择驱动程序

选择单击“添加”按钮，出现为新增的数据源选择驱动程序界面，如果要访问Access数据库，选择Microsoft Acess Driver（\*.mdb）。单击完成按钮。

(3) 数据源名称及对应数据库的所在位置

在设置数据源具体项目的对话框，在名称栏里为数据源起一个自己喜欢的名字。这个数据源就是指某个数据库。在“数据库选择”栏中选择一个已经准备好的数据库。

2．参照本章例子14.2。

3．参照本章例子14.3。

4．参照本章例子14.4。

5．使用预处理语句不仅减轻了数据库的负担，而且也提高了访问数据库的速度。

6．事务由一组SQL语句组成，所谓事务处理是指：应用程序保证事务中的SQL语句要么全部都执行，要么一个都不执行。步骤:

(1)使用setAutoCommit(boolean autoCommit)方法

con对象首先调用setAutoCommit(boolean autoCommit)方法，将参数autoCommit取值false来关闭默认设置：

con.setAutoCommit(false);

(2) 使用commit()方法。con调用commit()方法就是让事务中的SQL语句全部生效。

(3) 使用rollback()方法。con调用rollback()方法撤消事务中成功执行过的SQL语句对数据库数据所做的更新、插入或删除操作，即撤消引起数据发生变化的SQL语句操作，将数据库中的数据恢复到commi()方法执行之前的状态。

7．参照本章例子14.2。

习题15

1．4种状态：新建、运行、中断和死亡。

2．有4种原因的中断：

JVM将CPU资源从当前线程切换给其他线程，使本线程让出CPU的使用权处于中断状态。

线程使用CPU资源期间，执行了sleep(int millsecond)方法，使当前线程进入休眠状态。经过参数millsecond指定的豪秒数之后，该线程就重新进到线程队列中排队等待CPU资源，以便从中断处继续运行。

线程使用CPU资源期间，执行了wait()方法，使得当前线程进入等待状态。等待状态的线程不会主动进到线程队列中排队等待CPU资源，必须由其他线程调用notify()方法通知它，使得它重新进到线程队列中排队等待CPU资源，以便从中断处继续运行。

线程使用CPU资源期间，执行某个操作进入阻塞状态，比如执行读/写操作引起阻塞。进入阻塞状态时线程不能进入排队队列，只有当引起阻塞的原因消除时，线程才重新进到线程队列中排队等待CPU资源，以便从原来中断处开始继续运行。

3．死亡状态，不能再调用start()方法。

4．新建和死亡状态。

5．两种方法：用Thread类或其子类。

6．使用 setPrority(int grade)方法。

7．Java使我们可以创建多个线程，在处理多线程问题时，我们必须注意这样一个问题：当两个或多个线程同时访问同一个变量，并且一个线程需要修改这个变量。我们应对这样的问题作出处理，否则可能发生混乱。

8．当一个线程使用的同步方法中用到某个变量，而此变量又需要其它线程修改后才能符合本线程的需要，那么可以在同步方法中使用wait()方法。使用wait方法可以中断方法的执行，使本线程等待，暂时让出CPU的使用权，并允许其它线程使用这个同步方法。其它线程如果在使用这个同步方法时不需要等待，那么它使用完这个同步方法的同时，应当用notifyAll()方法通知所有的由于使用这个同步方法而处于等待的线程结束等待

9．不合理。

10．“吵醒”休眠的线程。一个占有CPU资源的线程可以让休眠的线程调用interrupt 方法“吵醒”自己，即导致休眠的线程发生InterruptedException异常，从而结束休眠，重新排队等待CPU资源。

11．

public class Xiti11

{ public static void main(String args[])

{ Cinema a=new Cinema();

a.zhang.start();

a.sun.start();

a.zhao.start();

}

}

class TicketSeller //负责卖票的类。

{ int fiveNumber=3,tenNumber=0,twentyNumber=0;

public synchronized void sellTicket(int receiveMoney)

{ if(receiveMoney==5)

{ fiveNumber=fiveNumber+1;

System.out.println(Thread.currentThread().getName()+

"给我5元钱，这是您的1张入场卷");

}

else if(receiveMoney==10)

{ while(fiveNumber<1)

{ try { System.out.println(Thread.currentThread().getName()+"靠边等");

wait();

System.out.println(Thread.currentThread().getName()+"结束等待");

}

catch(InterruptedException e) {}

}

fiveNumber=fiveNumber-1;

tenNumber=tenNumber+1;

System.out.println(Thread.currentThread().getName()+

"给我10元钱,找您5元,这是您的1张入场卷");

}

else if(receiveMoney==20)

{ while(fiveNumber<1||tenNumber<1)

{ try { System.out.println(Thread.currentThread().getName()+"靠边等");

wait();

System.out.println(Thread.currentThread().getName()+"结束等待");

}

catch(InterruptedException e) {}

}

fiveNumber=fiveNumber-1;

tenNumber=tenNumber-1;

twentyNumber=twentyNumber+1;

System.out.println(Thread.currentThread().getName()+

"给20元钱，找您一张5元和一张10元，这是您的1张入场卷");

}

notifyAll();

}

}

class Cinema implements Runnable

{ Thread zhang,sun,zhao;

TicketSeller seller;

Cinema()

{ zhang=new Thread(this);

sun=new Thread(this);

zhao=new Thread(this);

zhang.setName("张小有");

sun.setName("孙大名");

zhao.setName("赵中堂");

seller=new TicketSeller();

}

public void run()

{ if(Thread.currentThread()==zhang)

{ seller.sellTicket(20);

}

else if(Thread.currentThread()==sun)

{ seller.sellTicket(10);

}

else if(Thread.currentThread()==zhao)

{ seller.sellTicket(5);

}

}

}

12．参照本章例子9。

13．参照本章例子19。

14．BA

习题16

1．URL对象调用InputStream openStream() 方法可以返回一个输入流。

2．客户端的程序使用Socket类建立负责连接到服务器的套接字对象称为socket对象。

使用Socket的构造方法Socket(String host,int port)，建立连接到服务器的套接字对象。参考16.3.2

3．JEditorPane

4．会返回一个和客户端Socket对象相连接的Socket对象。

5． 域名/IP 地址 例如，www.sina.com.cn/202.108.35.210

6．

(1) 客户端

import java.net.\*;

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Client

{ public static void main(String args[])

{ new ComputerClient();

}

}

class ComputerClient extends Frame implements Runnable,ActionListener

{ Button connection,send;

TextField inputText,showResult;

Socket socket=null;

DataInputStream in=null;

DataOutputStream out=null;

Thread thread;

ComputerClient()

{ socket=new Socket();

setLayout(new FlowLayout());

Box box=Box.createVerticalBox();

connection=new Button("连接服务器");

send=new Button("发送");

send.setEnabled(false);

inputText=new TextField(12);

showResult=new TextField(12);

box.add(connection);

box.add(new Label("输入三角形三边的长度,用逗号或空格分隔:"));

box.add(inputText);

box.add(send);

box.add(new Label("收到的结果："));

box.add(showResult);

connection.addActionListener(this);

send.addActionListener(this);

thread=new Thread(this);

add(box);

setBounds(10,30,300,400);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

});

}

public void actionPerformed(ActionEvent e)

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

{ try //请求和服务器建立套接字连接：

{ if(socket.isConnected())

{}

else

{ InetAddress address=InetAddress.getByName("127.0.0.1");

InetSocketAddress socketAddress=new InetSocketAddress(address,4331);

socket.connect(socketAddress);

in =new DataInputStream(socket.getInputStream());

out = new DataOutputStream(socket.getOutputStream());

send.setEnabled(true);

thread.start();

}

}

catch (IOException ee){}

}

if(e.getSource()==send)

{ String s=inputText.getText();

if(s!=null)

{ try { out.writeUTF(s);

}

catch(IOException e1){}

}

}

}

public void run()

{ String s=null;

while(true)

{ try{ s=in.readUTF();

showResult.setText(s);

}

catch(IOException e)

{ showResult.setText("与服务器已断开");

break;

}

}

}

}

（2）服务器端

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class Server

{ public static void main(String args[])

{ ServerSocket server=null;

Server\_thread thread;

Socket you=null;

while(true)

{ try{ server=new ServerSocket(4331);

}

catch(IOException e1)

{ System.out.println("正在监听"); //ServerSocket对象不能重复创建

}

try{ System.out.println(" 等待客户呼叫");

you=server.accept();

System.out.println("客户的地址:"+you.getInetAddress());

}

catch (IOException e)

{ System.out.println("正在等待客户");

}

if(you!=null)

{ new Server\_thread(you).start(); //为每个客户启动一个专门的线程

}

}

}

}

class Server\_thread extends Thread

{ Socket socket;

DataOutputStream out=null;

DataInputStream in=null;

String s=null;

boolean quesion=false;

Server\_thread(Socket t)

{ socket=t;

try { out=new DataOutputStream(socket.getOutputStream());

in=new DataInputStream(socket.getInputStream());

}

catch (IOException e)

{}

}

public void run()

{ while(true)

{ double a[]=new double[3] ;

int i=0;

try{ s=in.readUTF();//堵塞状态，除非读取到信息

quesion=false;

StringTokenizer fenxi=new StringTokenizer(s," ,");

while(fenxi.hasMoreTokens())

{ String temp=fenxi.nextToken();

try{ a[i]=Double.valueOf(temp).doubleValue();i++;

}

catch(NumberFormatException e)

{ out.writeUTF("请输入数字字符");

quesion=true;

}

}

if(quesion==false)

{ double p=(a[0]+a[1]+a[2])/2.0;

out.writeUTF(" "+Math.sqrt(p\*(p-a[0])\*(p-a[1])\*(p-a[2])));

}

}

catch (IOException e)

{ System.out.println("客户离开");

return;

}

}

}

}

7．参照本章例子16.6及以下代码。

（1）服务器

Server.java

import java.io.\*;

import java.net.\*;

import java.util.zip.\*;

public class Server

{ public static void main(String args[])

{ ServerSocket server=null;

ServerThread thread;

Socket you=null;

while(true)

{ try{ server=new ServerSocket(4331);

}

catch(IOException e1)

{ System.out.println("正在监听");

}

try{ you=server.accept();

System.out.println("客户的地址:"+you.getInetAddress());

}

catch (IOException e)

{ System.out.println("正在等待客户");

}

if(you!=null)

{ new ServerThread(you).start();

}

}

}

}

class ServerThread extends Thread

{ Socket socket;

ZipOutputStream out;

String s=null;

ServerThread(Socket t)

{ socket=t;

try { out=new ZipOutputStream(socket.getOutputStream());

}

catch (IOException e){}

}

public void run()

{ try{out.putNextEntry(new ZipEntry("Example.java"));

FileInputStream reader=new FileInputStream("Example.java");

byte b[]=new byte[1024];

int n=-1;

while((n=reader.read(b,0,1024))!=-1)

{ out.write(b,0,n); //发送压缩后的数据到客户端。

}

out.putNextEntry(new ZipEntry("E.java"));

reader=new FileInputStream("E.java");

n=-1;

while((n=reader.read(b,0,1024))!=-1)

{ out.write(b,0,n); //发送压缩后的数据到客户端。

}

reader.close();

out.close();

}

catch (IOException e) {}

}

}

（2）客户端

Client.java

import java.net.\*;

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.zip.\*;

public class Client extends Frame implements Runnable,ActionListener

{ Button connection,getFile;

TextArea showResult;

Socket socket=null;

ZipInputStream in;

Thread thread;

public Client()

{ socket=new Socket();

connection=new Button("连接服务器,获取文件内容");

setLayout(new FlowLayout());

showResult=new TextArea(10,28);

add(connection);

add(showResult);

connection.addActionListener(this);

thread=new Thread(this);

setBounds(100,100,460,410);

setVisible(true);

addWindowListener(new WindowAdapter()

{ public void windowClosing(WindowEvent e)

{ System.exit(0);

}

});

}

public void run()

{ byte b[]=new byte[1024];

ZipEntry zipEntry=null;

while(true)

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

{ showResult.append("\n"+zipEntry.toString()+":\n");

int n=-1;

while((n=in.read(b,0,1024))!=-1)

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

showResult.append(str);

}

}

}

catch(IOException e) { }

}

}

public void actionPerformed(ActionEvent e)

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

{ try { if(socket.isConnected())

{}

else

{ InetAddress address=InetAddress.getByName("127.0.0.1");

InetSocketAddress socketAddress=new InetSocketAddress(address,4331);

socket.connect(socketAddress);

in=new ZipInputStream(socket.getInputStream());

thread.start();

}

}

catch (IOException ee)

{ System.out.println(ee);

}

}

}

public static void main(String args[])

{ Client win=new Client();

}

}

习题17

1．

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Xiti1 extends Applet implements ActionListener{

Button button;

TextField text;

int sum;

public void init() {

button=new Button("点点看...");

text=new TextField(20);

add(button);

add(text);

button.addActionListener(this);

}

public void actionPerformed(ActionEvent e){

String str=button.getLabel();

text.setText("按钮上写着:"+str);

}

}

超文本文件：

<applet code=Xiti1.class height=180 width=300>

</applet>

2．

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Xiti2 extends Applet implements ActionListener

{ TextField text1,text2;

Label label;

public void init()

{ text1=new TextField(10);

text2=new TextField(20);

Box box1=Box.createHorizontalBox();

Box box2=Box.createHorizontalBox();

Box boxV=Box.createVerticalBox();

box1.add(new Label("输入一个数回车确定:"));

box1.add(text1);

label=new Label("数的平方:");

box2.add(label);

box2.add(text2);

boxV.add(box1);

boxV.add(box2);

add(boxV);

text2.setEditable(false);

text1.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{ String number=e.getActionCommand();

try{ double n=Double.parseDouble(number);

double m=n\*n;

label.setText(n+"的平方:");

text2.setText(""+m);

text1.setText("");

validate();

}

catch(NumberFormatException exp)

{ text2.setText(""+exp);

}

}

}

3．参照本章例子17.2，17.3。