**Java2实用教程（第5版）最新答案 耿祥义**

<https://blog.csdn.net/lytwy123/article/details/83998175>

# 第1章

一、问答题

1．James Gosling

2．需3个步骤：

用文本编辑器编写源文件。

使用javac编译源文件,得到字节码文件。

使用解释器运行程序。

3．由类所构成，应用程序必须有一个类含有public static void main(String args[])方法，含有该方法的类称为应用程序的主类。不一定，但最多有一个public类。

4．set classpath=D:\jdk\jre\lib\rt.jar;.;

5.  java和class

6.   java Bird

7. 独行风格（大括号独占行）和行尾风格（左大扩号在上一行行尾，右大括号独占行）

二、选择题

1．B。 2．D。

三、阅读程序

1．（a）Person.java。（b）两个字节码，分别是Person.class和Xiti.class。（c）得到“NoSuchMethodError”，得到“NoClassDefFoundError: Xiti/class”，得到“您好，很高兴认识您 nice to meet you”

# 第2章

一、问答题

1．用来标识类名、变量名、方法名、类型名、数组名、文件名的有效字符序列称为标识符。标识符由字母、下划线、美元符号和数字组成，第一个字符不能是数字。false不是标识符。

2．关键字就是Java语言中已经被赋予特定意义的一些单词，不可以把关键字作为名字来用。不是关键字。class implements interface enum extends abstract。

3．boolean，char，byte，short，int，long，float，double。

4．float常量必须用F或f为后缀。double常量用D或d为后缀，但允许省略后缀。

5．一维数组名.length。二维数组名.length。

二、选择题

1．C。2．ADF。3．B。4．BE。5．【代码2】【代码3】【代码4】【代码5】。6．B。

三、阅读或调试程序

1．属于操作题，解答略。

2．属于操作题，解答略。

3．属于操作题，解答略。

4．【代码1】：4。【代码2】：b[0]=1。

5．【代码1】：40。【代码2】：7

四、编写程序

1.  public class E {

   public static void main(String args[]) {

     System.out.println((int)'你');

     System.out.println((int)'我');

     System.out.println((int)'他');

   }

}

2.   public class E {

   public static void main (String args[ ]) {

      char cStart='α',cEnd='ω';

      for(char c=cStart;c<=cEnd;c++)

        System.out.print(" "+c);

   }

}

# 第3章

一、问答题

1．boolean

2．不可以

3．boolean

4. 不是必须的

5．结束while语句的执行

6．可以

二、选择题

1．A。 2．C。 3．C。

三、阅读程序

1．你,苹,甜

2．Jeep好好

3．x=-5,y=-1

四、编程序题

1．public class Xiti1 {

  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);

   }

}

2．public class Xiti2 {

  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);

          }

      }

   }

}

3．class Xiti3 {

  public static void main(String args[]) {

      double sum=0,a=1,i=1;

      do { sum=sum+a;

           i++;

           a=(1.0/i)\*a;

       }

       while(i<=20);

       System.out.println("使用do-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);

   }

}

4．public class Xiti4 {

  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);

     }

  }

}

5．public class Xiti5 {

  public static void main(String args[]) {

     int m=8,item=m,i=1;

     long sum=0;

     for(i=1,sum=0,item=m;i<=10;i++) {

        sum=sum+item;

        item=item\*10+m;

     }

     System.out.println(sum);

  }

}

6． public class Xiti6 {

  public static void main(String args[]) {

      int n=1;

      long sum=0;

      while(true) {

        sum=sum+n;

        n++;

        if(sum>=8888)

          break;

      }

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

   }

}

# 第4章

一、问答题

1. 封装、继承和多态。

2．当类名由几个单词复合而成时，每个单词的首字母使用大写。

3．名字的首单词的首字母使用小写，如果变量的名字由多个单词组成，从第2个单词开始的其它单词的首字母使用大写。

4．属性

5．行为

6．用类创建对象时。没有类型

7．用类创建对象时。

8．一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。可以。

9．可以。不可以。

10．不可以。

11．一个类通过使用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。所有对象的类变量都分配给相同的一处内存，对象共享类变量。

12．代表调用当前方法的对象。不可以。

二、选择题

1．B。2．D。3．D。4．D。5．CD。6．【代码1】【代码4】。7．【代码4】。

三、阅读程序

1．【代码1】：1，【代码2】：121，【代码3】：121。

2．sum=-100。

3. 27。

4．【代码1】:100，【代码2】：20.0。

5. 上机实习题目，解答略。

6. 上机实习题目，解答略。

四、编程题

CPU.java

public class CPU {

   int speed;

   int getSpeed() {

      return speed;

   }

   public void setSpeed(int speed) {

      this.speed = speed;

   }

}

HardDisk.java

public class HardDisk {

   int amount;

   int getAmount() {

      return amount;

   }

   public void setAmount(int amount) {

      this.amount = amount;

   }

}

PC.java

public class PC {

    CPU cpu;

    HardDisk HD;

    void setCPU(CPU cpu) {

        this.cpu = cpu;

    }

     void setHardDisk(HardDisk HD) {

        this.HD = HD;

    }

    void show(){

       System.out.println("CPU速度:"+cpu.getSpeed());

       System.out.println("硬盘容量:"+HD.getAmount());

    }

}

Test.java

public class Test {

   public static void main(String args[]) {

       CPU cpu = new CPU();

       HardDisk HD=new HardDisk();

       cpu.setSpeed(2200);

       HD.setAmount(200);

       PC pc =new PC();

       pc.setCPU(cpu);

       pc.setHardDisk(HD);

       pc.show();

    }

}

# 第5章

一、问答题

1．不可以。

2．是。

3．不继承。

4．声明与父类同名的成员变量。

5．子类重写的方法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，重写的方法的名字、参数个数、参数的类型和父类的方法完全相同。重写方法的目的是隐藏继承的方法，子类通过方法的重写可以把父类的状态和行为改变为自身的状态和行为。

6．不可以。

7．Abstract类。

8．上转型对象不能操作子类新增的成员变量，不能调用子类新增的方法。上转型对象可以访问子类继承或隐藏的成员变量，可以调用子类继承的方法或子类重写的实例方法。

9．通过重写方法。

10．面向抽象编程目的是为了应对用户需求的变化，核心是让类中每种可能的变化对应地交给抽象类的一个子类类去负责，从而让该类的设计者不去关心具体实现。

二、选择题

1．C。2．D。3．CD。4．D。5．B。6．B。7．D。8．B。9．A。

三、阅读程序

1．【代码1】：15.0。【代码2】：8.0。

2．【代码1】：11。【代码2】：11。

3．【代码1】：98.0。【代码2】：12。【代码3】：98.0。【代码4】：9。

4．【代码1】：120。【代码2】：120。【代码3】：-100。

四、编程题

Animal.java

public abstract class Animal {

    public abstract void cry();

    public abstract String getAnimalName();

}

Simulator.java

public class Simulator {

   public void playSound(Animal animal) {

       System.out.print("现在播放"+animal.getAnimalName()+"类的声音:");

       animal.cry();

   }

}

Dog.java

public class Dog extends Animal {

   public void cry() {

      System.out.println("汪汪...汪汪");

   }

   public String getAnimalName() {

      return "狗";

   }

}

Cat.java

public class Cat extends Animal {

   public void cry() {

      System.out.println("喵喵...喵喵");

   }

   public String getAnimalName() {

      return "猫";

   }

}

Application.java

public class Example5\_13 {

   public static void main(String args[]) {

      Simulator simulator = new Simulator();

      simulator.playSound(new Dog());

      simulator.playSound(new Cat());

   }

}

# 第6章

一、问答题

1．不能。

2．不能。

3．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。

4．不可以。

5．可以。

二、选择题

1．D。2．AB。3．B。

三、阅读程序

1．【代码1】：15.0。【代码2】：8。

2．【代码1】：18。【代码2】：15。

四、编程题

Animal.java

public interface Animal {

    public abstract void cry();

    public abstract String getAnimalName();

}

Simulator.java

public class Simulator {

   public void playSound(Animal animal) {

       System.out.print("现在播放"+animal.getAnimalName()+"类的声音:");

       animal.cry();

   }

}

Dog.java

public class Dog implements Animal {

   public void cry() {

      System.out.println("汪汪...汪汪");

   }

   public String getAnimalName() {

      return "狗";

   }

}

Cat.java

public class Cat implements Animal {

   public void cry() {

      System.out.println("喵喵...喵喵");

   }

   public String getAnimalName() {

      return "猫";

   }

}

Application.java

public class Example5\_13 {

   public static void main(String args[]) {

      Simulator simulator = new Simulator();

      simulator.playSound(new Dog());

      simulator.playSound(new Cat());

   }

}

# 第7章

一、问答题

1．有效。

2．可以。

3．不可以。

4．一定是。

二、选择题

1．C。2．C。

三、阅读程序

1．大家好，祝工作顺利！

2．p是接口变量。

3．你好 fine thanks。

4．属于上机实习程序，解答略。

四、编程题

import java.util.\*;

public class E {

    public static void main (String args[ ]){

      Scanner reader = new Scanner(System.in);

      double sum = 0;

       int m = 0;

       while(reader.hasNextDouble()){

           double x = reader.nextDouble();

           assert x< 100:"数据不合理";

           m = m+1;

           sum = sum+x;

       }

       System.out.printf("%d个数的和为%f\n",m,sum);

       System.out.printf("%d个数的平均值是%f\n",m,sum/m);

    }

}

# 第8章

一、问答题

1．不是。"\\hello"是。

2．4和3。

3．false和true。

4．负数。

5．是true。

6．3和-1。

7．会发生NumberFormatException异常。

二、选择题

1．A。2．C。3．B。4．D。5．C。

三、阅读程序

1．【代码】：苹果。

2．【代码】：Love:Game。

3．【代码1】：15。【代码2】：abc我们。

4．【代码】：13579。

5．【代码】：9javaHello。

6．属于上机实习程序，解答略。

7．属于上机实习程序，解答略。

四、编程题

1．public class E {

  public static void main (String args[ ]) {

     String s1,s2,t1="ABCDabcd";

     s1=t1.toUpperCase();

     s2=t1.toLowerCase();

     System.out.println(s1);

     System.out.println(s2);

     String s3=s1.concat(s2);

      System.out.println(s3);

   }

}

2.   public class E {

  public static void main (String args[ ]) {

     String s="ABCDabcd";

     char cStart=s.charAt(0);

     char cEnd = s.charAt(s.length()-1);

     System.out.println(cStart);

     System.out.println(cEnd);

   }

}

3.   import java.util.\*;

public class E {

  public static void main (String args[ ]) {

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

      try{ year1=Integer.parseInt(args[0]);

           month1=Integer.parseInt(args[1]);

           day1=Integer.parseInt(args[2]);

           year2=Integer.parseInt(args[3]);

           month2=Integer.parseInt(args[4]);

           day2=Integer.parseInt(args[5]);

       }

       catch(NumberFormatException e)

         { year1=2012;

           month1=0;

           day1=1;

           year2=2018;

           month2=0;

           day2=1;

       }

       Calendar calendar=Calendar.getInstance();

       calendar.set(year1,month1-1,day1);

       long timeYear1=calendar.getTimeInMillis();

       calendar.set(year2,month2-1,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");

   }

}

4.   import java.util.\*;

public class E {

  public static void main (String args[ ]) {

   double a=0,b=0,c=0;

      a=12;

      b=24;

      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);

   }

}

5．public class E {

      public static void main (String args[ ]) {

        String str = "ab123you你是谁？";

        String regex = "\\D+";

        str = str.replaceAll(regex,"");

        System.out.println(str);

      }

}

6． import java.util.\*;

public class E {

   public static void main(String args[]) {

      String cost = "数学87分，物理76分，英语96分";

      Scanner scanner = new Scanner(cost);

      scanner.useDelimiter("[^0123456789.]+");

      double sum=0;

      int count =0;

      while(scanner.hasNext()){

         try{  double score = scanner.nextDouble();

               count++;

               sum = sum+score;

               System.out.println(score);

         }

         catch(InputMismatchException exp){

              String t = scanner.next();

         }

      }

      System.out.println("总分:"+sum+"分");

      System.out.println("平均分:"+sum/count+"分");

   }

}

# 第9章

一、问答题

1．Frame容器的默认布局是BorderLayout布局。

2．不可以。

3．ActionEvent。

4．DocumentEvent。

5．5个。

6．MouseMotionListener。

二、选择题

1．C。2．A。3．A。4．D。5．C。

三、编程题

1． import java.awt.\*;

import javax.swing.event.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class E {

   public static void main(String args[]) {

      Computer fr=new Computer();

   }

}

class Computer extends JFrame implements DocumentListener {

   JTextArea text1,text2;

   int count=1;

   double sum=0,aver=0;

   Computer() {

      setLayout(new FlowLayout());

      text1=new JTextArea(6,20);

      text2=new JTextArea(6,20);

      add(new JScrollPane(text1));

      add(new JScrollPane(text2));

      text2.setEditable(false);

      (text1.getDocument()).addDocumentListener(this);

      setSize(300,320);

      setVisible(true);

      validate();

      setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

   }

   public void changedUpdate(DocumentEvent e) {

      String s=text1.getText();

      String []a =s.split("[^0123456789.]+");

      sum=0;

      aver=0;

      for(int i=0;i<a.length;i++) {

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

        }

        catch(Exception ee) {}

     }

     aver=sum/count;

     text2.setText(null);

     text2.append("\n和:"+sum);

     text2.append("\n平均值:"+aver);

   }

   public void removeUpdate(DocumentEvent e){

      changedUpdate(e);

   }

   public void insertUpdate(DocumentEvent e){

      changedUpdate(e);

   }

}

2.   import java.awt.\*;

import javax.swing.event.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class E {

   public static void main(String args[]) {

      ComputerFrame fr=new ComputerFrame();

   }

}

class ComputerFrame extends JFrame implements ActionListener {

  JTextField text1,text2,text3;

  JButton buttonAdd,buttonSub,buttonMul,buttonDiv;

  JLabel label;

  public ComputerFrame() {

   setLayout(new FlowLayout());

   text1=new JTextField(10);

   text2=new JTextField(10);

   text3=new JTextField(10);

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

   label.setBackground(Color.green);

   add(text1);

   add(label);

   add(text2);

   add(text3);

   buttonAdd=new JButton("加");

   buttonSub=new JButton("减");

   buttonMul=new JButton("乘");

   buttonDiv=new JButton("除");

   add(buttonAdd);

   add(buttonSub);

   add(buttonMul);

   add(buttonDiv);

   buttonAdd.addActionListener(this);

   buttonSub.addActionListener(this);

   buttonMul.addActionListener(this);

   buttonDiv.addActionListener(this);

   setSize(300,320);

   setVisible(true);

   validate();

   setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

  }

  public void actionPerformed(ActionEvent e) {

    double n;

    if(e.getSource()==buttonAdd) {

       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()==buttonSub) {

       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()==buttonMul)

      {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()==buttonDiv)

      {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();

  }

}

3.   import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class E {

   public static void main(String args[]){

      Window win = new Window();

      win.setTitle("使用MVC结构");

      win.setBounds(100,100,420,260);

   }

}

class Window extends JFrame implements ActionListener {

   Lader lader;             //模型

   JTextField textAbove,textBottom,textHeight;   //视图

   JTextArea showArea;         //视图

   JButton controlButton;        //控制器

   Window() {

      init();

      setVisible(true);

      setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

   }

   void init() {

     lader = new Lader();

     textAbove = new JTextField(5);

     textBottom = new JTextField(5);

     textHeight = new JTextField(5);

     showArea = new JTextArea();

     controlButton=new JButton("计算面积");

     JPanel pNorth=new JPanel();

     pNorth.add(new JLabel("上底:"));

     pNorth.add(textAbove);

     pNorth.add(new JLabel("下底:"));

     pNorth.add(textBottom);

     pNorth.add(new JLabel("高:"));

     pNorth.add(textHeight);

     pNorth.add(controlButton);

     controlButton.addActionListener(this);

     add(pNorth,BorderLayout.NORTH);

     add(new JScrollPane(showArea),BorderLayout.CENTER);

   }

   public void actionPerformed(ActionEvent e) {

     try{

        double above = Double.parseDouble(textAbove.getText().trim());

        double bottom = Double.parseDouble(textBottom.getText().trim());

        double height = Double.parseDouble(textHeight.getText().trim());

        lader.setAbove(above) ;

        lader.setBottom(bottom);

        lader.setHeight(height);

        double area = lader.getArea();

        showArea.append("面积:"+area+"\n");

     }

     catch(Exception ex) {

        showArea.append("\n"+ex+"\n");

     }

   }

}

class Lader {

    double above,bottom,height;

    public double getArea() {

       double area = (above+bottom)\*height/2.0;

      return area;

   }

    public void setAbove(double a) {

      above = a;

    }

   public void setBottom(double b) {

      bottom = b;

   }

   public void setHeight(double c) {

     height = c;

   }

}

# 第10章

一、问答题

1．使用FileInputStream。

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

3．不可以。

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。

二、选择题

1．C。2．B。

三、阅读程序

1．【代码1】：51。【代码2】：0。

2．【代码1】：3。【代码2】：abc。【代码3】：1。【代码4】：dbc。

3．上机实习题，解答略。

四、编程题

1． import java.io.\*;

public class E {

   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){}

    }

}

2.   import java.io.\*;

public class E {

   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){}

   }

}

3.   import java.io.\*;

import java.util.\*;

public class E {

   public static void main(String args[]) {

      File file = new File("a.txt");

      Scanner sc = null;

      double sum=0;

      int count = 0;

      try { sc = new Scanner(file);

            sc.useDelimiter("[^0123456789.]+");

            while(sc.hasNext()){

               try{  double price = sc.nextDouble();

                    count++;

                    sum = sum+price;

                    System.out.println(price);

               }

               catch(InputMismatchException exp){

                    String t = sc.next();

               }

            }

            System.out.println("平均价格:"+sum/count);

      }

      catch(Exception exp){

         System.out.println(exp);

      }

   }

}

# 第11章

一、问答题

1．（1）添加数据源，（2）选择驱动程序，（3）命名数据源名称。

2．不必使用数据名称。

3．减轻数据库内部SQL语句解释器的负担。

4．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。

5．（1）用setAutoCommit()方法关闭自动提交模式，（2）用commit()方法处理事务，（3）用rollback()方法处理事务失败。

四、编程题

1.  import java.sql.\*;

import java.util.\*;

public class E {

   public static void main(String args[]) {

     Query query=new Query();

     String dataSource="myData";

     String tableName="goods";

     Scanner read=new Scanner(System.in);

     System.out.print("输入数据源名:");

     dataSource = read.nextLine();

     System.out.print("输入表名:");

     tableName = read.nextLine();

     query.setDatasourceName(dataSource);

     query.setTableName(tableName);

     query.setSQL("SELECT \* FROM "+tableName);

     query.inputQueryResult();

   }

}

class Query {

   String datasourceName="";        //数据源名

   String tableName="";            //表名

   String SQL;                    //SQL语句

   public Query() {

      try{  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

      }

      catch(ClassNotFoundException e) {

         System.out.print(e);

      }

   }

   public void setDatasourceName(String s) {

      datasourceName = s.trim();

   }

   public void setTableName(String s) {

      tableName = s.trim();

   }

   public void setSQL(String SQL) {

      this.SQL = SQL.trim();

   }

   public void inputQueryResult() {

      Connection con;

      Statement sql;

      ResultSet rs;

      try {

           String uri = "jdbc:odbc:"+datasourceName;

           String id = "";

           String password = "";

           con = DriverManager.getConnection(uri,id,password);

           DatabaseMetaData metadata = con.getMetaData();

           ResultSet rs1 = metadata.getColumns(null,null,tableName,null);

           int 字段个数 = 0;

           while(rs1.next()) {

              字段个数++;

           }

           sql = con.createStatement();

           rs = sql.executeQuery(SQL);

           while(rs.next()) {

             for(int k=1;k<=字段个数;k++) {

                 System.out.print(" "+rs.getString(k)+" ");

             }

             System.out.println("");

           }

           con.close();

       }

       catch(SQLException e) {

           System.out.println("请输入正确的表名"+e);

       }

   }

}

2.   import java.sql.\*;

import java.util.\*;

public class E {

   public static void main(String args[]) {

     Query query = new Query();

     String dataSource = "myData";

     String tableName = "goods";

     query.setDatasourceName(dataSource);

     query.setTableName(tableName);

     String name = "";

     Scanner read=new Scanner(System.in);

     System.out.print("商品名:");

     name = read.nextLine();

     String str="'%["+name+"]%'";

     String SQL = "SELECT \* FROM "+tableName+" WHERE name LIKE "+str;

     query.setSQL(SQL);

     System.out.println(tableName+"表中商品名是"+name+"的记录");

     query.inputQueryResult();

   }

}

class Query {

   String datasourceName="";        //数据源名

   String tableName="";            //表名

   String SQL;                    //SQL语句

   public Query() {

      try{  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

      }

      catch(ClassNotFoundException e) {

         System.out.print(e);

      }

   }

   public void setDatasourceName(String s) {

      datasourceName = s.trim();

   }

   public void setTableName(String s) {

      tableName = s.trim();

   }

   public void setSQL(String SQL) {

      this.SQL = SQL.trim();

   }

   public void inputQueryResult() {

      Connection con;

      Statement sql;

      ResultSet rs;

      try {

           String uri = "jdbc:odbc:"+datasourceName;

           String id = "";

           String password = "";

           con = DriverManager.getConnection(uri,id,password);

           DatabaseMetaData metadata = con.getMetaData();

           ResultSet rs1 = metadata.getColumns(null,null,tableName,null);

           int 字段个数 = 0;

           while(rs1.next()) {

              字段个数++;

           }

           sql = con.createStatement();

           rs = sql.executeQuery(SQL);

           while(rs.next()) {

             for(int k=1;k<=字段个数;k++) {

                 System.out.print(" "+rs.getString(k)+" ");

             }

             System.out.println("");

           }

           con.close();

       }

       catch(SQLException e) {

           System.out.println("请输入正确的表名"+e);

       }

   }

}

3．将例子5中的代码：

String SQL = "SELECT \* FROM "+tableName+" ORDER BY name";

更改为：

String SQL = "SELECT \* FROM "+tableName+" ORDER BY madeTime";

可达题目要求。

# 第12章

一、问答题

1．4种状态：新建、运行、中断和死亡。

2．有4种原因的中断：（1）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。（2）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。（3）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。（4）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。

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。

二、选择题

1．A。2．A。3．B。

三、阅读程序

1．属于上机调试题目，解答略。

2．属于上机调试题目，解答略。

3．属于上机调试题目，解答略。

4．属于上机调试题目，解答略。

5．属于上机调试题目，解答略。

6．属于上机调试题目，解答略

7．【代码】：BA。

8．属于上机调试题目，解答略

四、编写程序

1.  public class E {

 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);

       }

   }

}

2.  参照本章例子6

3．参照本章例子9

# 第13章

一、问答题

1．一个URL对象通常包含最基本的三部分信息：协议、地址、资源。

2．URL对象调用InputStream openStream() 方法可以返回一个输入流，该输入流指向URL对象所包含的资源。通过该输入流可以将服务器上的资源信息读入到客户端。

3．客户端的套接字和服务器端的套接字通过输入、输出流互相连接后进行通信。

4．使用方法accept()，accept()会返回一个和客户端Socket对象相连接的Socket对象。accept方法会堵塞线程的继续执行，直到接收到客户的呼叫。。

5．域名/IP。

四、编程题

1.  （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;

            }

      }

   }

}

2.   客户端Client.java

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 ChatClient();

   }

}

class ChatClient extends Frame implements Runnable,ActionListener

{  Button connection,send;

   TextField inputName,inputContent;

   TextArea chatResult;

   Socket socket=null;

   DataInputStream in=null;

   DataOutputStream out=null;

   Thread thread;

   String name="";

   public ChatClient ()

   {  socket=new Socket();

      Box box1=Box.createHorizontalBox();

      connection=new Button("连接服务器");

      send=new Button("发送");

      send.setEnabled(false);

      inputName=new TextField(6);

      inputContent=new TextField(22);

      chatResult=new TextArea();

      box1.add(new Label("输入妮称:"));

      box1.add(inputName);

      box1.add(connection);

      Box box2=Box.createHorizontalBox();

      box2.add(new Label("输入聊天内容:"));

      box2.add(inputContent);

      box2.add(send);

      connection.addActionListener(this);

      send.addActionListener(this);

      thread=new Thread(this);

      add(box1,BorderLayout.NORTH);

      add(box2,BorderLayout.SOUTH);

      add(chatResult,BorderLayout.CENTER);

      setBounds(10,30,400,280);

      setVisible(true);

      validate();

      addWindowListener(new WindowAdapter()

                   {  public void windowClosing(WindowEvent e)

                            {  System.exit(0);

                            }

                   });

   }

   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,666);

                socket.connect(socketAddress);

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

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

                name=inputName.getText();

                out.writeUTF("姓名:"+name);

                send.setEnabled(true);

                if(!(thread.isAlive()))

                   thread=new Thread(this);

                thread.start();

               }

         }

         catch (IOException ee){}

      }

     if(e.getSource()==send)

      {  String s=inputContent.getText();

         if(s!=null)

           {  try { out.writeUTF("聊天内容:"+name+":"+s);

                  }

              catch(IOException e1){}

           }

      }

   }

   public void run()

   {  String s=null;

      while(true)

       {    try{  s=in.readUTF();

                  chatResult.append("\n"+s);

               }

           catch(IOException e)

               {  chatResult.setText("与服务器已断开");

                  try { socket.close();

                      }

                  catch(Exception exp) {}

                  break;

               }

       }

  }

}

服务器端ChatServer.java

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class ChatServer

{  public static void main(String args[])

    { ServerSocket server=null;

      Socket you=null;

      Hashtable peopleList;

      peopleList=new Hashtable();

      while(true)

            { try  {  server=new ServerSocket(666);

                  }

             catch(IOException e1)

                  {  System.out.println("正在监听");

                  }

             try  {  you=server.accept();

                     InetAddress address=you.getInetAddress();

                     System.out.println("客户的IP:"+address);

                  }

             catch (IOException e) {}

             if(you!=null)

                  {  Server\_thread peopleThread=new Server\_thread(you,peopleList);

                     peopleThread.start();

                  }

             else {  continue;

                  }

           }

  }

}

class Server\_thread extends Thread

{  String name=null;

   Socket socket=null;

   File file=null;

   DataOutputStream out=null;

   DataInputStream  in=null;

   Hashtable peopleList=null;

   Server\_thread(Socket t,Hashtable list)

       { peopleList=list;

         socket=t;

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

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

             }

         catch (IOException e) {}

        }

  public void run()

  {   while(true)

      { String s=null;

        try{

            s=in.readUTF();

            if(s.startsWith("姓名:"))

              { name=s;

                boolean boo=peopleList.containsKey(name);

                if(boo==false)

                  { peopleList.put(name,this);

                  }

                else

                  { out.writeUTF("请换妮称:");

                    socket.close();

                    break;

                  }

              }

            else if(s.startsWith("聊天内容"))

              {  String message=s.substring(s.indexOf(":")+1);

                 Enumeration chatPersonList=peopleList.elements();

                 while(chatPersonList.hasMoreElements())

                 {     ((Server\_thread)chatPersonList.nextElement()).out.writeUTF("聊天内容:"+

message);

                 }

              }

           }

        catch(IOException ee)

           {     Enumeration chatPersonList=peopleList.elements();

                 while(chatPersonList.hasMoreElements())

                   { try

                     {  Server\_thread  th=(Server\_thread)chatPersonList.nextElement();

                        if(th!=this&&th.isAlive())

                         { th.out.writeUTF("客户离线:"+name);

                         }

                     }

                     catch(IOException eee){}

                   }

                 peopleList.remove(name);

                 try { socket.close();

                    }

                 catch(IOException eee){}

                 System.out.println(name+"客户离开了");

                 break;

           }

     }

  }

}

3．BroadCastWord.java

import java.io.\*;

import java.net.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.Timer;

public class BroadCastWord extends Frame implements ActionListener

{  int port;

   InetAddress group=null;

   MulticastSocket socket=null;

   Timer time=null;

   FileDialog open=null;

   Button select,开始广播,停止广播;

   File file=null;

   String FileDir=null,fileName=null;

   FileReader in=null;

   BufferedReader bufferIn=null;

   int token=0;

   TextArea 显示正在播放内容,显示已播放的内容;

  public BroadCastWord()

  { super("单词广播系统");

   select=new Button("选择要广播的文件");

   开始广播=new Button("开始广播");

   停止广播=new Button("停止广播");

   select.addActionListener(this);

   开始广播.addActionListener(this);

   停止广播.addActionListener(this);

   time=new Timer(2000,this);

   open=new FileDialog(this,"选择要广播的文件",FileDialog.LOAD);

   显示正在播放内容=new TextArea(10,10);

   显示正在播放内容.setForeground(Color.blue);

   显示已播放的内容=new TextArea(10,10);

   Panel north=new Panel();

   north.add(select);

   north.add(开始广播);

   north.add(停止广播);

   add(north,BorderLayout.NORTH);

   Panel center=new Panel();

   center.setLayout(new GridLayout(1,2));

   center.add(显示正在播放内容);

   center.add(显示已播放的内容);

   add(center,BorderLayout.CENTER);

   validate();

   try

      { port=5000;

       group=InetAddress.getByName("239.255.0.0");

       socket=new MulticastSocket(port);

       socket.setTimeToLive(1);

       socket.joinGroup(group);

      }

  catch(Exception e)

       { System.out.println("Error: "+ e);

       }

  setBounds(100,50,360,380);

  setVisible(true);

  addWindowListener(new WindowAdapter()

                     { public void windowClosing(WindowEvent e)

                       { System.exit(0);

                           }

                         });

 }

 public void actionPerformed(ActionEvent e)

 {  if(e.getSource()==select)

     {显示已播放的内容.setText(null);

      open.setVisible(true);

      fileName=open.getFile();

      FileDir=open.getDirectory();

      if(fileName!=null)

       { time.stop();

         file=new File(FileDir,fileName);

         try

            {  file=new File(FileDir,fileName);

               in=new FileReader(file);

               bufferIn=new BufferedReader(in);

            }

         catch(IOException ee) { }

       }

     }

   else if(e.getSource()==开始广播)

     {  time.start();

     }

   else if(e.getSource()==time)

     {  String s=null;

        try  {  if(token==-1)

                 { file=new File(FileDir,fileName);

                   in=new FileReader(file);

                   bufferIn=new BufferedReader(in);

                 }

              s=bufferIn.readLine();

              if(s!=null)

                { token=0;

                  显示正在播放内容.setText("正在广播的内容:\n"+s);

                  显示已播放的内容.append(s+"\n");

                  DatagramPacket packet=null;

                  byte data[]=s.getBytes();

                  packet=new DatagramPacket(data,data.length,group,port);

                  socket.send(packet);

                }

              else

                {  token=-1;

                }

            }

        catch(IOException ee) { }

     }

   else if(e.getSource()==停止广播)

     {  time.stop();

     }

 }

public static void main(String[] args)

   {  BroadCastWord broad=new BroadCastWord();

   }

}

Receive.java

import java.net.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Receive extends Frame implements Runnable,ActionListener

{  int port;

  InetAddress group=null;

  MulticastSocket socket=null;

  Button 开始接收,停止接收;

  TextArea 显示正在接收内容,显示已接收的内容;

  Thread thread;

  boolean 停止=false;

  public Receive()

   {  super("定时接收信息");

     thread=new Thread(this);

     开始接收=new Button("开始接收");

     停止接收=new Button("停止接收");

     停止接收.addActionListener(this);

     开始接收.addActionListener(this);

     显示正在接收内容=new TextArea(10,10);

     显示正在接收内容.setForeground(Color.blue);

     显示已接收的内容=new TextArea(10,10);

     Panel north=new Panel();

     north.add(开始接收);

     north.add(停止接收);

     add(north,BorderLayout.NORTH);

     Panel center=new Panel();

     center.setLayout(new GridLayout(1,2));

     center.add(显示正在接收内容);

     center.add(显示已接收的内容);

     add(center,BorderLayout.CENTER);

     validate();

     port=5000;

     try{ group=InetAddress.getByName("239.255.0.0");

         socket=new MulticastSocket(port);

         socket.joinGroup(group);

       }

    catch(Exception e) { }

    setBounds(100,50,360,380);

    setVisible(true);

    addWindowListener(new WindowAdapter()

                     { public void windowClosing(WindowEvent e)

                       { System.exit(0);

                           }

                         });

   }

  public void actionPerformed(ActionEvent e)

   { if(e.getSource()==开始接收)

      { 开始接收.setBackground(Color.blue);

        停止接收.setBackground(Color.gray);

        if(!(thread.isAlive()))

           { thread=new Thread(this);

           }

        try {  thread.start();

             停止=false;

           }

        catch(Exception ee) { }

      }

    if(e.getSource()==停止接收)

      { 开始接收.setBackground(Color.gray);

        停止接收.setBackground(Color.blue);

        thread.interrupt();

        停止=true;

      }

   }

 public void run()

  { while(true)

    {  byte data[]=new byte[8192];

       DatagramPacket packet=null;

       packet=new DatagramPacket(data,data.length,group,port);

       try {  socket.receive(packet);

             String message=new String(packet.getData(),0,packet.getLength());

             显示正在接收内容.setText("正在接收的内容:\n"+message);

             显示已接收的内容.append(message+"\n");

           }

      catch(Exception e)  { }

      if(停止==true)

           { break;

           }

    }

  }

public static void main(String args[])

  {  new Receive();

  }

}

# 第14章

一、问答题

1．2个参数。

2．6个参数。

3．7个参数。

4．（1）创建AffineTransform对象，（2）进行旋转操作，（3）绘制旋转的图形。

二、编写程序

1.  import java.awt.\*;

import javax.swing.\*;

class MyCanvas extends Canvas {

  static int pointX[]=new int[5],

       pointY[]=new int[5];

   public void paint(Graphics g) {

      g.translate(200,200) ;    //进行坐标变换,将新的坐标原点设置为(200,200)。

      pointX[0]=0;

      pointY[0]=-120;

      double arcAngle=(72\*Math.PI)/180;

      for(int i=1;i<5;i++) {

         pointX[i]=(int)(pointX[i-1]\*Math.cos(arcAngle)-pointY[i-1]\*Math.sin(arcAngle));

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

      }

      g.setColor(Color.red);

      int starX[]={pointX[0],pointX[2],pointX[4],pointX[1],pointX[3],pointX[0]};

      int starY[]={pointY[0],pointY[2],pointY[4],pointY[1],pointY[3],pointY[0]};

      g.drawPolygon(starX,starY,6);

   }

}

public class E {

   public static void main(String args[]) {

      JFrame f=new JFrame();

      f.setSize(500,450);

      f.setVisible(true);

      MyCanvas canvas=new MyCanvas();

      f.add(canvas,"Center");

      f.validate();

      f.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

  }

}

2.   import java.awt.\*;

import javax.swing.\*;

import java.awt.geom.\*;

class MyCanvas extends Canvas {

   public void paint(Graphics g) {

     g.setColor(Color.red) ;

      Graphics2D g\_2d=(Graphics2D)g;

      QuadCurve2D quadCurve=

      new  QuadCurve2D.Double(2,10,51,90,100,10);

      g\_2d.draw(quadCurve);

      quadCurve.setCurve(2,100,51,10,100,100);

      g\_2d.draw(quadCurve);

   }

}

public class E {

   public static void main(String args[]) {

      JFrame f=new JFrame();

      f.setSize(500,450);

      f.setVisible(true);

      MyCanvas canvas=new MyCanvas();

      f.add(canvas,"Center");

      f.validate();

      f.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

  }

}

3.   import java.awt.\*;

import javax.swing.\*;

import java.awt.geom.\*;

class MyCanvas extends Canvas {

   public void paint(Graphics g) {

      g.setColor(Color.red) ;

      Graphics2D g\_2d=(Graphics2D)g;

      CubicCurve2D cubicCurve=

      new CubicCurve2D.Double(0,70,70,140,140,0,210,70);

      g\_2d.draw(cubicCurve);

   }

}

public class E {

   public static void main(String args[]) {

      JFrame f=new JFrame();

      f.setSize(500,450);

      f.setVisible(true);

      MyCanvas canvas=new MyCanvas();

      f.add(canvas,"Center");

      f.validate();

      f.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

  }

}

4.   import java.awt.\*;

import javax.swing.\*;

import java.awt.geom.\*;

class Flower extends Canvas

{  public void paint(Graphics g)

   {  Graphics2D  g\_2d=(Graphics2D)g;

      //花叶两边的曲线:

      QuadCurve2D

      curve\_1=new  QuadCurve2D.Double(200,200,150,160,200,100);

      CubicCurve2D curve\_2=

      new  CubicCurve2D.Double(200,200,260,145,190,120,200,100);

      //花叶中的纹线:

      Line2D line=new Line2D.Double(200,200,200,110);

      QuadCurve2D leaf\_line1=

      new  QuadCurve2D.Double(200,180,195,175,190,170);

      QuadCurve2D leaf\_line2=

      new  QuadCurve2D.Double(200,180,210,175,220,170);

      QuadCurve2D leaf\_line3=

      new  QuadCurve2D.Double(200,160,195,155,190,150);

      QuadCurve2D leaf\_line4=

      new  QuadCurve2D.Double(200,160,214,155,220,150);

      //利用旋转来绘制花朵:

      AffineTransform trans=new AffineTransform();

      for(int i=0;i<6;i++)

      {   trans.rotate(60\*Math.PI/180,200,200);

          g\_2d.setTransform(trans);

          GradientPaint gradient\_1=

          new GradientPaint(200,200,Color.green,200,100,Color.yellow);

          g\_2d.setPaint(gradient\_1);

          g\_2d.fill(curve\_1);

          GradientPaint gradient\_2=new

          GradientPaint(200,145,Color.green,260,145,Color.red,true);

          g\_2d.setPaint(gradient\_2);

          g\_2d.fill(curve\_2);

          Color c3=new Color(0,200,0); g\_2d.setColor(c3);

          g\_2d.draw(line);

          g\_2d.draw(leaf\_line1); g\_2d.draw(leaf\_line2);

          g\_2d.draw(leaf\_line3); g\_2d.draw(leaf\_line4);

      }

      //花瓣中间的花蕾曲线:

      QuadCurve2D center\_curve\_1=

      new QuadCurve2D.Double(200,200,190,185,200,180);

      AffineTransform trans\_1=new AffineTransform();

      for(int i=0;i<12;i++)

      {  trans\_1.rotate(30\*Math.PI/180,200,200);

         g\_2d.setTransform(trans\_1);

         GradientPaint gradient\_3=

         new GradientPaint(200,200,Color.red,200,180,Color.yellow);

         g\_2d.setPaint(gradient\_3);

         g\_2d.fill(center\_curve\_1);

      }

   }

}

public class E {

   public static void main(String args[]) {

      JFrame f=new JFrame();

      f.setSize(500,450);

      f.setVisible(true);

      Flower canvas=new Flower();

      f.add(canvas,"Center");

      f.validate();

      f.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

  }

}

5． import java.awt.\*;

import javax.swing.\*;

import java.awt.geom.\*;

class Moon extends Canvas

{  public void paint(Graphics g)

   {  Graphics2D g\_2d=(Graphics2D)g;

      Ellipse2D ellipse1=

      new Ellipse2D. Double (20,80,60,60),

      ellipse2=

      new Ellipse2D. Double (40,80,80,80);

      g\_2d.setColor(Color.white);

      Area a1=new Area(ellipse1),

           a2=new Area(ellipse2);

      a1.subtract(a2);             //"差"

      g\_2d.fill(a1);

   }

}

public class E {

   public static void main(String args[]) {

      JFrame f=new JFrame();

      f.setSize(500,450);

      f.setVisible(true);

      Moon moon=new Moon();

      moon.setBackground(Color.black);

      f.add(moon,"Center");

      f.validate();

      f.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

  }

}

# 第15章

一、问答题

1．LinkedList使用链式存储结构，ArrayList使用顺序存储结构。

2．迭代器遍历在找到集合中的一个对象的同时，也得到待遍历的后继对象的引用，因此迭代器可以快速地遍历集合。

3．不是。

4．用HashMap<K,V>来存储。

二、阅读程序

1．8。

2．ABCD。

三、编写程序

1.  import java.util.\*;

public class E {

   public static void main(String args[]) {

      Stack<Integer> stack=new Stack<Integer>();

      stack.push(new Integer(3));

      stack.push(new Integer(8));

      int k=1;

      while(k<=10) {

        for(int i=1;i<=2;i++) {

          Integer F1=stack.pop();

          int f1=F1.intValue();

          Integer F2=stack.pop();

          int f2=F2.intValue();

          Integer temp=new Integer(2\*f1+2\*f2);

          System.out.println(""+temp.toString());

          stack.push(temp);

          stack.push(F2);

          k++;

        }

      }

   }

}

2.   import java.util.\*;

class Student implements Comparable {

   int english=0;

   String name;

   Student(int english,String name) {

      this.name=name;

      this.english=english;

   }

   public int compareTo(Object b) {

      Student st=(Student)b;

      return (this.english-st.english);

   }

}

public class E {

  public static void main(String args[]) {

     List<Student> list=new LinkedList<Student>();

     int score []={65,76,45,99,77,88,100,79};

     String name[]={"张三","李四","旺季","加戈","为哈","周和","赵李","将集"};

     for(int i=0;i<score.length;i++){

             list.add(new Student(score[i],name[i]));

     }

     Iterator<Student> iter=list.iterator();

     TreeSet<Student> mytree=new TreeSet<Student>();

     while(iter.hasNext()){

         Student stu=iter.next();

         mytree.add(stu);

     }

     Iterator<Student> te=mytree.iterator();

     while(te.hasNext()) {

        Student stu=te.next();

        System.out.println(""+stu.name+" "+stu.english);

     }

  }

}

3.  import java.util.\*;

class UDiscKey implements Comparable {

   double key=0;

   UDiscKey(double d) {

     key=d;

   }

   public int compareTo(Object b) {

     UDiscKey disc=(UDiscKey)b;

     if((this.key-disc.key)==0)

        return -1;

     else

        return (int)((this.key-disc.key)\*1000);

  }

}

class UDisc{

    int amount;

    double price;

    UDisc(int m,double e) {

       amount=m;

       price=e;

   }

}

public class E {

   public static void main(String args[ ]) {

      TreeMap<UDiscKey,UDisc>  treemap= new TreeMap<UDiscKey,UDisc>();

      int amount[]={1,2,4,8,16};

      double price[]={867,266,390,556};

      UDisc UDisc[]=new UDisc[4];

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

         UDisc[k]=new UDisc(amount[k],price[k]);

      }

      UDiscKey key[]=new UDiscKey[4] ;

      for(int k=0;k<key.length;k++) {

         key[k]=new UDiscKey(UDisc[k].amount);

      }

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

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

      }

      int number=treemap.size();

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

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

      while(iter.hasNext()) {

         UDisc disc=iter.next();

         System.out.println(""+disc.amount+"G "+disc.price+"元");

      }

      treemap.clear();

      for(int k=0;k<key.length;k++) {

         key[k]=new UDiscKey(UDisc[k].price);

      }

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

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

      }

      number=treemap.size();

      collection=treemap.values();

      iter=collection.iterator();

      while(iter.hasNext()) {

         UDisc disc=iter.next();

         System.out.println(""+disc.amount+"G "+disc.price+"元");

      }

    }

}

# 第16章

1． 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);

         }

   }

}

2． import java.applet.\*;

import java.awt.\*;

public class Rect extends Applet {

   int w,h;

   public void init() {

      String s1=getParameter("width");  //从html得到"width"的值。

      String s2=getParameter("height");  //从html得到"height"的值。

      w=Integer.parseInt(s1);

      h=Integer.parseInt(s2);

   }

   public void paint(Graphics g) {

      g.drawRect(10,10,w,h);

   }

}

/\*<applet code= Rect.class width=280 height=500>

     <Param name="width"  value ="150">

     <Param name="height"  value ="175">

</applet>\*/