实验 1

1. 答：

略

2. 答：

略

3. 答：

略

4. 答：

```
package 实验;


public class 实验1_4 {
  public static void main(String argv[]) {
    System.out.println("C++                    Java");
    System.out.println("    开始        结束");
  }
}
```

5. 答：

```
package 实验;


public class 实验1_5 {
  public static void main(String[] args) {
    System.out.println("Welcome to the Java world!");
  }
}
```

6. 答：

```
package 实验;
```

```
import java.math.*;
import java.util.*;

public class 实验1_6 {
  public static BigInteger compute(int n) {
    if (n <= 1)
      return new BigInteger("" + 1);
    return new BigInteger("" + n).multiply(compute(n - 1));
  }

  public static void main(String[] args) {
    Calendar beginTime = new java.util.GregorianCalendar();
    System.out.println("开始时间" + beginTime);
    System.out.println(
        compute(Integer.parseInt(args[0]))
        );
        Calendar endTime = new java.util.GregorianCalendar();
        System.out.println("结束时间" + beginTime);
        long dd = endTime.getTimeInMillis() - beginTime.getTimeInMillis();
        double seconds = (double) (dd) / 1000;
        System.out.println("时间差是: " + seconds + " (秒) ");
    }
  }
```

7、答:

略

8、答:

略

实验 2

1、答：

```
package 实验.组织;
public class 大学类 {
    String name;
    String address;
    int type;
    大学类() {
    }
}
package 实验.组织;

public class 学院类 {
    大学类  university;
    String name;
    String address;
    int type;
    public 学院类() {
    }
}
package 实验.组织;

public class 系类 {
    学院类  college;
    String name;
    String address;
    int type;
    public 系类() {
    }
}
package 实验.组织;

public class 班级类 {
    学院类  college;
    String name;
    String major;
    int grad;
    public 班级类() {
    }
}
```

```
package 实验.教学;

public class 课程类 {
    String name;
    int credit;
    int hours;
    String teacher;
    String address;
    public 课程类() {
    }
}
package 实验.学生;

import 实验.组织.*;

public class 大学生类 {
    int std_no;
    String name;
    int sex;
    班级类 className;
    public 大学生类() {
    }
}
package 实验.学生;
import 实验.组织.*;
public class 研究生类 {
    int std_no;
    String name;
    int sex;
    班级类 className;
    public 研究生类() {
    }
}
package 实验.学生;

import 实验.组织.*;

public class 博士生类 {
    int std_no;
    String name;
    int sex;
    班级类 className;
```

```
    public 博士生类0{
    }
}

2、答:
import java.util.*;

class 书类 {
    int 编号;
    String 书名;
    String 作者;
    String 出版社;
    float 定价;
    书类(int 编号, String 书名, String 作者, String 出版社, float 定价) {
      this.编号 = 编号;
      this.书名 = 书名;
      this.作者 = 作者;
      this.出版社 = 出版社;
      this.定价 = 定价;
      System.out.println("构造了一本书:" + 书名);
    }
}

class 读者类 {
    int 编号;
    String 姓名;
    读者类(int 编号, String 姓名) {
      this.编号 = 编号;
      this.姓名 = 姓名;
      System.out.println("构造了一个学生:" + 姓名);
    }
}

class 借书记录类 {
    int 读者编号;
    int 书编号;
    int type; //1:借书,0:还书
    Calendar 借还书日期;

    借书记录类(int 读者编号, int 书编号, int type, Calendar 借还书日期
        ) {
      this.读者编号 = 读者编号;
      this.书编号 = 书编号;
      this.type = type;
```

```
        this.借还书日期 = 借还书日期;
        System.out.println("读者" + 读者编号 + "借了书" + 书编号 +
                           ", 借书日期是: " + 借还书日期.get(Calendar.YEAR) +
    "年" +
                           (借还书日期.get(Calendar.MONTH) + 1) + "月" +
                           借还书日期.get(Calendar.DATE) + "日"
            );
    }

    void 还书() {
        this.type = 0;
        this.借还书日期 = new GregorianCalendar();
        System.out.println("读者" + 读者编号 + "还了书" + 书编号 +
                           ", 还书日期是: " + 借还书日期.get(Calendar.YEAR) +
    "年" +
                           (借还书日期.get(Calendar.MONTH) + 1) + "月" +
                           借还书日期.get(Calendar.DATE) + "日"
            );
    }
}

public class 实验2_2 {
    public 实验2_2() {
        书类 书[] = new 书类[3];
        书[0] = new 书类(1, "Java", "卫", "西安交通大学出版社", 34.5F);
        书[1] = new 书类(2, "C++", "谭", "清华大学出版社", 38F);
        书[2] = new 书类(3, "Oracle", "侯", "高等教育出版社", 50F);

        读者类 读者[] = new 读者类[2];
        读者[0] = new 读者类(1, "张三");
        读者[1] = new 读者类(2, "李四");

        借书记录类 借书记录[] = new 借书记录类[5];
        借书记录[0] = new 借书记录类(1, 1, 1, new GregorianCalendar());
        借书记录[1] = new 借书记录类(1, 2, 1, new GregorianCalendar());
        借书记录[2] = new 借书记录类(2, 1, 1, new GregorianCalendar());
        借书记录[3] = new 借书记录类(2, 2, 1, new GregorianCalendar());
        借书记录[4] = new 借书记录类(2, 3, 1, new GregorianCalendar());

        借书记录[0].还书();
        借书记录[1].还书();
        借书记录[4].还书();
    }
```

```
    public static void main(String[] args) {
        实验2_2 实验2_2 = new 实验2_2();
    }
}
```

3、答：

```
package 实验;

import java.util.*;

import java.lang.*;


public class 整数计算器类 {

  public int x;

  整数计算器类(int x) {

    this.x = x;

  }

  public void ten_two() { //十进制转二进制

    int y = x;

    System.out.print("十进制:" + y + " 转二进制:");

    do {

      System.out.print( (y % 2));

      y = y / 2;

    }

    while (y != 0);

    System.out.println();

  }


  public void two_eight() { //二进制转八进制

    System.out.println("二进制转八进制,请输入二进制: ");

    Scanner sc = new Scanner(System.in);

    String a = sc.nextLine();
```

```
        System.out.println("八进制: " +
    Integer.toOctalString(Integer.valueOf(a, 2)));
      }


      public void sixteen_two() { //十六进制转二进制
        System.out.println("十六进制转二进制,请输入十六进制: ");
        Scanner sc = new Scanner(System.in);
        String a = sc.nextLine();
        System.out.println("二进制: " +
    Integer.toBinaryString(Integer.valueOf(a, 16)));
      }


      public 整数计算器类 add(整数计算器类 b) {
        整数计算器类 c = new 整数计算器类(this.x + b.x);
        return c;
      }

      public 整数计算器类 min(整数计算器类 b) {
        整数计算器类 c = new 整数计算器类(this.x - b.x);
        return c;
      }

      public 整数计算器类 mul(整数计算器类 b) {
        整数计算器类 c = new 整数计算器类(this.x * b.x);
        return c;
      }

      public 整数计算器类 dev(整数计算器类 b) {
        整数计算器类 c = new 整数计算器类(this.x / b.x);
        return c;
```

```
    }

    public static void main(String[] args) {
        整数计算器类 a = new 整数计算器类(27);
        a.ten_two();
        a.two_eight();
        a.sixteen_two();
    }
}
```

实验 3

1. 答:

```
public class 实验3_1 {
    public static void main(String args[]) {
        boolean a[] = new boolean[4];
        int tem, flage = 0;
        for (tem = 1; tem <= 4 && flage != 3; tem++) {
            flage = 0;
            a[0] = (tem != 1);
            a[1] = (tem == 3);
            a[2] = (tem == 4);
            a[3] = (tem != 4);
            while (flage != 3) {
                for (int i = 0; i <= 3; i++) {
                    if (a[i]) {
                        flage++;
                    }
                }
                if (flage == 3) {
                    System.out.print("做好事的人是:  ");
                    switch (tem) {
                        case 1:
                            System.out.println("张三! ");
                        case 2:
                            System.out.println("李四! ");
                        case 3:
```

```
                System.out.println("王五！");
            case 4:
                System.out.println("卫六！");
                    }
                }
            }
        }
    }
```

2. 答：

```
public class 实验3_2 {
    static long inPut;
    static int X;
    static boolean outPut;
    static void setN(long inPut) {
        实验3_2.inPut = inPut;
    }

    static long getN() {
        return inPut;
    }

static int getO() {
    return X;
}

public static boolean su(long inPut) {
    int i;
    for (i= 1; i <= inPut / 2; i++) {
        if ( (inPut % i) == 0) {
            break;
        }
    }
    return true;
}

public static long divide(long inPut) {
    long y = 1;
    long m = 2;
    for (m = 2; m <= inPut; m++) {
        if (su(m)) {
            while ( (inPut % m) == 0) {
                inPut = inPut / m;
```

```
            y = y + m;
          }
        }
      }
      return y;
    }

    static int prove(long inPut) {
      boolean a;
      while ( (inPut != 6) && (inPut != 7) && (inPut != 8)) {
        a = su(inPut);
        if (a) {
          inPut = inPut + 1;
        }
        if ( (inPut == 6) || (inPut == 7) || (inPut == 8)) {
          break;
        }
        inPut = divide(inPut);
      }
      return (int) inPut;
    }

    public static void main(String args[]) {
      实验3_2 number1 = new 实验3_2();
      number1.setN( (int) (Math.random() * 15));
      System.out.println("The input integer:  ");
      System.out.println(number1.getN());
      outPut = su(number1.getN());
      if (outPut) {
        if (number1.getN() <= 6) {
          System.out.print("The input number<=6");
        }
        else {
          System.out.print("The input number>=7");
        }
        X = number1.prove(number1.getN());
        System.out.print("The output:  ");
        System.out.println(number1.getO());
      }
      else {
        System.out.print("This number is undivisble!");
        System.out.print("The output:  ");
        System.out.println("number1.prove(number1.getN())");
      }
```

```
        }
    }

3、答：
    import java.util.*;
    import java.lang.*;

class MyException
    extends Exception {
  MyException(String s) {
    super(s);
  }

  MyException() {
    super();
  }
}

public class 实验3_3 {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    try {
      System.out.println("输入被除数：");
      int a = sc.nextInt();
      System.out.println("输入除数：");
      int b = sc.nextInt();
      if (a / b < 1) {
        throw new MyException("商小于1");
      }
      System.out.println("商：" + a / b);
    }
```

```
    catch (NumberFormatException e1) {
      System.out.println(e1);
    }
    catch (ArithmeticException e2) {
      System.out.println(e2);
    }
      catch (Exception e3) {
        System.out.println(e3);
      }
      finally {
        System.out.println("End!");
      }
    }
  }
```

实验 4

1、答：

```
import java.util.*;

public class 实验4_1 { //取出所有的0 的个数
    public static int zero(double a){
        int counter = 0;
        while (a < 1) {
            a = a * 10;
            counter++;
        }
        return counter;
    }

    //去除小数点
    public static long qudian(double a) {
        int i = 0;
        long b = 0;
        while (a % 10 != 0)
```

```
        a = a * 10;
    b = (long) a / 10;
    return b;
}

public static void main(String args[]) {

    int i, j, k, n, m = 0, b = 0, c = 0;
    double x = 1, x1 = 1, x2 = 1;
    int l[] = new int[100];
    System.out.println("请输入所要的行数：");
    Scanner sc = new Scanner(System.in);
    n = sc.nextInt();

    for (i = 1; i <= n; i++) {
      x = x / 2;
      b = zero(x);
      c = zero(2 * x);
      if (b - c > 0)
        m = m + b - c;
      x1 = x;
      for (j = 1; j <= (b - c); j++)
        x1 = x1 * 10;
        System.out.print(".");
        for (k = 1; k < m; k++)
          System.out.print("0");
        System.out.println(qudian(x1));
        x = x1;
      }
    }
  }
```

2. 答：

```
import java.util.*;

class 学生类 {
  int 学号;
  String 姓名;
  int 性别;
  String 班级;
  public 学生类(int 学号, String 姓名, int 性别, String 班级) {
    this.学号 = 学号;
    this.姓名 = 姓名;
    this.性别 = 性别;
```

```
        this.班级 = 班级;
    }
}

public class 实验4_2 {
    public 实验4_20 {
        学生类 学生[] = new 学生类[10];
        学生[0] = new 学生类(1, "张", '男', "电气61");
        学生[1] = new 学生类(2, "李", '男', "电气61");
        学生[2] = new 学生类(3, "王", '女', "电气63");
        学生[3] = new 学生类(4, "卫", '男', "电气66");
        学生[4] = new 学生类(5, "樊", '女', "电气71");
        学生[5] = new 学生类(6, "赵", '男', "能动63");
        学生[6] = new 学生类(7, "冯", '男', "能动65");
        学生[7] = new 学生类(8, "吴", '女', "能动69");
        学生[8] = new 学生类(9, "姚", '女', "能动610");
        学生[9] = new 学生类(10, "董", '男', "信计61");

        System.out.println("请输入学生的姓名: ");
        Scanner sc = new Scanner(System.in);
        String name = sc.nextLine();
        boolean flag = false;
        for (int i = 0; i < 学生.length; i++) {
            if (学生[i].姓名.equals(name)) {
                System.out.println("该学生的学号为: " + 学生[i].学号);
                flag = true;
                break;
            }
        }
        if (!flag) {
            System.out.println("无此人! ");
        }
        for (int i = 0; i < 学生.length - 1; i++) {
            for (int j = i + 1; j < 学生.length; j++) {
                if (学生[j].姓名.compareTo(学生[i].姓名) < 0) {
                    学生类 temp = 学生[i];
                    学生[i] = 学生[j];
                    学生[j] = temp;
                }
            }
        }
        System.out.println("按姓名排序完成! ");

        for (int i = 0; i < 学生.length - 1; i++) {
```

```
        for (int j = i + 1; j < 学生.length; j++) {
            if (学生[j].学号 < 学生[i].学号) {
                学生类 temp = 学生[i];
                学生[i] = 学生[j];
                学生[j] = temp;
            }
        }
    }
        System.out.println("按学号排序完成！");
    }

    public static void main(String[] args) {
        实验4_2 实验4_2 = new 实验4_2();
    }
}
```

3、答：

```
import java.util.*;

class 计数器类 {
    String word;
    int count;
    计数器类(String word, int count) {
        this.word = word;
        this.count = count;
    }
}

public class 实验4_3 {
    public 实验4_3() { //只处理英文
        System.out.println("请输入文章字符串：");
        Scanner sc = new Scanner(System.in);
        String article = sc.nextLine();

        String word[] = article.split(" ,;");
        计数器类 计数器[] = new 计数器类[word.length];
        int count = 0;

        for (int i= 0; i < word.length; i++) {
            for (int j = 0; j < count; j++) {
                if (word[i].compareTo(计数器[j].word) == 0) {
                    计数器[j].count++;
                }
                else {
```

```
                计数器[count++] = new 计数器类(word[i], 1);
            }
        }
    }
    for (int i = 0; i < Math.min(10, 计数器.length); i++) {
        System.out.print(计数器[i].word + ",");
    }
}

    public static void main(String[] args) {
        实验4_3 实验4_3 = new 实验4_3();
    }
}
```

实验 5

1. 答：

```
import java.util.*;

class 计数器类 {
    String word;
    int count;
    计数器类(String word, int count) {
        this.word = word;
        this.count = count;
    }
}

public class 实验5_1 {
    public 实验5_1() {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入文章字符串：");
        String article = sc.nextLine();

        String word[] = article.split(" ,;");
        计数器类 计数器;
        Vector vector = new Vector();
        int count = 0;

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

```
        for (int j = 0; j < vector.size(); j++) {
            计数器 = (计数器类) (vector.get(j));
            if (word[i].compareTo(计数器.word) == 0) {
                计数器.count++;
                vector.set(j, 计数器);
            }
            else {
                计数器 = new 计数器类(word[i], 1);
                vector.add(计数器);
            }
        }
    }
    for (int i = 0; i < vector.size(); i++) {
        计数器 = (计数器类) (vector.get(i));
        System.out.print(计数器.word + " 的数量为" + 计数器.count);
    }
  }

  public static void main(String[] args) {
    实验5_1 实验5_1 = new 实验5_1();
  }
}
```

2、答：

```
import java.util.*;

class 节日类 {
  String 日期;
  String 节气;
  节日类(String 日期, String 节气) {
    this.日期 = 日期;
    this.节气 = 节气;
  }
}

public class 实验5_2 {
  public 实验5_2() {
    Scanner sc = new Scanner(System.in);
    System.out.println("请输入年月日：");
    int year = sc.nextInt();
    int month = sc.nextInt();
    int day = sc.nextInt();

    节日类 节日[] = new 节日类[10]; //只放了10个实验数据
```

```
节日[0] = new 节日类("1-1", "元旦");
节日[1] = new 节日类("3-8", "妇女节");
节日[2] = new 节日类("5-1", "劳动节");
节日[3] = new 节日类("5-4", "青年节");
节日[4] = new 节日类("6-1", "儿童节");

节日[5] = new 节日类("1-1", "元旦");
节日[6] = new 节日类("8-1", "建军节");
节日[7] = new 节日类("9-10", "教师节");
节日[8] = new 节日类("10-1", "国庆节");
节日[9] = new 节日类("2-14", "情人节");

for (int i = 0; i < 节日.length; i++) {
   String s = month + "-" + day;
   if (节日[i].日期.compareTo(s) == 0) {
      System.out.println(节日[i].节气);
      break;
   }
}
}
```

```
   public static void main(String[] args) {
      new 实验5_20;
   }
}
```

3. 答：

```
public class 实验5_3 {
  public 实验5_30 {
     int a[] = new int[6];
     int m1, m2, m3, m4, m5, m6;
     for (int i = 0; i < 6; i++)
        a[i] = 0;
     for (; a[0] < 2; a[0]++)
  for (; a[1] < 2; a[1]++)
    for (; a[2] < 2; a[2]++)
      for (; a[3] < 2; a[3]++)
        for (; a[4] < 2; a[4]++)
          for (; a[5] < 2; a[5]++) {
            if (a[3] == 0)
               a[4] = 0;
            m1 = a[1] + a[0];
            m2 = a[0] + a[3];
            m3 = a[0] + a[4] + a[5];
```

```
            m4 = a[1] + a[2];
            m5 = a[2] + a[3];

            if (m1 != 0 && m2 != 2 && m3 == 2 && m4 != 1 && m5 == 1)

                for (int k = 0; k < 6; k++)
                    System.out.println(a[k]);
        }
    }

    public static void main(String[] args) {
        实验5_3 实验5_3 = new 实验5_3();
    }
}
```

4. 答:

```
import java.util.*;

class 实验5_4 {
  static int plus(int a, int b) { //定义一个方法用于计算 i+.....+j 的和
    int s = 0;
    for (int i = a; i <= b; i++) {
      s = s + i;
    }
    return (s);
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int a = sc.nextInt();
    for (int i = 1; i < a / 2 + 1; i++) {
      for (int j = i; j < a / 2 + 2 + 1; j++) {
        if (plus(i, j) == a) {
          System.out.print("" + a + "=" + i);
          for (int k = i + 1; k <= j; k++) {
            System.out.print("" + "+" + k);
          }
          System.out.println("");
        }
      }
    }
  }
}
```

实验 6

1、答：

```
class 计算机书类
      extends 书类 {
   int 打折;
   int 类型;
}

class 数学书类
      extends 书类 {
   boolean 是名著吗;

}

class 文学书类
      extends 书类 {
   String 样本章;
}

class 社科书类
      extends 书类 {
   String 观点;
   }

class 外文书类
      extends 书类 {
   int 语种;
   }

class 杂类书类
      extends 书类 {
   int 打折;
   int 类型;
}

public class 实验6_1 {
   public 实验6_10 {
   }

   public static void main(String[] args) {
      new 实验6_10;
```

```
    }
  }

2. 答:
  import java.awt.*;

  abstract class 形状 {
    Color 颜色;
    public abstract void 画(Graphics g);
  }

  class 点
      extends 形状 {
    Point 坐标;
    public void 画(Graphics g) {
      g.setColor(颜色);
      g.fillOval(坐标.x - 1, 坐标.y - 1, 2, 2);
    }
  }

  class 线段
      extends 形状 {
    点 起始点;
    点 结束点;
    public void 画(Graphics g) {
      g.setColor(颜色);
      g.drawLine(起始点.坐标.x, 起始点.坐标.y,
                 结束点.坐标.x, 结束点.坐标.y);
    }
  }

  class 矩形
      extends 形状 {
    点 起始点;
    点 结束点;
    public void 画(Graphics g) {
      g.setColor(颜色);
      g.drawRect(起始点.坐标.x, 起始点.坐标.y,
                 结束点.坐标.x, 结束点.坐标.y);
    }
  }

  class 椭圆
      extends 形状 {
```

```
点 起始点;
点 结束点;
public void 画(Graphics g){
    g.setColor(颜色);
    g.drawOval(起始点.坐标.x, 起始点.坐标.y,
                    结束点.坐标.x, 结束点.坐标.y);
}
}

public class 实验6_2{
  public 实验6_20{
  }

  public static void main(String[] args){
     new 实验6_20;
  }
}
```

实验 7

1. 答：

```
import java.util.*;

class 时钟显示定时器任务类
      extends TimerTask{
   public void run(){
      GregorianCalendar now=new GregorianCalendar();
      System.out.println("现在是: " + now.get(Calendar.YEAR) + "年" +
                                 (now.get(Calendar.MONTH) + 1) + "月" +
                                 now.get(Calendar.DATE) + "日" +
                                 now.get(Calendar.HOUR) + "时" +
                                 now.get(Calendar.MINUTE) + "分" +
                                 now.get(Calendar.SECOND) + "秒");
   }
}

class 秒表显示定时器任务类
      extends TimerTask{
  long beingSecond;
  秒表显示定时器任务类0{
```

```
        beingSecond = 0;
    }

    public void run() {
        System.out.println("已经过了 " + beingSecond++ +"秒");
    }
}

public class 实验7_1 {
    public 实验7_10 {
        Scanner scanner = new Scanner(System.in);
        时钟显示定时器任务类 myAddTimerTask1 = new 时钟显示定时器任务类();
        秒表显示定时器任务类 myAddTimerTask2 = new 秒表显示定时器任务类();
        Timer myTimer = new Timer();
        myTimer.schedule(myAddTimerTask1, 1000, 1000);
        myTimer.schedule(myAddTimerTask2, 1000, 1000);
    }

    public static void main(String[] args) {
        new 实验7_10();
    }
}
```

2. 答：

```
被加数 类：

class 被加数
      extends Thread {

    private static int start;

    private static int end;

    public static int 被加数;

    static{

        start=1;

        end=50;

        被加数=1;

    }


public void run() {

    System.out.println("\r\n被加数线程开始运行");
```

```
for (int i = start; i <= end; i++) {
  System.out.print("产生被加数:" + i+"\n");
  被加数=i;
  try{
     sleep(100);
   }catch(InterruptedException e){
        System.out.println("被加数进程中段错误！");
      }
  }
  被加数=0;
  System.out.println("\r\n被加数线程运行结束");
 }
}
加数 类:
class 加数
  extends Thread {
private static int start;
private static int end;
public static int 加数;
static{
   start=1;
   end=50;
   加数=1;
 }
public void run() {
  System.out.println("\r\n加数线程开始运行");
  for (int i = start; i <= end; i++) {
    System.out.print("产生加数:" + i+"\n");
    加数=i;
```

```
        try{
            sleep(100);
          }catch(InterruptedException e){
              System.out.println("加数进程中段错误！");
          }
    }
        加数=0;
        System.out.println("\r\n加数线程运行结束");
      }
    }
    求和 类：
    class 求和
        extends Thread {
            private static int 和;
      static{
        和=0;
      }

    public void run() {
        System.out.println("\r\n求和线程开始运行");
        for (;;) {
          和=加数.加数+被加数.被加数;
          if(和==0)break;
          else System.out.print("计算和=" + 和+"\n");
          try{
            sleep(100);
          }catch(InterruptedException e){
              System.out.println("求和进程中段错误！");
          }
    }
```

```
        System.out.println("\r\n求和线程运行结束");
    }
}
```

ThreadTester 类:

```
public class ThreadTester{
    public static void main(String args[]){
        被加数 A=new 被加数();
        加数 B=new 加数();
        求和 Sum=new 求和();
        A.start();
        B.start();
        Sum.start();
    }
}
```

实验 8

1. 答:

```
import java.io.*;

public class StudentsInformation
    implements Serializable {
  private String 姓名;
  private String 学号;
  private String 性别;
  private String 院系班级;
  private String 联系方式;

  public StudentsInformation() {
    姓名 = null;
```

```
        学号 = null;
        性别 = null;
        院系班级 = null;
        联系方式 = null;
    }

    public StudentsInformation(String name, String num, String sex,
String grade, String call) {
        姓名 = name;
        学号 = num;
        性别 = sex;
        院系班级 = grade;
        联系方式 = call;
    }

    public void display() {
      System.out.println("姓名:" + 姓名);
      System.out.println("学号:" + 学号);
      System.out.println("性别:" + 性别);
      System.out.println("院系班级:" + 院系班级);
      System.out.println("联系方式:" + 联系方式);
    }

    public String get姓名() {
      return 姓名;
    }

    public String get院系班级() {
      return 院系班级;
    }
```

```
    public String get联系方式() {
      return 联系方式;
    }

    public String get性别() {
      return 性别;
    }

    public String get学号() {
      return 学号;
    }

    public void set姓名(String 姓名) {
      this.姓名 = 姓名;
    }

    public void set学号(String 学号) {
      this.学号 = 学号;
    }

    public void set性别(String 性别) {
      this.性别 = 性别;
    }

    public void set联系方式(String 联系方式) {
      this.联系方式 = 联系方式;
    }

    public void set院系班级(String 院系班级) {
```

```
        this.院系班级 = 院系班级;
    }

    private void readObject(ObjectInputStream ois) throws IOException,
            ClassNotFoundException {
        ois.defaultReadObject();
    }

    private void writeObject(ObjectOutputStream oos) throws IOException
{
        oos.defaultWriteObject();
    }

}
```

Function 类:

```
import java.io.*;
import java.util.*;
import java.io.IOException;

public class Function {
  static Scanner sc = new Scanner(System.in);
  static String choice = sc.nextLine();
  public static void show(StudentsInformation students[], int size) {
    for (int i = 0; i < size; i++) {
      if (students[i].get姓名() != null) {
        System.out.println(
            "**********学生信息**********
**");
        students[i].display();
      }
```

```
        }
    }

    public static StudentsInformation[] Add(StudentsInformation
students[],
                                            int size) {
        int s = size + 1;
        StudentsInformation stu[] = new StudentsInformation[s];
        int i;
        for (i = 0; i < (s - 1); i++) {
            stu[i] = new StudentsInformation();
            stu[i].set姓名(students[i].get姓名());
            stu[i].set学号(students[i].get学号());
            stu[i].set性别(students[i].get性别());
            stu[i].set院系班级(students[i].get院系班级());
            stu[i].set联系方式(students[i].get联系方式());
        }
        System.out.println("请输入新增学生的全部信息:");
        stu[s - 1] = new StudentsInformation();
        System.out.print("姓名:");
        stu[s - 1].set姓名(sc.nextLine());
        System.out.print("学号:");
        stu[s - 1].set学号(sc.nextLine());
        System.out.print("性别:");
        stu[s - 1].set性别(sc.nextLine());
        System.out.print("院系班级:");
        stu[s - 1].set院系班级(sc.nextLine());
        System.out.print("联系方式:");
        stu[s - 1].set联系方式(sc.nextLine());
        return stu;
```

```
    }

    public static void Change(StudentsInformation students[], int size)
{
        System.out.println("请输入修改学生的姓名及其全部信息:");
        StudentsInformation stu = new StudentsInformation();
        System.out.print("姓名:");
        stu.set姓名(sc.nextLine());
        System.out.print("学号:");
        stu.set学号(sc.nextLine());
        System.out.print("性别:");
        stu.set性别(sc.nextLine());
        System.out.print("院系班级:");
        stu.set院系班级(sc.nextLine());
        System.out.print("联系方式:");
        stu.set联系方式(sc.nextLine());
        for (int i = 0; i < size; i++) {
          String temp = students[i].get姓名();
          if (temp.equals(stu.get姓名())) {
            students[i].set姓名(stu.get姓名());
            students[i].set学号(stu.get学号());
            students[i].set性别(stu.get性别());
            students[i].set院系班级(stu.get院系班级());
            students[i].set联系方式(stu.get联系方式());
          }
        }
    }

    public static void Delete(StudentsInformation students[], int size)
```

```
{
        System.out.println("请输入所要删除的学生姓名：");
        String name;
        name = sc.nextLine();
        for (int i = 0; i < size; i++) {
          if (name.equals(students[i].get姓名())) {
            students[i].set姓名(null);
            students[i].set学号(null);
            students[i].set性别(null);
            students[i].set院系班级(null);
            students[i].set联系方式(null);
          }
        }
      }
      public static void Savefile(StudentsInformation students[], int
size) throws
          IOException {
        System.out.print("保存信息? (Yes/No)");
        choice = sc.nextLine();
        if (choice.equals("Yes")) {
          System.out.print("请输入新建的文件的名称：");
          String filename = sc.nextLine();
          FileWriter fw = new FileWriter(filename);
          for (int i = 0; i < size; i++) {
            if (students[i].get姓名() != null) {
              fw.write(students[i].get姓名());
              fw.write('\n');
            }
            if (students[i].get学号() != null) {
```

```
            fw.write(students[i].get学号());
            fw.write('\n');
        }
        if (students[i].get性别() != null) {
            fw.write(students[i].get性别());
            fw.write('\n');
        }
        if (students[i].get院系班级() != null) {
            fw.write(students[i].get院系班级());
            fw.write('\n');
        }
        if (students[i].get联系方式() != null) {
            fw.write(students[i].get联系方式());
            fw.write('\n');
        }
            }
            fw.close();
        }
    }

}
```

Workmain 类:

```
import java.io.*;
import java.util.*;
import java.io.IOException;

public class Workmain {
    public static void main(String args[]) {
        System.out.print("请输入需要建立的学生信息的个数:");
        Scanner sc = new Scanner(System.in);
```

```
        int size = sc.nextInt();
        sc.nextLine(); //清除键盘缓存
        StudentsInformation students[] = new StudentsInformation[size];
        for (int i = 0; i < size; i++) {
            students[i] = new StudentsInformation();
            System.out.println("学生" + (i + 1) + ":");
            System.out.print("姓名:");
            students[i].set姓名(sc.nextLine());
            System.out.print("学号:");
            students[i].set学号(sc.nextLine());
            System.out.print("性别:");
            students[i].set性别(sc.nextLine());
            System.out.print("院系班级:");
            students[i].set院系班级(sc.nextLine());
            System.out.print("联系方式:");
            students[i].set联系方式(sc.nextLine());
        }
        Function.show(students, size);
        for (; ; ) {
            System.out.print("增加新学生信息? (Yes/No)");
            String choice = sc.nextLine();
            if (choice.equals("Yes")) {
                students = Function.Add(students, size);
                size++;
                Function.show(students, size);
            }
            else {
                break;
            }
        }
```

```
        for (; ; ) {
            System.out.print("修改学生信息?(Yes/No)");
            String choice = sc.nextLine();
            if (choice.equals("Yes")) {
                Function.Change(students, size);
                Function.show(students, size);
            }
            else {
                break;
            }
        }
        for (; ; ) {
            System.out.print("删除学生信息? (Yes/No)");
            String choice = sc.nextLine();
            if (choice.equals("Yes")) {
                Function.Delete(students, size);
                Function.show(students, size);
            }
            else {
                break;
            }
        }
        try {
            Function.Savefile(students, size);
        }
        catch (IOException e) {
            System.out.println(e);
        }
    }
}
```

2、答：

```
import java.io.*;
import java.util.*;

public class 实验8_2 {
    Stack dirStack = new Stack();
    String dName;
    String driver;

public 实验8_20 {
    System.out.print("请输入待查找的文件或目录名：");
    Scanner scanner = new Scanner(System.in);
    dName = scanner.nextLine(); //从键盘输入目录名
    dName = dName.trim();
    System.out.print("请输入要搜索的磁盘：");
    driver = scanner.nextLine(); //从键盘输入盘符
    driver = driver.trim();
    if (dName.length() > 0 && driver.length() > 0) {
        dirStack.push(newFile(driver));
        findSubdir();
    }
}

private void findSubdir() {
    while (! (dirStack.empty())) {
        try {
            File file = (File) dirStack.pop();
            String fileList[];
            fileList = file.list();

            for (int i = 0; i < fileList.length; i++) {
                System.out.println(fileList[i]);
                if (fileList[i].indexOf(dName) >= 0) {
                    System.out.println(fileList[i]);
                }
                if (! (new File(fileList[i]).isFile())) {
                    dirStack.push(new File(driver + "\\" + fileList[i]));
                }
            }
        }
        catch (Exception e) {
        }
    }
```

```
    }

    public static void main(String[] args) {
        new 实验8_20();
    }
}
```

实验 9

1、答：

略

2、答：

```
import java.awt.*;
import javax.swing.*;

import java.awt.Point;
import java.awt.*;
import java.awt.event.*;

class QiPan {//棋盘类
    private Point point; //棋盘交叉点坐标
    private int which; //棋子类型：-1—无子,0—黑子,1-白子

    QiPan() {
        point = null;
        which = -1;
    }

    public Point getPoint() {
        return point;
    }

    public int getWhich() {
        return which;
    }

    public void setPoint(Point point) {
        this.point = point;
    }
```

```
    public void setWhich(int which) {
        this.which = which;
    }
}

class QiZi { //棋子类
    private Point point; //棋子中心点坐标
    private static int r = 12; //棋子半径
    private int which; //棋子类型：0—黑子,1-白子
    private boolean exist; //状态：true-已落子,false-未下

    QiZi() {
        point = null;
        which = 0;
        exist = false;
    }

    public boolean isExist() {
        return exist;
    }

    public Point getPoint() {
        return point;
    }

    static public int getR() {
        return r;
    }

    public int getWhich() {
        return which;
    }

    public void setExist(boolean exist) {
        this.exist = exist;
    }

    public void setPoint(Point point) {
        this.point = point;
    }

    static public void setR(int r) {
        QiZi.r = r;
```

```
    }

    public void setWhich(int which) {
        this.which = which;
    }
}

public class 实验9_2
    extends JFrame {
QiPan[][] qiPan = new QiPan[19][19];
QiZi[][] qiZi = new QiZi[2][182];
int count[] = {
    0, 0};
int turn = 0;

BorderLayout borderLayout1 = new BorderLayout();

public 实验9_20 {
  try {
    jbInit();
  }
  catch (Exception exception) {
    exception.printStackTrace();
  }
}

private void jbInit() throws Exception {
  getContentPane().setLayout(borderLayout1);
  for (int i = 0; i < 19; i++) { //棋盘初始化
    for (int j = 0; j < 19; j++) {
      qiPan[i][j] = new QiPan();
      qiPan[i][j].setPoint(new Point(25 + 30 * j, 25 + 30 * i));
    }
  }
  this.addMouseListener(new LeftMousePressedDaemon()); //增加鼠标事件监听器
  this.setLocation(200, 100);
  this.setSize(580, 580);
  this.setVisible(true);
  this.addWindowListener(new 实验9_2_this_windowAdapter(this));
}

public void paint(Graphics g) {
  g.setColor(Color.BLUE);
```

```
for (int i = 0; i < 18; i++) { //显示棋盘
    for (int j = 0; j < 18; j++) {
        g.drawRect(qiPan[i][j].getPoint().x, qiPan[i][j].getPoint().y,
                30, 30);
    }
}

for (int i = 0; i < 2; i++) { //显示棋子
    for (int j = 0; j < count[i]; j++) {
        if (qiZi[i][j].isExist()) {
            if (i == 0) {
                g.setColor(Color.RED);
            }
            else {
                g.setColor(Color.GREEN);
            }
            g.fillOval(qiZi[i][j].getPoint().x, qiZi[i][j].getPoint().y,
                    2 * QiZi.getR(), 2 * QiZi.getR());
        }
    }
}
}

boolean isWin(int i, int j) {
    return false;
}

class LeftMousePressedDaemon
        extends MouseAdapter { //鼠标按下事件监听器类

    public void mousePressed(MouseEvent mE) {
        if (mE.getButton() == MouseEvent.BUTTON1) {
            boolean flag = false;
            Rectangle rectangle;
            int i = 0;
            int j = 0;
            exit:
                for (i = 0; i < 19; i++) {
                    for (j = 0; j < 19; j++) {
                        rectangle = new Rectangle(
                                qiPan[i][j].getPoint().x - 15,
                                qiPan[i][j].getPoint().y - 15, 30, 30);
                        if (rectangle.contains(mE.getPoint())) { //判断鼠标点击位置的合理性
                            flag = true;
```

```
                    break exit;
                }
            }
        }
        if (flag == false) { //鼠标点击位置不合理
            return;
        }
        if (qiPan[i][j].getWhich() != -1) { //鼠标点击位置已有子
            return;
        }
        ////鼠标点击位置合理是进行以下工作：
        qiPan[i][j].setWhich(turn);
        qiZi[turn][count[turn]] = new QiZi();
        qiZi[turn][count[turn]].setPoint(
                new Point(qiPan[i][j].getPoint().x - QiZi.getR(),
                        qiPan[i][j].getPoint().y - QiZi.getR()));
        qiZi[turn][count[turn]].setWhich(turn);
        qiZi[turn][count[turn]].setExist(true);

        Graphics g = getGraphics();
        if (turn == 0) {
            g.setColor(Color.RED);
        }
        else {
            g.setColor(Color.GREEN);
        }
        g.fillOval(qiZi[turn][count[turn]].getPoint().x,
                qiZi[turn][count[turn]].getPoint().y,
                2 * QiZi.getR(), 2 * QiZi.getR());
        count[turn]++;
        turn = 1 - turn; //轮到对方
      }
    }
}

public static void main(String[] args) {
    实验9_2 实验9_2 = new 实验9_2();
}

public void this_windowClosing(WindowEvent e) {
    System.exit(-1);
}
}
```

```
class 实验9_2_this_windowAdapter
    extends WindowAdapter {
  private 实验9_2 adaptee;
  实验9_2_this_windowAdapter(实验9_2 adaptee) {
    this.adaptee = adaptee;
  }

  public void windowClosing(WindowEvent e) {
      adaptee.this_windowClosing(e);
    }
  }
```

3、答：

```
package 实验;

import java.awt.*;
import javax.swing.*;
import java.awt.Rectangle;
import java.util.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class Record {
  long id;
  long beginMSeconds;
  long endMSeconds;
  Record(long id) {
    this.id = id;
    beginMSeconds = new GregorianCalendar().getTimeInMillis();
    endMSeconds = beginMSeconds;
  }

  public String toString() {
    return "" + id + "-" + beginMSeconds + "-" + endMSeconds;
  }
}

public class 实验9_3
    extends JFrame {
  public 实验9_3() {
    try {
      jbInit();
    }
    catch (Exception exception) {
```

```
            exception.printStackTrace();
        }
    }

    private void jbInit() throws Exception {
        getContentPane().setLayout(null);
        jLabel1.setToolTipText("");
        jLabel1.setText("正在上机人员清单");
        jLabel1.setBounds(new Rectangle(40, 13, 132, 26));
        jScrollPane1.setBounds(new Rectangle(16, 57, 186, 237));
        jScrollPane2.setBounds(new Rectangle(257, 57, 187, 238));
        jLabel3.setText("请输入条形码：");
        jLabel3.setBounds(new Rectangle(37, 310, 88, 24));
        jTextField1.setBounds(new Rectangle(128, 312, 150, 19));
        jButton1.setBounds(new Rectangle(347, 312, 74, 19));
        jButton1.setText("确认");
        jButton1.addActionListener(new 实验9_3_jButton1_actionAdapter(this));
        this.getContentPane().add(jScrollPane1);
        jScrollPane1.getViewport().add(jList1);
        this.getContentPane().add(jScrollPane2);
        this.getContentPane().add(jButton1);
        this.getContentPane().add(jLabel3);
        this.getContentPane().add(jTextField1);
        this.getContentPane().add(jLabel1);
        this.getContentPane().add(jLabel2);
        jScrollPane2.getViewport().add(jList2);
        jLabel2.setToolTipText("");
        jLabel2.setText("已经下机人员清单");
        jLabel2.setBounds(new Rectangle(282, 14, 132, 26));

        this.setLocation(200, 200);
        this.setSize(500, 360);
        this.setVisible(true);
    }

    public static void main(String[] args) {
        实验9_3 实验9_3_对象 = new 实验9_3();

    }

    JLabel jLabel1 = new JLabel();
    JLabel jLabel2 = new JLabel();
    JScrollPane jScrollPane1 = new JScrollPane();
    JScrollPane jScrollPane2 = new JScrollPane();
```

```
JList jList1 = new JList();
JList jList2 = new JList();
JLabel jLabel3 = new JLabel();
JTextField jTextField1 = new JTextField();
JButton jButton1 = new JButton();
Vector 正在上机人员列表 = new Vector();
Vector 已经下机人员列表 = new Vector();
public void jButton1_actionPerformed(ActionEvent e) {
  boolean flag = false;
  long newid;
  try {
    newid = Long.parseLong(jTextField1.getText());
  }
  catch (Exception e2) {
    newid = 0;
  }
  for (int i = 0; i < 正在上机人员列表.size(); i++) {
    Record record = (Record) 正在上机人员列表.get(i);
    long id = record.id;
    if (newid == id) {
      record.endMSeconds = new GregorianCalendar().getTimeInMillis();
      已经下机人员列表.add(record);
      正在上机人员列表.remove(i);
      this.jList1.setListData(正在上机人员列表);
      this.jList2.setListData(已经下机人员列表);
      flag = true;
      break;
    }
  }
  if (!flag) {
    Record record = new Record(newid);
    正在上机人员列表.add(record);
    this.jList1.setListData(正在上机人员列表);

  }
}

class 实验9_3_jButton1_actionAdapter
    implements ActionListener {
  private 实验9_3 adaptee;
  实验9_3_jButton1_actionAdapter(实验9_3 adaptee) {
    this.adaptee = adaptee;
```

```
    }

    public void actionPerformed(ActionEvent e) {
        adaptee.jButton1_actionPerformed(e);
    }
}
```

实验 10

1、答：

略

2、答：

```
package 实验;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
import java.util.*;
import java.net.*;

public class 实验10_2
    extends JApplet {
  JScrollPane jScrollPane1 = new JScrollPane();
  JList links;

  public void init() {
    setLayout(new BorderLayout());
    Vector vector = new Vector();
    int i = 1;
    String s;
    while ( (s = getParameter("site" + i)) != null) {
      vector.add(s);
      i++;
    }
    links = new JList(vector);
    links.addMouseListener(new MyMouseAdapter());
    jScrollPane1.getViewport().add(links);
    this.getContentPane().add(jScrollPane1, BorderLayout.CENTER);
```

```
    }

    class MyMouseAdapter
        extends MouseAdapter {
      public void mouseClicked(MouseEvent e) {
        AppletContext context = getAppletContext();
        try {
          context.showDocument(new URL( (String) links.getSelectedValue()),
                               "sitewin");
          showStatus("正在访问网站: " + links.getSelectedValue() + "...");
        }
        catch (Exception e2) {
          showStatus("网站错误: ");
        }
      }
    }
}
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=GBK">
<title>
我收藏的好网站
</title>
</head>
<body>
我收藏的好网站<br>
<applet
  codebase = "."
  code    = "实验.实验10_2.class"
  name    = "实验10_2"
  width   = "400"
  height  = "300"
  hspace  = "0"
  vspace  = "0"
  align   = "middle"
>
  <param name=site1 value="http://java.sun.com">
  <param name=site2 value="http://www.sf.net">
  <param name=site3 value="http://www.apache.org">
  <param name=site4 value="http://www.mysql.com">
  <param name=site5 value="http://www.eclipse.com">
  <param name=site6 value="http://www.bluej.org">
  <param name=site7 value="http://www.cjsdn.net">
</applet>
```

```
    </body>
    </html>
```

3、答：

```
    package 实验;

    import java.awt.*;
    import java.awt.event.*;
    import java.applet.*;
    import javax.swing.*;

    public class 实验10_3
        extends JApplet {
      boolean isStandalone = false;
      BorderLayout borderLayout1 = new BorderLayout();

      JLabel jLabel1 = new JLabel();
      JTextField num1 = new JTextField();
      JLabel jLabel2 = new JLabel();
      JTextField num2 = new JTextField();
      JButton add = new JButton();
      JButton sub = new JButton();
      JButton mul = new JButton();
      JButton div = new JButton();
      JButton dec = new JButton();
      JButton oct = new JButton();
      JButton hex = new JButton();
      JButton jButton8 = new JButton();
      JButton jButton9 = new JButton();
      JButton sqrt = new JButton();
      JButton cos = new JButton();
      JButton sin = new JButton();
      JButton abs = new JButton();
      JLabel jLabel3 = new JLabel();
      JTextField num3 = new JTextField();

    public String getParameter(String key, String def) {
       return isStandalone ? System.getProperty(key, def) :
            (getParameter(key) != null ? getParameter(key) : def);
    }

    public 实验10_3() {
       try {
          jbInit();
```

```
    }
    catch (Exception ex) {
      ex.printStackTrace();
    }
  }

  public void init() {
    try {
      jbInit();
    }
    catch (Exception e) {
      e.printStackTrace();
    }
  }

  private void jbInit() throws Exception {
    this.getContentPane().setLayout(null);
    jLabel1.setToolTipText("");
    jLabel1.setText("第 1 个数：");
    jLabel1.setBounds(new Rectangle(56, 41, 65, 28));
    num2.setText("0");
    num2.setBounds(new Rectangle(337, 47, 127, 24));
    jLabel2.setToolTipText("");
    jLabel2.setText("第 2 个数：");
    jLabel2.setBounds(new Rectangle(272, 42, 65, 28));
    mul.setBounds(new Rectangle(279, 167, 83, 25));
    mul.setText("*");
    mul.addActionListener(new 实验 10_3_mul_actionAdapter(this));
    div.setBounds(new Rectangle(392, 167, 83, 25));
    div.setText("/");
    div.addActionListener(new 实验 10_3_div_actionAdapter(this));
    dec.setBounds(new Rectangle(52, 267, 83, 25));
    dec.setText("dec");
    dec.addActionListener(new 实验 10_3_dec_actionAdapter(this));
    oct.setBounds(new Rectangle(166, 267, 83, 25));
    oct.setText("oct");
    oct.addActionListener(new 实验 10_3_oct_actionAdapter(this));
    hex.setBounds(new Rectangle(279, 267, 83, 25));
    hex.setText("hex");
    hex.addActionListener(new 实验 10_3_hex_actionAdapter(this));
    jButton8.setBounds(new Rectangle(392, 268, 83, 25));
    jButton8.setText("bin");
    jButton8.addActionListener(new 实验 10_3_jButton8_actionAdapter(this));
    jButton9.setBounds(new Rectangle(321, 210, 83, 25));
```

```
jButton9.setText("jButton9");
sqrt.setBounds(new Rectangle(392, 216, 83, 25));
sqrt.setText("sqrt");
sqrt.addActionListener(new 实验10_3_sqrt_actionAdapter(this));
cos.setBounds(new Rectangle(166, 218, 83, 25));
cos.setText("cos");
cos.addActionListener(new 实验10_3_cos_actionAdapter(this));
sin.setBounds(new Rectangle(52, 222, 83, 25));
sin.setText("sin");
sin.addActionListener(new 实验10_3_sin_actionAdapter(this));
abs.setBounds(new Rectangle(279, 215, 83, 25));
abs.setText("abs");
abs.addActionListener(new 实验10_3_abs_actionAdapter(this));
sub.setBounds(new Rectangle(166, 167, 83, 25));
sub.setText("-");
sub.addActionListener(new 实验10_3_sub_actionAdapter(this));
add.setBounds(new Rectangle(52, 167, 83, 25));
add.setText("+");
add.addActionListener(new 实验10_3_add_actionAdapter(this));
num3.setText("0");
num3.setBounds(new Rectangle(235, 103, 127, 24));
jLabel3.setToolTipText("");
jLabel3.setText("第3个数：");
jLabel3.setBounds(new Rectangle(175, 99, 65, 28));
num1.setText("0");
this.getContentPane().add(num1);
this.getContentPane().add(jLabel1);
this.getContentPane().add(abs);
this.getContentPane().add(mul);
this.getContentPane().add(add);
this.getContentPane().add(sub);
this.getContentPane().add(div);
this.getContentPane().add(oct);
this.getContentPane().add(dec);
this.getContentPane().add(hex);
this.getContentPane().add(jButton8);
this.getContentPane().add(sin);
this.getContentPane().add(cos);
this.getContentPane().add(sqrt);
this.getContentPane().add(jLabel2);
this.getContentPane().add(num2);
this.getContentPane().add(num3);
this.getContentPane().add(jLabel3);
num1.setBounds(new Rectangle(119, 47, 127, 24));
```

```
        this.setSize(500, 500);
        this.setLocation(100, 100);
        this.setVisible(true);
    }

    public String getAppletInfo() {
        return "Applet Information";
    }

    public String[][] getParameterInfo() {
        return null;
    }

    public void add_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        double v2 = Double.parseDouble(num2.getText());
        double v3 = v1 + v2;
        num3.setText("" + v3);
    }

    public void sub_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        double v2 = Double.parseDouble(num2.getText());
        double v3 = v1 - v2;
        num3.setText("" + v3);

    }

    public void mul_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        double v2 = Double.parseDouble(num2.getText());
        double v3 = v1 * v2;
        num3.setText("" + v3);
    }

    public void div_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        double v2 = Double.parseDouble(num2.getText());
        double v3 = v1 / v2;
        num3.setText("" + v3);
    }

    public void sin_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
```

```
        double v3 = Math.sin(v1);
        num3.setText("" + v3);
    }

    public void cos_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        double v3 = Math.cos(v1);
        num3.setText("" + v3);
    }

    public void abs_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        double v3 = Math.abs(v1);
        num3.setText("" + v3);
    }

    public void sqrt_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        double v3 = Math.sqrt(v1);
        num3.setText("" + v3);
    }

    public void dec_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        double v3 = v1;
        num3.setText("" + v3);
    }

    public void oct_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        int v3 = (int) v1;
        num3.setText(Integer.toOctalString(v3));
    }

    public void hex_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        int v3 = (int) v1;
        num3.setText(Integer.toHexString(v3));
    }

    public void jButton8_actionPerformed(ActionEvent e) {
        double v1 = Double.parseDouble(num1.getText());
        int v3 = (int) v1;
        num3.setText(Integer.toBinaryString(v3));
```

```
    }
}

class 实验10_3_jButton8_actionAdapter
    implements ActionListener {
  private 实验10_3 adaptee;
  实验10_3_jButton8_actionAdapter(实验10_3 adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.jButton8_actionPerformed(e);
  }
}

class 实验10_3_hex_actionAdapter
    implements ActionListener {
  private 实验10_3 adaptee;
  实验10_3_hex_actionAdapter(实验10_3 adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.hex_actionPerformed(e);
  }
}

class 实验10_3_oct_actionAdapter
    implements ActionListener {
  private 实验10_3 adaptee;
  实验10_3_oct_actionAdapter(实验10_3 adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.oct_actionPerformed(e);
  }
}

class 实验10_3_dec_actionAdapter
    implements ActionListener {
  private 实验10_3 adaptee;
  实验10_3_dec_actionAdapter(实验10_3 adaptee) {
    this.adaptee = adaptee;
```

```
    }

    public void actionPerformed(ActionEvent e) {
        adaptee.dec_actionPerformed(e);
    }
}

class 实验10_3_sqrt_actionAdapter
        implements ActionListener {
    private 实验10_3 adaptee;
    实验10_3_sqrt_actionAdapter(实验10_3 adaptee) {
        this.adaptee = adaptee;
    }

    public void actionPerformed(ActionEvent e) {
        adaptee.sqrt_actionPerformed(e);
    }
}

class 实验10_3_abs_actionAdapter
        implements ActionListener {
    private 实验10_3 adaptee;
    实验10_3_abs_actionAdapter(实验10_3 adaptee) {
        this.adaptee = adaptee;
    }

    public void actionPerformed(ActionEvent e) {
        adaptee.abs_actionPerformed(e);
    }
}

class 实验10_3_cos_actionAdapter
        implements ActionListener {
    private 实验10_3 adaptee;
    实验10_3_cos_actionAdapter(实验10_3 adaptee) {
        this.adaptee = adaptee;
    }

    public void actionPerformed(ActionEvent e) {
        adaptee.cos_actionPerformed(e);
    }
}

class 实验10_3_sin_actionAdapter
```

```
    implements ActionListener {
private 实验10_3 adaptee;
实验10_3_sin_actionAdapter(实验10_3 adaptee) {
    this.adaptee = adaptee;
}

public void actionPerformed(ActionEvent e) {
    adaptee.sin_actionPerformed(e);
    }
}

class 实验10_3_div_actionAdapter
    implements ActionListener {
  private 实验10_3 adaptee;
  实验10_3_div_actionAdapter(实验10_3 adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.div_actionPerformed(e);
  }
}

class 实验10_3_mul_actionAdapter
    implements ActionListener {
  private 实验10_3 adaptee;
  实验10_3_mul_actionAdapter(实验10_3 adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.mul_actionPerformed(e);
  }
}

class 实验10_3_sub_actionAdapter
    implements ActionListener {
  private 实验10_3 adaptee;
  实验10_3_sub_actionAdapter(实验10_3 adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.sub_actionPerformed(e);
```

```
    }
}

class 实验10_3_add_actionAdapter
      implements ActionListener {
   private 实验10_3 adaptee;
   实验10_3_add_actionAdapter(实验10_3 adaptee) {
      this.adaptee = adaptee;
   }

   public void actionPerformed(ActionEvent e) {
      adaptee.add_actionPerformed(e);
   }
}
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=GBK">
<title>计算器小程序</title>
</head>
<body>
计算器小程序<br>
<applet
   codebase = "."
   code   = "实验.实验10_3.class"
   name   = "TestApplet"
   width  = "400"
   height = "300"
   hspace = "0"
   vspace = "0"
   align  = "middle"
>
</applet>
</body>
</html>
```

实验 11

1、答：

略

2、答：

略

3、答：

```
<%@ page contentType="text/html; charset=GBK" %>
<html>
<head>
<title>杨辉三角形</title>
</head>
<body bgcolor="#ffffff">
<h1>杨辉三角形</h1>
<%
  int n=1;
  try{
     n=Integer.parseInt(request.getParameter("N"));
  }catch (Exception e){
     n=1;
  }
     int yangHui[][] = new int[n][];
     for (int i = 0; i < yangHui.length; i++) { //计算杨辉三角形
        yangHui[i] = new int[i + 1];
        yangHui[i][0] = 1;
        for (int j = 1; j < yangHui[i].length - 1; j++) {
           yangHui[i][j] = yangHui[i - 1][j] + yangHui[i - 1][j - 1];
        }
        yangHui[i][yangHui[i].length - 1] = 1;
     }
     for (int i = 0; i < yangHui.length; i++) { //显示杨辉三角形
        for (int j = 0; j < yangHui[i].length; j++) {
           out.print(yangHui[i][j] + "\t");
        }
        out.println("<br>");
     }
%>
<form method="post" action="exercise11_3.jsp">
<br><br>
<input type="text" name="N" value="6">
<input type="submit" name="Submit" value="Submit">
<input type="reset" value="Reset">
</form>
</body>
</html>
```

4、答：

```
<%@ page contentType="text/html; charset=GBK" %>
```

```
<html>
<head>
<title>打开一个网站</title>
</head>
<body bgcolor="#ffffff">
<h1>打开一个网站</h1>
<%
    String url=request.getParameter("URL");
    if(url==null){
      url="";
    }
    url=url.trim();
    if(url!=""){
        out.println("<br>");
        out.println("<iframe src=\""+url+"\" name=\"clientwindow\" "+
            "width=\"600\" height=\"460\" scrolling=\"auto\""+
            " frameborder=\"0\" marginwidth=\"50\" marginheight=\"50\">"+
            "内容子窗口"+
            "</iframe>");
        out.println("<br>");
    }
%>
<form method="post" action="exercise11_4.jsp">
<br><br>
<input type="text" name="URL" value="">
<input type="submit" name="Submit" value="Submit">
<input type="reset" value="Reset">
</form>
</body>
</html>
```

实验 12

1、答：

```
import java.net.*;
import java.io.*;
import java.util.*;

public class 实验12_1 {
  public 实验12_10 {
    try {
```

```
        System.out.print("请输入网站地址: ");
        Scanner scanner = new Scanner(System.in);
        URL url1 = new URL(scanner.nextLine()); //建立 URL 类的对象
        BufferedReader dis = new BufferedReader(
              new InputStreamReader(url1.openStream())); //输入流的对象
        String line;
        int count = 0;
        while ( (line = dis.readLine()) != null) {
            System.out.println(line); //取得 URL 首页文件中的每一行
            count++;
        }
        dis.close();
        System.out.println("共" + count + "行");
    }
    catch (IOException exception) {
        System.err.println("" + exception);
    }
}

public static void main(String[] args) {
    new 实验12_1();
}
```

2、答：

```
package 实验;

import java.net.*;
import java.io.*;
import java.util.*;

public class 实验12_2服务器端 {
   public 实验12_2服务器端() {
      ServerSocket server;
      Socket socket;
      System.out.println("服务已经启动!\r\n");
      boolean done = false;
      try {
         server = new ServerSocket(4700);
         socket = server.accept();
         System.out.println("有客来!\r\n");
         BufferedReader in = new BufferedReader(new
                                                InputStreamReader
                                                (socket.getInputStream()));
```

```
            PrintStream out = new PrintStream(socket.getOutputStream());
            out.println("您好! 欢迎您的带来。");
            do {
               String s = in.readLine();
               if (s == null) {
                  s = "";
               }
               s = s.trim();
               System.out.println(s);
               if (s == null) {
                  continue;
               }
               long clientTime = Long.parseLong(s);
               Calendar now = new GregorianCalendar();
               long serverTime = now.getTimeInMillis();
               out.println("服务器与客户端时间相差 " + (serverTime - clientTime) + " 毫秒");
            }
            while (true);
         }
         catch (Exception e) {
         }
      }

      public static void main(String[] args) {
         new 实验12_2服务器端();
      }
   }

   import java.net.*;
   import java.io.*;
   import java.util.*;

   public class 实验12_2客户端 {
      Socket socket;
      BufferedReader in;
      PrintStream out;

      public 实验12_2客户端() {
         try {
            socket = new Socket("localhost", 4700);
            in = new BufferedReader(new InputStreamReader(socket.
                  getInputStream
                  ()));
            out = new PrintStream(socket.getOutputStream());
```

```
        while (true) {
            String s = in.readLine();
            System.out.println(s);
            Scanner scanner = new Scanner(System.in);
            s = scanner.nextLine();
            if (s != null) {
                Calendar now = new GregorianCalendar();
                out.println("" + now.getTimeInMillis());
            }
        }
        catch (IOException e2) {
        }
    }

    public static void main(String[] args) {
        new 实验12_2客户端();
    }
}
```

3、答：

```
package 实验;
import java.io.*;
import java.net.*;

class Person {
  Socket socket;
  BufferedReader in;
  PrintStream out;
  String name;
  int i;
  static int count;
  static int whoLeft;
}

public class 实验12_3服务器端 {
  Person person[] = new Person[2];
  public 实验12_3服务器端() {
    person[0] = new Person();
    person[0].i = 0;
    person[0].name = "Mis.Cafe";
    person[1] = new Person();
    person[1].name = "Mr.Java";
```

```
        person[1].i = 1;
        try {
            ServerSocket server = new ServerSocket(4700);
            System.out.println("聊天室开业了!");
            for (int i = 0; i < 2; i++, Person.count++) {

                person[i].socket = server.accept();
                System.out.println(person[i].name + " 连进来了.");
                person[i].in = new BufferedReader(new InputStreamReader
                                                  (person[i].socket.getInputStream()));
                person[i].out = new PrintStream(person[i].socket.getOutputStream());

                 person[i].out.println("您好! " + person[i].name + " 同志, 欢迎进入聊天室 。
\r");
                person[i].out.println("敲bye 或 exit 或 quit 则离开聊天室。\r");
                (new NotebookThread(i)).start();
                if (Person.count == 1) {
                  person[0].out.println("两个人都到齐了, 你俩可以聊天了!\r");
                  person[1].out.println("两个人都到齐了, 你俩可以聊天了!\r");
                }
                Person.whoLeft = 1 - 2 * i;
            }
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }

    public static void main(String[] args) {
        new 实验12_3服务器端();
    }

    class NotebookThread
            extends Thread {
        int i;
        public NotebookThread(int i) {
            this.i = i;
        }

        public void run() {
            try {
                while (true) {
                    String s = person[i].in.readLine();
```

```
            if (s == null) {
                s = " ";
            }
            else {
                s = s.trim();
                if (s.equalsIgnoreCase("bye") ||
                        s.equalsIgnoreCase("exit") ||
                        s.equalsIgnoreCase("quit") ||
                        s.equalsIgnoreCase("quit!")) {
                    Person.count--;
                    break;
                }
                if (s.equalsIgnoreCase("quit!")) {
                    System.exit(1);
                }
                if (Person.count == 2) {
                    person[1 - i].out.println(person[i].name + "的话:" + s + "\r");
                }
                else {
                    System.out.println(s);
                }
            }
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
package 实验;

import java.applet.*;
import java.awt.*;
import java.io.*;
import java.net.*;
import java.awt.event.*;

public class 实验12_3客户端
    extends Applet {
  public void init() {
    setLayout(new BorderLayout());
    ta = new TextArea(5, 40);
    ll = new Label("请输入聊天信息:");
```

```
        ta.setEditable(false);

        add(ta, BorderLayout.NORTH);
        add(ll);
        tf = new ClientTextField(40);
        tf.addKeyListener(new MyKeyAdapter());
        add(tf, BorderLayout.SOUTH);
    }

    public void start() {
        try {
            socket = new Socket("localhost", 4700);

            in = new BufferedReader(new InputStreamReader(socket.getInputStream
                ()));
            out = new PrintStream(socket.getOutputStream());

        }
        catch (IOException e) {
            System.out.println(e);
        }
        Receiver r = new Receiver(in, ta);
        r.start();
    }

    public void stop() {
        try {
            socket.close();
        }
        catch (IOException e) {
        }
    }

    public void send() {
        String s = tf.getText();
        out.println(s);
        tf.setText("");
    }

    TextArea ta;
    Label ll;
    ClientTextField tf;
    Socket socket;
    BufferedReader in;
```

```
PrintStream out;

class MyKeyAdapter
    extends KeyAdapter {
  public void keyPressed(KeyEvent kE) {
    if (kE.getKeyCode() == kE.VK_ENTER) {
      send();
    }
  }
}

class ClientTextField
    extends TextField {
  public ClientTextField(int i) {
    super(i);
  }
}

class Receiver
    extends Thread {
  public Receiver(BufferedReader d, TextArea t) {
    in = d;
    ta = t;

  }

  public void run() {
    while (true) {
      try {
        String s = in.readLine();
        if (s != null)
          ta.append(s + "\n");
        else
          this.destroy();
      }
      catch (IOException e) {
      }
    }
  }

  private BufferedReader in;
  TextArea ta;
}
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=GBK">
<title>两个客户端之间的通讯小程序</title>
</head>
<body>
两个客户端之间的通讯小程序<br>
<applet
    codebase = "."
    code     = "实验.实验12_3.class"
    name     = "实验12_3"
    width    = "400"
    height   = "300"
    hspace   = "0"
    vspace   = "0"
    align    = "middle"
>
</applet>
</body>
</html>
```