【程序1】     
题目：古典问题：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？     
//这是一个菲波拉契数列问题  
public class lianxi01 {  
public static void main(String[] args) {  
System.out.println("第1个月的兔子对数:    1");  
System.out.println("第2个月的兔子对数:    1");  
int f1 = 1, f2 = 1, f, M=24;  
     for(int i=3; i<=M; i++) {  
      f = f2;  
      f2 = f1 + f2;  
      f1 = f;  
      System.out.println("第" + i +"个月的兔子对数: "+f2);  
         }  
}  
}

【程序2】     
题目：判断101-200之间有多少个素数，并输出所有素数。   
程序分析：判断素数的方法：用一个数分别去除2到sqrt(这个数)，如果能被整除， 则表明此数不是素数，反之是素数。     
public class lianxi02 {  
public static void main(String[] args) {  
    int count = 0;  
    for(int i=101; i<200; i+=2) {  
     boolean b = false;  
     for(int j=2; j<=Math.sqrt(i); j++)   
     {  
        if(i % j == 0) { b = false; break; }   
         else           { b = true; }  
     }  
        if(b == true) {count ++;System.out.println(i );}  
                                  }  
    System.out.println( "素数个数是: " + count);  
}  
}

【程序3】     
题目：打印出所有的 "水仙花数 "，所谓 "水仙花数 "是指一个三位数，其各位数字立方和等于该数本身。例如：153是一个 "水仙花数 "，因为153=1的三次方＋5的三次方＋3的三次方。  
public class lianxi03 {  
public static void main(String[] args) {  
     int b1, b2, b3;   
     for(int m=101; m<1000; m++) {   
      b3 = m / 100;  
      b2 = m % 100 / 10;  
      b1 = m %    10;  
      if((b3\*b3\*b3 + b2\*b2\*b2 + b1\*b1\*b1) == m) {  
      System.out.println(m+"是一个水仙花数"); }  
     }  
}  
}

【程序4】     
题目：将一个正整数分解质因数。例如：输入90,打印出90=2\*3\*3\*5。     
程序分析：对n进行分解质因数，应先找到一个最小的质数k，然后按下述步骤完成：     
(1)如果这个质数恰等于n，则说明分解质因数的过程已经结束，打印出即可。     
(2)如果n <> k，但n能被k整除，则应打印出k的值，并用n除以k的商,作为新的正整数你n,重复执行第一步。     
(3)如果n不能被k整除，则用k+1作为k的值,重复执行第一步。    
import java.util.\*;  
public     class     lianxi04{   
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);  
        System.out.print( "请键入一个正整数:     ");   
        int    n    = s.nextInt();  
        int k=2;   
        System.out.print(n + "=" );  
        while(k <= n) {  
          if(k == n) {System.out.println(n);break;}  
            else if( n % k == 0) {System.out.print(k + "\*");n = n / k; }   
                    else    k++;  
                   }  
     }  
    }   
【程序5】     
题目：利用条件运算符的嵌套来完成此题：学习成绩> =90分的同学用A表示，60-89分之间的用B表示，60分以下的用C表示。     
import java.util.\*;  
public class lianxi05 {  
public static void main(String[] args) {  
     int x;  
     char grade;  
     Scanner s = new Scanner(System.in);  
     System.out.print( "请输入一个成绩: ");   
     x = s.nextInt();    
     grade = x >= 90 ? 'A'  
           : x >= 60 ? 'B'  
           :'C';  
    System.out.println("等级为："+grade);  
    
}  
}   
【程序6】     
题目：输入两个正整数m和n，求其最大公约数和最小公倍数。     
/\*\*在循环中，只要除数不等于0，用较大数除以较小的数，将小的一个数作为下一轮循环的大数，取得的余数作为下一轮循环的较小的数，如此循环直到较小的数的值为0，返回较大的数，此数即为最大公约数，最小公倍数为两数之积除以最大公约数。\* /  
import java.util.\*;  
public    class     lianxi06     {   
public static void main(String[] args) {  
int     a ,b,m;  
Scanner s = new Scanner(System.in);  
System.out.print( "键入一个整数： ");   
a = s.nextInt();  
System.out.print( "再键入一个整数： ");   
b = s.nextInt();  
      deff cd = new deff();  
      m = cd.deff(a,b);  
      int n = a \* b / m;  
      System.out.println("最大公约数: " + m);  
      System.out.println("最小公倍数: " + n);  
}   
}  
class deff{  
public int deff(int x, int y) {  
     int t;  
     if(x < y) {  
      t = x;  
      x = y;  
      y = t;  
     }    
     while(y != 0) {  
      if(x == y) return x;  
      else {  
       int k = x % y;  
       x = y;  
       y = k;  
      }  
     }  
     return x;  
}  
}   
【程序7】     
题目：输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。

**package** WuYang;

**import** java.util.\*;

**public** **class** lianxi07 {

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

**int** abcCount=0;//英文字母个数

**int** spaceCount=0;//空格键个数

**int** numCount=0;//数字个数

**int** otherCount=0;//其他字符个数

Scanner scan = **new** Scanner(System.*in*);//扫描器接受控制台的输入信息

System.*out*.println("输入一组字符");

String str=scan.nextLine();//取出控制台的一行信息,也就是你输入的信息

**char**[] ch = str.toCharArray();//把取道的字符串变成一个char数组

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

**if**(Character.*isLetter*(ch[i])){

//判断是否字母

abcCount++;

}

**else** **if**(Character.*isDigit*(ch[i])){

//判断是否数字

numCount++;

}

**else** **if**(Character.*isSpaceChar*(ch[i])){

//判断是否空格键

spaceCount++;

}

**else**{

//以上都不是则认为是其他字符

otherCount++;

}

}

System.*out*.println("字母个数："+abcCount);

System.*out*.println("数字个数："+numCount);

System.*out*.println("空格个数："+spaceCount);

System.*out*.println("其他字符个数："+otherCount);

}

}

【程序8】     
题目：求s=a+aa+aaa+aaaa+aa...a的值，其中a是一个数字。例如2+22+222+2222+22222(此时共有5个数相加)，几个数相加有键盘控制。     
import java.util.\*;  
public class lianxi08 {  
public static void main(String[] args) {  
     long a , b = 0, sum = 0;  
     Scanner s = new Scanner(System.in);  
     System.out.print("输入数字a的值： ");  
     a = s.nextInt();  
     System.out.print("输入相加的项数：");  
     int n = s.nextInt();  
     int i = 0;  
     while(i < n) {  
      b = b + a;  
      sum = sum + b;  
      a = a \* 10;  
      ++ i;  
     }  
      System.out.println(sum);  
}  
}   
【程序9】     
题目：一个数如果恰好等于它的因子之和，这个数就称为 "完数 "。例如6=1＋2＋3.编程     找出1000以内的所有完数。     
public class lianxi09 {  
public static void main(String[] args) {  
     System.out.println("1到1000的完数有： ");  
     for(int i=1; i<1000; i++) {  
      int t = 0;  
      for(int j=1; j<= i/2; j++) {  
       if(i % j == 0) {  
        t = t + j;  
       }  
      }  
      if(t == i) {  
       System.out.print(i + "     ");  
      }  
     }  
}

}

【程序10】     
题目：一球从100米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在 第10次落地时，共经过多少米？第10次反弹多高？   
**public** **class** lianxi10 {

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

**double** h = 100,s = 0;

**for**(**int** i=1; i<=10; i++) {

s = s + 2\*h;

h = h / 2;

}

s=s-100;

System.*out*.println("经过路程：" + s);

System.*out*.println("最后高度：" + h);

}

}   
【程序11】     
题目：有1、2、3、4四个数字，能组成多少个互不相同且一个数字中无重复数字的三位数？并把他们都输入。     
public class lianxi11 {  
public static void main(String[] args) {  
     int count = 0;  
     for(int x=1; x<5; x++) {  
      for(int y=1; y<5; y++) {  
       for(int z=1; z<5; z++) {  
        if(x != y && y != z && x != z) {  
         count ++;  
         System.out.println(x\*100 + y\*10 + z );  
        }  
       }  
      }  
     }  
     System.out.println("共有" + count + "个三位数");  
}  
}   
【程序12】     
题目：企业发放的奖金根据利润提成。利润(I)低于或等于10万元时，奖金可提10%；利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可可提成7.5%；20万到40万之间时，高于20万元的部分，可提成5%；40万到60万之间时高于40万元的部分，可提成3%；60万到100万之间时，高于60万元的部分，可提成1.5%，高于100万元时，超过100万元的部分按1%提成，从键盘输入当月利润，求应发放奖金总数？     
import java.util.\*;  
public class lianxi12 {  
public static void main(String[] args) {  
     double x = 0,y = 0;  
     System.out.print("输入当月利润（万）：");  
     Scanner s = new Scanner(System.in);  
     x = s.nextInt();  
     if(x > 0 && x <= 10) {  
     y = x \* 0.1;  
     } else if(x > 10 && x <= 20) {  
      y = 10 \* 0.1 + (x - 10) \* 0.075;  
     } else if(x > 20 && x <= 40) {  
      y = 10 \* 0.1 + 10 \* 0.075 + (x - 20) \* 0.05;  
     } else if(x > 40 && x <= 60) {  
      y = 10 \* 0.1 + 10 \* 0.075 + 20 \* 0.05 + (x - 40) \* 0.03;  
     } else if(x > 60 && x <= 100) {  
      y = 20 \* 0.175 + 20 \* 0.05 + 20 \* 0.03 + (x - 60) \* 0.015;   
     } else if(x > 100) {  
      y = 20 \* 0.175 + 40 \* 0.08 + 40 \* 0.015 + (x - 100) \* 0.01;  
     }  
     System.out.println("应该提取的奖金是 " + y + "万");  
}  
}

【程序13】     
题目：一个整数，它加上100后是一个完全平方数，再加上168又是一个完全平方数，请问该数是多少？     
public class lianxi13 {  
public static void main(String[] args) {  
     for(int x =1; x<100000; x++) {  
      if(Math.sqrt(x+100) % 1 == 0) {  
       if(Math.sqrt(x+168) % 1 == 0) {  
        System.out.println(x + "加100是一个完全平方数，再加168又是一个完全平方数");  
       }  
      }  
     }  
}  
}  
  
/\*按题意循环应该从-100开始（整数包括正整数、负整数、零），这样会多一个满足条件的数-99。  
但是我看到大部分人解这道题目时都把题中的“整数”理解成正整数，我也就随大流了。\*/  
【程序14】    
题目：输入某年某月某日，判断这一天是这一年的第几天？     
import java.util.\*;  
public class lianxi14 {  
public static void main(String[] args) {  
     int year, month, day;  
     int days = 0;  
     int d = 0;  
     int e;  
     input fymd = new input();  
     do {  
     e = 0;  
     System.out.print("输入年：");  
     year =fymd.input();  
     System.out.print("输入月：");  
     month = fymd.input();  
     System.out.print("输入天：");  
     day = fymd.input();  
     if (year < 0 || month < 0 || month > 12 || day < 0 || day > 31) {  
     System.out.println("输入错误，请重新输入！");  
     e=1 ;   
     }  
     }while( e==1);

      for (int i=1; i <month; i++) {  
      switch (i) {  
      case 1:  
      case 3:  
      case 5:  
      case 7:  
      case 8:  
      case 10:  
      case 12:  
       days = 31;  
      break;  
      case 4:  
      case 6:  
      case 9:  
      case 11:  
       days = 30;  
      break;  
      case 2:  
       if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0)) {  
        days = 29;  
       } else {  
        days = 28;  
       }  
       break;  
      }  
      d += days;  
      }  
     System.out.println(year + "-" + month + "-" + day + "是这年的第" + (d+day) + "天。");  
}  
}  
class input{  
public int input() {  
     int value = 0;  
     Scanner s = new Scanner(System.in);  
     value = s.nextInt();  
     return value;  
}  
}

【程序15】     
题目：输入三个整数x,y,z，请把这三个数由小到大输出。     
import java.util.\*;  
public class lianxi15 {  
public static void main(String[] args) {  
     input fnc = new input();  
     int x=0, y=0, z=0;  
     System.out.print("输入第一个数字：");  
      x = fnc.input();  
     System.out.print("输入第二个数字：");  
      y = fnc.input();  
     System.out.print("输入第三个数字：");  
      z = fnc.input();     
    if(x > y) {  
      int t = x;  
      x = y;  
      y = t;  
     }  
    if(x > z) {  
      int t = x;  
      x = z;  
      z = t;  
     }  
    if(y > z) {  
      int t = y;  
      y = z;  
      z = t;  
     }  
    System.out.println( "三个数字由小到大排列为： "+x + " " + y + " " + z);  
}  
}  
class input{  
public int input() {  
     int value = 0;  
     Scanner s = new Scanner(System.in);  
     value = s.nextInt();  
     return value;  
}  
}   
【程序16】  
题目：输出9\*9口诀。       
public class lianxi16 {  
public static void main(String[] args) {  
     for(int i=1; i<10; i++) {  
      for(int j=1; j<=i; j++) {  
       System.out.print(j + "\*" + i + "=" + j\*i + "    " );  
         if(j\*i<10){System.out.print(" ");}  
}  
          System.out.println();  
     }  
}  
}   
【程序17】     
题目：猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个     第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下     的一半零一个。到第10天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。     
public class lianxi17 {  
public static void main(String[] args) {  
     int x = 1;  
     for(int i=2; i<=10; i++) {  
      x = (x+1)\*2;  
     }  
     System.out.println("猴子第一天摘了 " + x + " 个桃子");  
}  
}

【程序18】     
题目：两个乒乓球队进行比赛，各出三人。甲队为a,b,c三人，乙队为x,y,z三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a说他不和x比，c说他不和x,z比，请编程序找出三队赛手的名单。     
public class lianxi18 {

static char[] m = { 'a', 'b', 'c' };  
static char[] n = { 'x', 'y', 'z' };

public static void main(String[] args) {

   for (int i = 0; i < m.length; i++) {  
    for (int j = 0; j < n.length; j++) {

     if (m[i] == 'a' && n[j] == 'x') {

      continue;

} else if (m[i] == 'a' && n[j] == 'y') {

      continue;

     } else if ((m[i] == 'c' && n[j] == 'x')

       || (m[i] == 'c' && n[j] == 'z')) {

      continue;

     } else if ((m[i] == 'b' && n[j] == 'z')

       || (m[i] == 'b' && n[j] == 'y')) {

      continue;

     } else

      System.out.println(m[i] + " vs " + n[j]);

    }

   }

}

}

【程序19】     
题目：打印出如下图案（菱形）     
     \*     
   \*\*\*     
 \*\*\*\*\*     
\*\*\*\*\*\*\*     
 \*\*\*\*\*     
   \*\*\*     
    \*     
public class lianxi19 {  
public static void main(String[] args) {  
    int H = 7, W = 7;//高和宽必须是相等的奇数  
    for(int i=0; i<(H+1) / 2; i++) {  
     for(int j=0; j<W/2-i; j++) {  
      System.out.print(" ");

     }  
     for(int k=1; k<(i+1)\*2; k++) {  
      System.out.print('\*');  
     }  
     System.out.println();  
    }  
    for(int i=1; i<=H/2; i++) {  
     for(int j=1; j<=i; j++) {  
      System.out.print(" ");  
     }  
     for(int k=1; k<=W-2\*i; k++) {  
      System.out.print('\*');  
     }  
     System.out.println();  
    }  
}  
}

【程序20】     
题目：有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13...求出这个数列的前20项之和。

public class lianxi20 {  
public static void main(String[] args) {  
    int x = 2, y = 1, t;  
    double sum = 0;  
    for(int i=1; i<=20; i++) {  
     sum = sum + (double)x / y;  
     t = y;  
     y = x;  
     x = y + t;  
     }  
System.out.println("前20项相加之和是： " + sum);  
}  
}

【程序21】     
题目：求1+2!+3!+...+20!的和     
public class lianxi21 {  
public static void main(String[] args) {  
    long sum = 0;   
    long fac = 1;  
    for(int i=1; i<=20; i++) {  
     fac = fac \* i;  
     sum += fac;  
    }  
    System.out.println(sum);  
}

}

【程序22】     
题目：利用递归方法求5!。     
public class lianxi22 {  
public static void main(String[] args) {  
       int n = 5;  
    rec fr = new rec();  
    System.out.println(n+"! = "+fr.rec(n));  
}  
}  
class rec{  
public long rec(int n) {  
    long value = 0 ;  
    if(n ==1 ) {  
     value = 1;  
    } else   {  
     value = n \* rec(n-1);  
    }  
    return value;  
}  
}

【程序23】     
题目：有5个人坐在一起，问第五个人多少岁？他说比第4个人大2岁。问第4个人岁数，他说比第3个人大2岁。问第三个人，又说比第2人大两岁。问第2个人，说比第一个人大两岁。最后问第一个人，他说是10岁。请问第五个人多大？

public class lianxi23 {  
public static void main(String[] args) {  
    int age = 10;  
     for(int i=2; i<=5; i++) {  
     age =age+2;  
    }  
    System.out.println(age);  
}  
}

【程序24】     
题目：给一个不多于5位的正整数，要求：一、求它是几位数，二、逆序打印出各位数字。     
//使用了长整型最多输入18位  
import java.util.\*;  
public class lianxi24 {  
public static void main(String[] args) {  
   Scanner s = new Scanner(System.in);  
   System.out.print("请输入一个正整数：");  
   long a = s.nextLong();  
   String ss = Long.toString(a);  
    char[] ch = ss.toCharArray();  
    int j=ch.length;  
    System.out.println(a + "是一个"+ j +"位数。");  
    System.out.print("按逆序输出是：");  
    for(int i=j-1; i>=0; i--) {  
    System.out.print(ch[i]);  
   }  
   }  
   }  
【程序25】     
题目：一个5位数，判断它是不是回文数。即12321是回文数，个位与万位相同，十位与千位相同。     
import java.util.\*;  
public class lianxi25 {  
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    int a;  
    do{  
     System.out.print("请输入一个5位正整数：");  
      a = s.nextInt();  
      }while(a<10000||a>99999);  
     String ss =String.valueOf(a);  
     char[] ch = ss.toCharArray();  
     if(ch[0]==ch[4]&&ch[1]==ch[3]){  
     System.out.println("这是一个回文数");}  
     else {System.out.println("这不是一个回文数");}  
    }  
    }  
//这个更好，不限位数  
import java.util.\*;  
public class lianxi25a {  
public static void main(String[] args) {  
   Scanner s = new Scanner(System.in);  
   boolean is =true;  
   System.out.print("请输入一个正整数：");  
   long a = s.nextLong();  
   String ss = Long.toString(a);  
   char[] ch = ss.toCharArray();  
   int j=ch.length;  
   for(int i=0; i<j/2; i++) {  
   if(ch[i]!=ch[j-i-1]){is=false;}  
   }  
   if(is==true){System.out.println("这是一个回文数");}  
     else {System.out.println("这不是一个回文数");}  
    }  
**}**【程序26】     
题目：请输入星期几的第一个字母来判断一下是星期几，如果第一个字母一样，则继续   判断第二个字母。     
import java.util.\*;  
public class lianxi26 {  
public static void main(String[] args) {  
    getChar tw = new getChar();  
    System.out.println("请输入星期的第一个大写字母：");  
    char ch = tw.getChar();  
    switch(ch) {  
     case 'M':   
      System.out.println("Monday");  
      break;  
     case 'W':   
      System.out.println("Wednesday");  
      break;  
     case 'F':  
      System.out.println("Friday");  
      break;  
     case 'T': {  
      System.out.println("请输入星期的第二个字母：");  
      char ch2 = tw.getChar();  
      if(ch2 == 'U') {System.out.println("Tuesday"); }  
      else if(ch2 == 'H') {System.out.println("Thursday"); }  
      else {System.out.println("无此写法！");  
       }  
     };   
      break;  
     case 'S': {  
       System.out.println("请输入星期的第二个字母：");  
      char ch2 = tw.getChar();  
      if(ch2 == 'U') {System.out.println("Sunday"); }  
       else if(ch2 == 'A') {System.out.println("Saturday"); }  
       else {System.out.println("无此写法！");  
       }  
     };  
      break;  
default:System.out.println("无此写法！");  
}  
   }  
}  
  
class getChar{  
public char getChar() {  
    Scanner s = new Scanner(System.in);  
    String str = s.nextLine();  
    char ch = str.charAt(0);  
    if(ch<'A' || ch>'Z') {  
     System.out.println("输入错误，请重新输入");  
     ch=getChar();  
    }  
    return ch;  
}  
}    
【程序27】     
题目：求100之内的素数     
//使用除sqrt(n)的方法求出的素数不包括2和3  
public class lianxi27 {  
public static void main(String[] args) {  
    boolean b =false;  
    System.out.print(2 + " ");  
    System.out.print(3 + " ");  
    for(int i=3; i<100; i+=2) {  
     for(int j=2; j<=Math.sqrt(i); j++) {  
      if(i % j == 0) {b = false;  
                      break;  
       } else{b = true;}  
     }  
   if(b == true) {System.out.print(i + " ");}  
    }  
   }  
}  
//该程序使用除1位素数得2位方法，运行效率高通用性差。  
public class lianxi27a {  
public static void main(String[] args) {  
    int[] a = new int[]{2, 3, 5, 7};  
   for(int j=0; j<4; j++)System.out.print(a[j] + " ");  
    boolean b =false;  
    for(int i=11; i<100; i+=2) {  
     for(int j=0; j<4; j++) {  
      if(i % a[j] == 0) {b = false;  
                      break;  
       } else{b = true;}  
     }  
   if(b == true) {System.out.print(i + " ");}  
    }  
   }  
}  
【程序28】     
题目：对10个数进行排序     
import java.util.\*;  
public class lianxi28 {  
public static void main(String[] args) {  
Scanner s = new Scanner(System.in);  
   int[] a = new int[10];  
   System.out.println("请输入10个整数：");  
   for(int i=0; i<10; i++) {  
    a[i] = s.nextInt();  
   }

   for(int i=0; i<10; i++) {  
    for(int j=i+1; j<10; j++) {  
     if(a[i] > a[j]) {  
      int t = a[i];  
      a[i] = a[j];  
      a[j] = t;  
     }  
    }  
   }  
   for(int i=0; i<10; i++) {  
    System.out.print(a[i] + " ");  
   }  
}  
}  
【程序29】     
题目：求一个3\*3矩阵对角线元素之和

import java.util.\*;  
public class lianxi29 {  
public static void main(String[] args) {  
   Scanner s = new Scanner(System.in);  
   int[][] a = new int[3][3];  
System.out.println("请输入9个整数：");  
   for(int i=0; i<3; i++) {  
    for(int j=0; j<3; j++) {  
     a[i][j] = s.nextInt();  
    }  
   }  
   System.out.println("输入的3 \* 3 矩阵是:");  
   for(int i=0; i<3; i++) {  
    for(int j=0; j<3; j++) {  
     System.out.print(a[i][j] + " ");  
    }  
    System.out.println();  
   }  
   int sum = 0;  
   for(int i=0; i<3; i++) {  
    for(int j=0; j<3; j++) {  
     if(i == j) {  
      sum += a[i][j];  
     }  
    }  
   }  
   System.out.println("对角线之和是：" + sum);  
}  
}

【程序30】     
题目：有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中。      
//此程序不好，没有使用折半查找插入

**import** java.util.\*;

**public** **class** lianxi30 {

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

**int**[] a = **new** **int**[]{1, 2, 6, 14, 25, 36, 37,55};

**int**[] b = **new** **int**[a.length+1];

**int** t1=0, t2 = 0;

**int** i =0;

Scanner s= **new** Scanner(System.*in*);

System.*out*.print("请输入一个整数：");

**int** num = s.nextInt();

**if**(num >= a[a.length-1]) {

b[b.length-1] = num;

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

b[i] = a[i];

}

} **else** {

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

**if**(num >= a[i]) {

b[i] = a[i];

} **else** {

b[i] = num;

**break**;

}

}

**for**(**int** j=i+1; j<b.length; j++) {

b[j] = a[j-1];

}

}

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

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

}

}

}

【程序31】  
题目：将一个数组逆序输出。     
import java.util.\*;  
public class lianxi31 {  
public static void main(String[] args) {  
   Scanner s = new Scanner(System.in);  
   int a[] = new int[20];  
System.out.println("请输入多个正整数（输入-1表示结束）：");  
   int i=0,j;   do{  
      a[i]=s.nextInt();  
      i++;  
   }while (a[i-1]!=-1);

   System.out.println("你输入的数组为：");  
   for( j=0; j<i-1; j++) {  
    System.out.print(a[j]+"   ");  
}  
   System.out.println("\n数组逆序输出为：");  
   for( j=i-2; j>=0; j=j-1) {  
    System.out.print(a[j]+"   ");  
}  
    }  
   }  
【程序32】     
题目：取一个整数a从右端开始的4～7位。     
import java.util.\*;  
public class lianxi32 {  
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    System.out.print("请输入一个7位以上的正整数：");  
    long a = s.nextLong();  
    String ss = Long.toString(a);  
    char[] ch = ss.toCharArray();  
    int j=ch.length;  
    if (j<7){System.out.println("输入错误！");}  
    else {  
     System.out.println("截取从右端开始的4～7位是："+ch[j-7]+ch[j-6]+ch[j-5]+ch[j-4]);  
     }  
    }  
    }   
【程序33】    
题目：打印出杨辉三角形（要求打印出10行如下图）        
            1     
          1    1     
        1    2    1     
      1    3    3    1     
    1    4    6    4    1     
1    5    10    10    5    1     
…………  
public class lianxi33 {  
public static void main(String[] args) {  
    int[][] a = new int[10][10];  
   for(int i=0; i<10; i++) {  
    a[i][i] = 1;  
    a[i][0] = 1;  
   }  
   for(int i=2; i<10; i++) {  
    for(int j=1; j<i; j++) {  
     a[i][j] = a[i-1][j-1] + a[i-1][j];  
    }  
   }  
     for(int i=0; i<10; i++) {  
    for(int k=0; k<2\*(10-i)-1; k++) {  
     System.out.print(" ");  
    }  
    for(int j=0; j<=i; j++) {  
     System.out.print(a[i][j] + "   ");  
    }  
    System.out.println();  
   }  
}  
}  
【程序34】     
题目：输入3个数a,b,c，按大小顺序输出。     
import java.util.Scanner;  
public class lianxi34 {  
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    System.out.println("请输入3个整数：");  
    int a = s.nextInt();  
    int b = s.nextInt();  
    int c = s.nextInt();  
      if(a < b) {  
     int t = a;  
     a = b;  
     b = t;  
    }  
      if(a < c) {  
     int t = a;  
     a = c;  
     c = t;  
    }  
     if(b < c) {  
     int t = b;  
     b = c;  
     c = t;  
    }  
    System.out.println("从大到小的顺序输出:");  
    System.out.println(a + " " + b + " " + c);  
}  
}   
【程序35】     
题目：输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组。

**import** java.util.\*;

**public** **class** lianxi35 {

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

**int** N = 8;

**int**[] a = **new** **int**[N];// 创建一个八个元素的数组

Scanner s = **new** Scanner(System.*in*);

**int** idx1 = 0, idx2 = 0;

System.*out*.println("请输入8个整数：");

**for** (**int** i = 0; i < N; i++) {

a[i] = s.nextInt();

}

// 获得输入的八个数字

System.*out*.println("你输入的数组为：");

**for** (**int** i = 0; i < N; i++) {

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

}

// 输出输入的数组

**int** max = a[0], min = a[0];

**for** (**int** i = 0; i < N; i++) {

**if** (a[i] > max) {

max = a[i];

idx1 = i;

}

// 找出最大的数和其下标

**if** (a[i] < min) {

min = a[i];

idx2 = i;

}// 找出最大的数和其下标

}

**if** (idx1 != 0) {

**int** temp = a[0];

a[0] = a[idx1];

a[idx1] = temp;

}// 最大的数和第一个数交换位置

**if** (idx2 != N - 1) {

**int** temp = a[N - 1];

a[N - 1] = a[idx2];

a[idx2] = temp;

}// 最小的数和最后一个数交换位置

System.*out*.println("\n交换后的数组为：");

**for** (**int** i = 0; i < N; i++) {

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

}

}

}

【程序36】     
题目：有n个整数，使其前面各数顺序向后移m个位置，最后m个数变成最前面的m个数     
import java.util.Scanner;  
public class lianxi36 {  
public static void main(String[] args) {  
   int N =10;  
   int[] a = new int[N];  
   Scanner s = new Scanner(System.in);  
   System.out.println("请输入10个整数：");  
   for(int i=0; i<N; i++) {  
    a[i] = s.nextInt();  
   }  
   System.out.print("你输入的数组为：");  
   for(int i=0; i<N; i++) {  
     System.out.print(a[i] + " ");  
   }  
   System.out.print("\n请输入向后移动的位数：");  
   int m = s.nextInt();  
   int[] b = new int[m];  
   for(int i=0; i<m; i++) {  
    b[i] = a[N-m+i];  
   }  
   for(int i=N-1; i>=m; i--) {  
   a[i] = a[i-m];  
   }  
   for(int i=0; i<m; i++) {  
    a[i] = b[i];  
   }  
System.out.print("位移后的数组是：");  
   for(int i=0; i<N; i++) {  
    System.out.print(a[i] + " ");  
   }  
}  
}  
【程序37】     
题目：有n个人围成一圈，顺序排号。从第一个人开始报数（从1到3报数），凡报到3的人退出圈子，问最后留下的是原来第几号的那位。     
import java.util.Scanner;  
public class lianxi37 {  
public static void main(String[] args) {  
   Scanner s = new Scanner(System.in);  
   System.out.print("请输入排成一圈的人数：");  
   int n = s.nextInt();  
   boolean[] arr = new boolean[n];  
   for(int i=0; i<arr.length; i++) {  
    arr[i] = true;  
   }//数组赋值都是true  
   int leftCount = n;  
   int countNum = 0;  
   int index = 0;  
   while(leftCount > 1) {  
    if(arr[index] == true) {  
     countNum ++;   
     if(countNum == 3) {  
      countNum =0;  
      arr[index] = false;  
      leftCount --;  
     }  
    }  
     index ++;  
     if(index == n) {  
     index = 0;  
    }  
   }  
    for(int i=0; i<n; i++) {  
    if(arr[i] == true) {  
     System.out.println("原排在第"+(i+1)+"位的人留下了。");  
    }  
   }  
}  
}  
【程序38】     
题目：写一个函数，求一个字符串的长度，在main函数中输入字符串，并输出其长度。     
/\*………………  
\*……题目意思似乎不能用length()函数     \*/  
import java.util.\*;  
public class lianxi38 {  
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    System.out.println("请输入一个字符串：");  
    String str = s.nextLine();  
     System.out.println("字符串的长度是："+str.length());  
    }  
    }   
【程序39】     
题目：编写一个函数，输入n为偶数时，调用函数求1/2+1/4+...+1/n,当输入n为奇数时，调用函数1/1+1/3+...+1/n(利用指针函数)     
//没有利用指针函数  
import java.util.\*;  
public class lianxi39 {  
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    System.out.print("请输入一个正整数 n= ");  
    int n = s.nextInt();  
    System.out.println("相应数列的和为：" + sum(n));  
   }  
public static double sum(int n) {  
    double res = 0;  
    if(n % 2 == 0) {  
     for(int i=2; i<=n; i+=2) {  
      res += (double)1 / i;  
     }   
    } else {  
     for(int i=1; i<=n; i+=2) {  
      res += (double)1 / i ;  
     }  
    }  
    return res;  
}  
}   
【程序40】     
题目：字符串排序。     
public class lianxi40 {  
public static void main(String[] args) {  
   int N=5;  
   String temp = null;  
   String[] s = new String[N];  
   s[0] = "matter";  
   s[1] = "state";  
   s[2] = "solid";  
   s[3] = "liquid";  
   s[4] = "gas";  
   for(int i=0; i<N; i++) {  
    for(int j=i+1; j<N; j++) {  
     if(compare(s[i], s[j]) == false) {  
      temp = s[i];  
      s[i] = s[j];  
      s[j] = temp;  
     }  
    }  
   }  
    for(int i=0; i<N; i++) {  
    System.out.println(s[i]);  
   }  
}  
static boolean compare(String s1, String s2) {  
   boolean result = true;  
   for(int i=0; i<s1.length() && i<s2.length(); i++) {  
    if(s1.charAt(i) > s2.charAt(i)) {  
     result = false;  
     break;  
    } else if(s1.charAt(i) <s2.charAt(i)) {  
     result = true;  
     break;  
    } else {  
     if(s1.length() < s2.length()) {  
      result = true;  
     } else {  
      result = false;  
     }  
    }  
   }  
   return result;  
}  
}  
**【程序41】     
题目：海滩上有一堆桃子，五只猴子来分。第一只猴子把这堆桃子凭据分为五份，多了一个，这只猴子把多的一个扔入海中，拿走了**一份。第二只猴子把剩下的桃子又平均分成五份，又多了一个，它同样把多的一个扔入海中，拿走了一份，第三、第四、第五只猴子都是这样做的，问海滩上原来最少有多少个桃子？     
public class lianxi41 {   
public static void main (String[] args) {   
int i,m,j=0,k,count;   
for(i=4;i<10000;i+=4)   
   { count=0;   
     m=i;   
     for(k=0;k<5;k++)   
        {   
         j=i/4\*5+1;   
         i=j;   
         if(j%4==0)   
            count++;   
            else break;   
       }   
    i=m;   
if(count==4)   
{System.out.println("原有桃子 "+j+" 个");   
break;}   
}   
}   
}   
【程序42】     
题目：809\*??=800\*??+9\*??+1    其中??代表的两位数,8\*??的结果为两位数，9\*??的结果为3位数。求??代表的两位数，及809\*??后的结果。     
//题目错了！809x=800x+9x+1 这样的方程无解。去掉那个1就有解了。  
public class lianxi42 {   
public static void main (String[] args) {   
int a=809,b,i;  
for(i=10;i<13;i++)  
{b=i\*a ;  
if(8\*i<100&&9\*i>=100)  
System.out.println ("809\*"+i+"="+"800\*"+i+"+"+"9\*"+i+"="+b);}  
}  
}   
【程序43】     
题目：求0—7所能组成的奇数个数。     
//组成1位数是4个。  
//组成2位数是7\*4个。  
//组成3位数是7\*8\*4个。  
//组成4位数是7\*8\*8\*4个。  
//......  
public class lianxi43 {   
public static void main (String[] args) {   
int sum=4;  
int j;  
System.out.println("组成1位数是 "+sum+" 个");  
sum=sum\*7;  
System.out.println("组成2位数是 "+sum+" 个");  
for(j=3;j<=9;j++){  
sum=sum\*8;   
System.out.println("组成"+j+"位数是 "+sum+" 个");  
}  
}  
}   
【程序44】     
题目：一个偶数总能表示为两个素数之和。     
//由于用除sqrt(n)的方法求出的素数不包括2和3，  
//因此在判断是否是素数程序中人为添加了一个3。  
import java.util.\*;  
public class lianxi44 {  
public static void main(String[] args) {  
Scanner s = new Scanner(System.in);  
int n,i;  
do{  
     System.out.print("请输入一个大于等于6的偶数：");  
     n = s.nextInt();  
    } while(n<6||n%2!=0);   //判断输入是否是>=6偶数,不是,重新输入  
fun fc = new fun();  
    for(i=2;i<=n/2;i++){  
    if((fc.fun(i))==1&&(fc.fun(n-i)==1))   
    {int j=n-i;  
     System.out.println(n+" = "+i+" + "+j);  
     } //输出所有可能的素数对  
   }  
}  
}  
class fun{  
public int fun (int a)    //判断是否是素数的函数  
{  
int i,flag=0;  
if(a==3){flag=1;return(flag);}  
for(i=2;i<=Math.sqrt(a);i++){  
   if(a%i==0) {flag=0;break;}  
      else flag=1;}  
return (flag) ;//不是素数,返回0,是素数,返回1  
}  
}  
//解法二  
import java.util.\*;  
public class lianxi44 {  
public static void main(String[] args) {  
Scanner s = new Scanner(System.in);  
int n;  
do{  
     System.out.print("请输入一个大于等于6的偶数：");  
     n = s.nextInt();  
    } while(n<6||n%2!=0);   //判断输入是否是>=6偶数,不是,重新输入

    for(int i=3;i<=n/2;i+=2){  
    if(fun(i)&&fun(n-i)) {  
      System.out.println(n+" = "+i+" + "+(n-i));  
      } //输出所有可能的素数对  
   }  
}  
static boolean fun (int a){    //判断是否是素数的函数  
boolean flag=false;  
if(a==3){flag=true;return(flag);}  
for(int i=2;i<=Math.sqrt(a);i++){  
   if(a%i==0) {flag=false;break;}  
      else flag=true;}  
return (flag) ;  
}  
}  
【程序45】     
题目：判断一个素数能被几个9整除     
//题目错了吧？能被9整除的就不是素数了！所以改成整数了。  
import java.util.\*;  
public class lianxi45 {   
public static void main (String[] args) {   
   Scanner s = new Scanner(System.in);  
   System.out.print("请输入一个整数：");  
    int num = s.nextInt();  
    int   tmp = num;  
    int count = 0;   
       for(int i = 0 ; tmp%9 == 0 ;){  
           tmp = tmp/9;  
            count ++;  
          }  
     System.out.println(num+" 能够被 "+count+" 个9整除。");  
     }  
}   
【程序46】     
题目：两个字符串连接程序     
import java.util.\*;  
public class lianxi46 {  
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    System.out.print("请输入一个字符串：");  
    String str1 = s.nextLine();  
    System.out.print("请再输入一个字符串：");  
    String str2 = s.nextLine();  
    String str = str1+str2;  
    System.out.println("连接后的字符串是："+str);  
    }  
    }   
【程序47】     
题目：读取7个数（1—50）的整数值，每读取一个值，程序打印出该值个数的＊。     
import java.util.\*;  
public class lianxi47 {  
public static void main(String[] args) {  
Scanner s = new Scanner(System.in);  
int n=1,num;  
while(n<=7){  
         do{  
          System.out.print("请输入一个1--50之间的整数：");  
             num= s.nextInt();  
           }while(num<1||num>50);  
      for(int i=1;i<=num;i++)  
      {System.out.print("\*");  
      }  
System.out.println();  
n ++;  
}  
}  
}   
【程序48】     
题目：某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，加密规则如下：每位数字都加上5,然后用和除以10的余数代替该数字，再将第一位和第四位交换，第二位和第三位交换。     
import java.util.\*;  
public class lianxi48   {   
public static void main(String args[]) {   
Scanner s = new Scanner(System.in);  
int num=0,temp;  
do{  
   System.out.print("请输入一个4位正整数：");  
      num = s.nextInt();  
     }while (num<1000||num>9999);   
int a[]=new int[4];   
a[0] = num/1000; //取千位的数字   
a[1] = (num/100)%10; //取百位的数字   
a[2] = (num/10)%10; //取十位的数字   
a[3] = num%10; //取个位的数字   
for(int j=0;j<4;j++)   
{   
a[j]+=5;   
a[j]%=10;   
}   
for(int j=0;j<=1;j++)   
    {   
    temp = a[j];   
    a[j] = a[3-j];   
    a[3-j] =temp;   
    }   
System.out.print("加密后的数字为：");   
for(int j=0;j<4;j++)   
System.out.print(a[j]);   
}   
}   
【程序49】     
题目：计算字符串中子串出现的次数     
import java.util.\*;  
public class lianxi49 {   
public static void main(String args[]){  
Scanner s = new Scanner(System.in);  
    System.out.print("请输入字符串：");  
    String str1 = s.nextLine();  
    System.out.print("请输入子串：");  
    String str2 = s.nextLine();  
int count=0;   
if(str1.equals("")||str2.equals(""))   
   {   
   System.out.println("你没有输入字符串或子串,无法比较!");   
   System.exit(0);   
   }   
else   
   {   
    for(int i=0;i<=str1.length()-str2.length();i++)   
     {   
     if(str2.equals(str1.substring(i, str2.length()+i)))   
      //这种比法有问题，会把"aaa"看成有2个"aa"子串。   
       count++;   
       }   
System.out.println("子串在字符串中出现: "+count+" 次");   
}   
}  
}   
【程序50】     
题目：有五个学生，每个学生有3门课的成绩，从键盘输入以上数据（包括学生号，姓名，三门课成绩），计算出平均成绩，把原有的数据和计算出的平均分数存放在磁盘文件 "stud "中。  
import java.io.\*;   
import java.util.\*;  
public class lianxi50 {   
public static void main(String[] args){   
   Scanner ss = new Scanner(System.in);  
   String [][] a = new String[5][6];  
   for(int i=1; i<6; i++) {  
    System.out.print("请输入第"+i+"个学生的学号：");  
    a[i-1][0] = ss.nextLine();  
    System.out.print("请输入第"+i+"个学生的姓名：");  
    a[i-1][1] = ss.nextLine();  
    for(int j=1; j<4; j++) {  
       System.out.print("请输入该学生的第"+j+"个成绩：");  
       a[i-1][j+1] = ss.nextLine();  
       }  
System.out.println("\n");  
   }  
//以下计算平均分  
float avg;  
int sum;  
for(int i=0; i<5; i++) {  
sum=0;  
   for(int j=2; j<5; j++) {  
   sum=sum+ Integer.parseInt(a[i][j]);  
      }  
   avg= (float)sum/3;  
   a[i][5]=String.valueOf(avg);  
}  
//以下写磁盘文件   
String s1;   
try {   
    File f = new File("C:\\stud");   
    if(f.exists()){   
      System.out.println("文件存在");   
      }else{   
         System.out.println("文件不存在，正在创建文件");   
          f.createNewFile();//不存在则创建   
        }   
BufferedWriter output = new BufferedWriter(new FileWriter(f));   
for(int i=0; i<5; i++) {  
for(int j=0; j<6; j++) {  
   s1=a[i][j]+"\r\n";  
   output.write(s1);      
    }  
}  
output.close();   
System.out.println("数据已写入c盘文件stud中！");  
   } catch (Exception e) {   
     e.printStackTrace();   
     }   
}  
}

自己写的程序：

1. **判断一个数是否为素数**

**public** **class** lianxi33 {

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

**for** (**int** i = 1; i <=10000; i++) {

**int** n = i;

**if** (*isPrime*(n)) {

System.*out*.println(i + "是素数");

}

}

}

// 输入一个数判断其是否为素数

**public** **static** **boolean** isPrime(**int** n) {

**if** (n <= 1) {

**return** **false**;

}

**if** (n == 2) {

**return** **true**;

}

**if** (n % 2 == 0) {

**return** **false**;

}

**for** (**int** i = 3; i <= (**int**) (Math.*floor*(Math.*sqrt*(n))) + 1; i = i + 2) {

**if** (n % i == 0) {

**return** **false**;

}

}

**return** **true**;

}

}

2、二分法查找

**public** **class** BinarySearch {

**public** **static** **int** binarySearch(**int**[] a, **int** x) {

// 在a[0]<=a[1]<=...<=a[n-1]中搜索x

// 找到x则返回x的所在位置，否则返回-1

**int** left = 0;

**int** right = a.length - 1;

**while** (left <= right) {

**int** middle = (left + right) / 2;

**if** (x == a[middle])

**return** middle;

**if** (x > a[middle])

left = middle + 1;

**else**

right = middle - 1;

}

**return** -1;

}

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

**int**[] a = { 0, 1, 3, 6, 7, 10, 21, 34, 36, 40, 76 };

**int** x = 34;// 设定要查找的数

**int** position = *binarySearch*(a, x);

System.*out*.println(x + "在数组中的位置是" + position);

}

}

1. 比较器 二分查找

**import** java.util.TreeSet;

**import** java.util.Iterator;

**public** **class** Student **implements** Comparable<Student> {

**private** **int** id;

**private** String name;

**private** String dengji;

**public** Student (**int** id, String name, String dengji) {

**this**.id = id;

**this**.name = name;

**this**.dengji=dengji;

}

**public** **void** setId (**int** id) {

**this**.id = id;

}

**public** **void** setName (String name) {

**this**.name = name;

}

**public** **void** setDengji (String dengji) {

**this**.dengji = dengji;

}

**public** **int** getId () {

**return** id;

}

**public** String getName () {

**return** name;

}

**public** String getdengji () {

**return** dengji;

}

/\* Student 类的字符串表达式，形如：

\* 2 张三 \*/

**public** String toString () {

**return** (id + "\t" + name+"\t" + dengji);

}

/\* 实现 Comparable 接口中的 compareTo 方法，

\* 通常大于时返回一个正数，小于时返回一个负数，

\* 等于时返回零，具体情况可以自行决定。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 这里我根据 id 号的大小进行了比较。由于 TreeSet

\* 会根据 compareTo 的结果来排序，因此输出结果

\* 应该是按照 id 号从小到大排序的。

\* 如果要根据姓名进行排序，只需对这个方法进行相应的修改。\*/

**public** **int** compareTo (Student arg) {

**if** (id > arg.id)

**return** 1;

**else** **if** (id == arg.id)

**return** 0;

**else**

**return** -1;

}

/\* 以下为主方法，输出结果是：

\* 3 John

\* 5 Tom

\* 7 Alice

\* 9 David

\* 可以看到不同于输入顺序，

\* TreeSet 已经将其排序了。\*/

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

TreeSet<Student> tset = **new** TreeSet<Student>();

tset.add(**new** Student(5, "Tom" , "a"));

tset.add(**new** Student(3, "John", "b" ));

tset.add(**new** Student(9, "David", "c" ));

tset.add(**new** Student(7, "Alice" , "d"));

Iterator<Student> itor = tset.iterator();

**while** (itor.hasNext()) {

System.*out*.println(itor.next().toString());

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BinarySearch

**import** java.util.Comparator;

**public** **class** BinarySearch {

**public** **static** <T **extends** Comparable<T>> **int** binarySearch(T[] a, T x,

Comparator<T> comparator) {

**if** (a == **null**) {

**throw** **new** IllegalArgumentException("被查找数组为null！");

} **else** **if** (x == **null**) {

**throw** **new** IllegalArgumentException("被查询元素为null！");

}

**int** left = 0;

**int** right = a.length - 1;

**while** (left <= right) {

**int** middle = (left + right) / 2;

**if** (comparator != **null**) {

**if** (comparator.compare(x, a[middle]) == 0)

**return** middle;

**if** (comparator.compare(x, a[middle]) > 0)

left = middle + 1;

**else**

right = middle - 1;

}

**else** { **if**(x.compareTo(a[middle])==0)

**return** middle;

**if**(x.compareTo(a[middle])>0)

left = middle + 1;

**else** right = middle - 1; }

}

**return** -1;

}

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

Student[] a = **new** Student[4];

a[0] = (**new** Student(3, "Tom"));

a[1] = (**new** Student(5, "John"));

a[2] = (**new** Student(7, "David"));

a[3] = (**new** Student(9, "Alice"));

Student x = **new** Student(7, "John");// 设定要查找的数

**int** position = *binarySearch*(a, x, **new** StudentComparator());

System.*out*.println(x + "在数组中的位置是" + position);

}

}

Student

import java.util.TreeSet;

import java.util.Iterator;

public class Student implements Comparable<Student> {

private int id;

private String name;

public Student (int id, String name) {

this.id = id;

this.name = name;

}

public void setId (int id) {

this.id = id;

}

public void setName (String name) {

this.name = name;

}

public int getId () {

return id;

}

public String getName () {

return name;

}

/\* Student 类的字符串表达式，形如：

\* 2 张三 \*/

public String toString () {

return (id + "\t" + name+"\t" );

}

/\* 实现 Comparable 接口中的 compareTo 方法，

\* 通常大于时返回一个正数，小于时返回一个负数，

\* 等于时返回零，具体情况可以自行决定。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 这里我根据 id 号的大小进行了比较。由于 TreeSet

\* 会根据 compareTo 的结果来排序，因此输出结果

\* 应该是按照 id 号从小到大排序的。

\* 如果要根据姓名进行排序，只需对这个方法进行相应的修改。\*/

public int compareTo (Student arg) {

if (id > arg.id)

return 1;

else if (id == arg.id)

return 0;

else

return -1;

}

/\* 以下为主方法，输出结果是：

\* 3 John

\* 5 Tom

\* 7 Alice

\* 9 David

\* 可以看到不同于输入顺序，

\* TreeSet 已经将其排序了。\*/

public static void main (String args[]) {

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

tset.add(new Student(5, "Tom" ));

tset.add(new Student(3, "John"));

tset.add(new Student(9, "David"));

tset.add(new Student(7, "Alice"));

Iterator<Student> itor = tset.iterator();

while (itor.hasNext()) {

System.out.println(itor.next().toString());

}

}

}

StudentComparator

**import** java.util.Comparator;

/\*\*

\* Student 比较器

\*/

**public** **class** StudentComparator **implements** Comparator<Student>

{

@Override

**public** **int** compare(Student s1, Student s2)

{

**if**(s1.getId()>s2.getId())

**return** 1;

**else** **if**(s1.getId()==s2.getId())

**return** 0;

**else**

**return** -1;

}

}

package test1;   
   
/\*\*  
 \* Title: Hello Java World  
 \* Description: 简单的Java程序,只显示一个信息。  
 \* filename: HelloWorld.java  
 \*/   
 public class HelloWorld {   
  public static void main(String[] args) {   
    System.out.println("Hello Java World!");   
  }   
 }

package test2;   
/\*\*  
 \* Title: Java语言流程演示  
 \* Description: 演示Java中几种常用的流程控制操作  
 \* Filename: flowDome.java  
 \*/   
 public class flowDemo{   
   public static void main(String[] arges){   
     int iPara1,iPara2,iEnd;   
     if(arges.length!=3)   
     {   
       System.out.println("USE :java flowDome parameter1 parameter2 circle");   
       System.out.println("parameter1 : 比较条件1，数字类型");   
       System.out.println("parameter2 : 比较条件2，数字类型");   
       System.out.println("circle ：循环次数");   
       System.out.println("ego:java flowDome 1 2 5");   
       return;   
     }else{   
       iPara1 = Integer.parseInt(arges[0]);   
       iPara2 = Integer.parseInt(arges[1]);   
       iEnd = Integer.parseInt(arges[2]);   
     }   
     //if语句   
     if(iPara2>iPara1)   
     {   
      System.out.println("if 条件满足！");   
      System.out.println("第2个数比第1个数大！");   
     }   
     else   
     {   
      System.out.println("if 条件不满足！");   
      System.out.println("第2个数比第1个数小！");   
     }   
     //for循环操作   
     for(int i=0;i<iEnd;i++)   
     {   
       System.out.println("这是for 第"+i+"次循环");   
     }   
     //while循环操作   
     int i=0;   
     while(i<iEnd)   
     {   
      System.out.println("这是while 第"+i+"次循环");   
      i++;   
     }   
     //do-while循环操作   
     int j=0;   
     do   
     {   
      System.out.println("这是do-while 第"+j+"次循环");   
      j++;   
     }while(j<iEnd);   
   }   
 }

package test3;   
   
/\*\*  
 \* Title: 数组数据操作  
 \* Description: 演示一维数组和多维数组的初始化和基本操作  
 \* Filename: myArray.java  
 \*/   
 public class  myArray{   
   //初始化数组变量   
   char[] cNum = {'1','2','3','4','5','6','7','8','9','0'};   
   char[] cStr = {'a','b','c','d','e','f','g','h',   
                  'i','j','k','l','m','n','o','p',   
                  'q','r','s','t','u','v','w','x','y','z'};   
   int[] iMonth = {31,28,31,30,31,30,31,31,30,31,30,31};   
   String[] sMail = {"@","."};   
/\*\*  
 \*<br>方法说明：校验电子邮件  
 \*<br>输入参数：String sPara 被校验的电子邮件字符  
 \*<br>返回类型：boolean 如果校验的格式符合电子邮件格式返回true；否则返回false  
 \*/      
   public boolean isMail(String sPara){   
    for(int i=0;i<sMail.length;i++){   
      if(sPara.indexOf(sMail[i])==-1)   
        return false;            
    }   
    return true;   
   }   
/\*\*  
 \*<br>方法说明：判断是否是数字  
 \*<br>输入参数：String sPara。 需要判断的字符串  
 \*<br>返回类型：boolean。如果都是数字类型，返回true；否则返回false  
 \*/      
   public boolean isNumber(String sPara){   
     int iPLength = sPara.length();   
     for(int i=0;i<iPLength;i++){   
      char cTemp = sPara.charAt(i);   
      boolean bTemp = false;   
      for(int j=0;j<cNum.length;j++){   
        if(cTemp==cNum[j]){   
          bTemp = true;   
          break;   
        }   
      }   
      if(!bTemp) return false;    
     }   
    return true;   
   }   
/\*\*  
 \*<br>方法说明：判断是否都是英文字符  
 \*<br>输入参数：String sPara。要检查的字符  
 \*<br>返回类型：boolean。如果都是字符返回true；反之为false  
 \*/      
   public boolean isString(String sPara){   
     int iPLength = sPara.length();   
     for(int i=0;i<iPLength;i++){   
      char cTemp = sPara.charAt(i);   
      boolean bTemp = false;   
      for(int j=0;j<cStr.length;j++){   
        if(cTemp==cStr[j]){   
          bTemp = true;   
          break;   
        }   
      }   
      if(!bTemp) return false;    
     }   
    return true;   
   }   
/\*\*  
 \*<br>方法说明：判断是否是闰年  
 \*<br>输入参数：int iPara。要判断的年份  
 \*<br>返回类型：boolean。如果是闰年返回true，否则返回false  
 \*/      
   public boolean chickDay(int iPara){   
     return iPara%100==0&&iPara%4==0;   
   }   
/\*\*  
 \*<br>方法说明：检查日期格式是否正确  
 \*<br>输入参数：String sPara。要检查的日期字符  
 \*<br>返回类型：int 0 日期格式正确，-1 月或这日不合要求， -2 年月日格式不正确   
 \*/   
   public int chickData(String sPara){

    @SuppressWarnings("unused")   
    boolean bTemp = false；   
    //所输入日期长度不正确   
    if(sPara.length()!=10) return -2;

//获取年   
    String sYear = sPara.substring(0,4);

//判断年是否为数字   
    if(!isNumber(sYear)) return -2;

//获取月份   
    String sMonth = sPara.substring(5,7); 2011 05 05  
    //判断月份是否为数字   
    if(!isNumber(sMonth)) return -2;   
    //获取日   
    String sDay = sPara.substring(8,10);   
    //判断日是否为数字   
    if(!isNumber(sDay)) return -2;   
    //将年、月、日转换为数字

    int iYear = Integer.parseInt(sYear);   
    int iMon = Integer.parseInt(sMonth);   
    int iDay = Integer.parseInt(sDay);   
    if(iMon>12) return -1;   
    //闰年二月处理   
    if(iMon==2&&chickDay(iYear)){   
      if(iDay>29) return 2;   
    }else{   
      if(iDay>iMonth[iMon-1]) return -1;   
    }   
    return 0;   
   }   
/\*\*  
 \*<br>方法说明：主方法，测试用  
 \*<br>输入参数：  
 \*<br>返回类型：  
 \*/    
   public static void main(String[] arges){   
     myArray mA = new myArray();   
     //校验邮件地址   
     boolean bMail = mA.isMail("[tom@163.com](mailto:tom@163.com)");   
     System.out.println("1 bMail is "+bMail);   
     bMail = mA.isMail("[tom@163com](mailto:tom@163com)");   
     System.out.println("2 bMail is "+bMail);   
     //演示是否是数字   
     boolean bIsNum = mA.isNumber("1234");   
     System.out.println("1：bIsNum="+bIsNum);   
     bIsNum = mA.isNumber("123r4");   
     System.out.println("2：bIsNum="+bIsNum);   
     //演示是否是英文字符   
     boolean bIsStr = mA.isString("wer");   
     System.out.println("1：bIsStr="+bIsStr);   
     bIsStr = mA.isString("wer3");   
     System.out.println("2：bIsStr="+bIsStr);   
     //演示检查日期   
     int iIsTime = mA.chickData("2003-12-98");   
     System.out.println("1：iIsTime="+iIsTime);   
     iIsTime = mA.chickData("2003-111-08");   
     System.out.println("2：iIsTime="+iIsTime);   
     iIsTime = mA.chickData("2003-10-08");   
     System.out.println("3：iIsTime="+iIsTime);   
     iIsTime = mA.chickData("2000-02-30");   
     System.out.println("4：iIsTime="+iIsTime);   
   }   
 }

package test4;   
   
import java.util.\*;   
/\*\*  
 \* Title: 矢量操作<  
 \* Description: 演示一个矢量（Vector）的基本操作  
 \* Filename: operateVector.java  
 \*/   
public class operateVector    
{   
/\*  
\*<br>方法说明：生成一个4\*4的二维Vector，供使用。  
\*<br>输入参数：  
\*<br>输出变量：Vector  
\*<br>其它说明：  
\*/   
    public Vector<Object> buildVector(){   
       Vector<Object> vTemps = new Vector<Object>();   
       for(int i=0;i<4;i++){   
          Vector<Object> vTemp = new Vector<Object>();   
          for (int j=0;j<4;j++){   
            vTemp.addElement("Vector("+i+")("+j+")");   
          }   
          vTemps.addElement(vTemp);   
       }   
       return vTemps;   
    }   
/\*  
\*<br>方法说明：插入数据  
\*<br>输入参数：Vector vTemp 待插入的数据对象  
\*<br>输入参数：int iTemp 插入数据的位置  
\*<br>输入参数：Object oTemp 插入数据值  
\*<br>输出变量：Vector 结果  
\*<br>其它说明：如果插入位置超出实例实际的位置将返回null

\*/   
    public Vector<Object> insert(Vector<Object> vTemp,int iTemp,Object oTemp){   
        if(iTemp>vTemp.size()){   
            print("数据超界!");   
            return null;   
        }else{   
             vTemp.insertElementAt(oTemp,iTemp);   
        }   
        return vTemp;   
    }   
/\*  
\*<br>方法说明：移除数据  
\*<br>输入参数：Vector vTemp 待删除矢量对象  
\*<br>输入参数：int iTemp 删除数据的位置  
\*<br>输出变量：Vector  
\*<br>其它说明：如果删除超界的数据，将返回null  
\*/   
    public Vector<Object> delete(Vector<Object> vTemp,int iTemp){   
        if(iTemp>vTemp.size()){   
            print("数据超界!");   
            return null;   
        }else{   
             vTemp.removeElementAt(iTemp);   
        }   
        return vTemp;   
    }   
/\*  
\*<br>方法说明：修改数据  
\*<br>输入参数：Vector vTemp 待修改矢量对象  
\*<br>输入参数：int iTemp 修改数据的位置  
\*<br>输入参数：Object oTemp 修改数据值  
\*<br>输出变量：Vector  
\*<br>其它说明：如果修改位置超界的数据，将返回null  
\*/   
    public Vector<Object> updata(Vector<Object> vTemp,int iTemp,Object oTemp){   
        if(iTemp>vTemp.size()){   
            print("数据超界!");   
            return null;   
        }else{   
             vTemp.setElementAt(oTemp,iTemp);   
        }   
        return vTemp;   
    }   
/\*  
\*<br>方法说明：输出信息  
\*<br>输入参数：String sTemp 输出信息名称  
\*<br>输入参数：Object oTemp 输出信息值  
\*<br>返回变量：无  
\*/   
    public void print(String sTemp,Vector<Object> oTemp){   
        System.out.println(sTemp+"数据：");   
        this.print(oTemp);   
    }   
/\*\*  
 \*<br>方法说明：打印输出（过载）  
 \*<br>输入参数：Object oPara 输出的对象  
 \*<br>返回类型：无  
 \*/   
    public void print(Object oPara){   
        System.out.println(oPara);   
    }   
/\*\*  
 \*<br>方法说明：打印输出（过载）  
 \*<br>输入参数：Vector vPara 显示输出矢量对象  
 \*<br>返回类型：无  
 \*/   
    public void print(Vector<Object> vPara){   
       for(int i=0;i<vPara.size();i++){   
        System.out.println(vPara.elementAt(i));   
       }   
    }   
/\*\*  
 \*<br>方法说明：主方法，程序入口  
 \*<br>输入参数：String[] args  
 \*<br>返回类型：无  
 \*/   
    public static void main(String[] args)    
    {   
        operateVector ov = new operateVector();   
        Vector<Object> vTemp = ov.buildVector();   
        ov.print("vTemp0",vTemp);   
        Vector<Object> vResult = ov.insert(vTemp,2,"添加的数据");   
        ov.print("vResult",vResult);   
        Vector<Object> vResultup = ov.updata(vResult,2,"修改的数据");   
        ov.print("vResultmp",vResultup);   
        Vector<Object> vResultnow = ov.delete(vResultup,2);   
        ov.print("vResultnow",vResultnow);   
    }   
}

package test5;   
   
import java.util.\*;   
/\*\*  
 \* Title: 哈希表操作  
 \* Description: 这是一个权限认证的例子，使用了哈希表作为数据的存储  
 \* Filename: RoleRight.java  
 \*/   
 public class RoleRight   
 {   
 private static Hashtable<String, String> rightList = new Hashtable<String, String>();   
/\*\*  
 \*<br>方法说明：初始化数据  
 \*<br>输入参数：  
 \*<br>返回类型：  
 \*/   
 public void init()   
 {   
    String[] accRoleList = {"admin","satrap","manager","user","guest"};   
    String[] rightCodeList = {"10001","10011","10021","20011","24011"};   
    for(int i=0;i<accRoleList.length;i++)   
    {   
      rightList.put(accRoleList[i],rightCodeList[i]);   
    }   
 }   
/\*\*  
 \*<br>方法说明：获取角色权限代码  
 \*<br>输入参数：String accRole 角色名称  
 \*<br>返回类型：String 权限代码  
 \*/   
 public String getRight(String accRole)   
 {   
    if(rightList.containsKey(accRole))   
      return (String)rightList.get(accRole);   
    else   
      return null;   
 }   
/\*\*  
 \*<br>方法说明：添加角色和代码信息  
 \*<br>输入参数：String accRole 角色名称  
 \*<br>输入参数：String rightCode 角色权限代码   
 \*<br>返回类型：void （无）  
 \*/   
 public void insert(String accRole,String rightCode)   
 {   
   rightList.put(accRole,rightCode);   
 }   
/\*\*  
 \*<br>方法说明：删除角色权限  
 \*<br>输入参数：String accRole 角色名称  
 \*<br>返回类型：void（无）  
 \*/   
 public void delete(String accRole)   
 {   
   if(rightList.containsKey(accRole))   
     rightList.remove(accRole);   
 }   
/\*\*  
 \*<br>方法说明：修改角色权限代码  
 \*<br>输入参数：String accRole 角色名称  
 \*<br>输入参数：String rightCode 角色权限代码   
 \*<br>返回类型：void（无）  
 \*/   
 public void update(String accRole,String rightCode)   
 {   
   //this.delete(accRole);   
   this.insert(accRole,rightCode);   
 }   
/\*\*  
 \*<br>方法说明：打印哈希表中角色和代码对应表  
 \*<br>输入参数：无  
 \*<br>返回类型：无  
 \*/   
 public void print()   
 {   
    Enumeration<String> RLKey = rightList.keys();   
    while(RLKey.hasMoreElements())   
    {   
        String accRole = RLKey.nextElement().toString();   
        print(accRole+"="+this.getRight(accRole));   
    }   
 }   
/\*\*  
 \*<br>方法说明：打印信息（过载）  
 \*<br>输入参数：Object oPara 打印的信息内容  
 \*<br>返回类型：无  
 \*/   
 public void print(Object oPara)   
 {   
    System.out.println(oPara);   
 }   
/\*\*  
 \*<br>方法说明：主方法，  
 \*<br>输入参数：  
 \*<br>返回类型：  
 \*/   
 public static void main(String[] args)   
 {   
    RoleRight RR = new RoleRight();   
    RR.init();   
    RR.print();   
    RR.print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");   
    RR.insert("presider","10110");   
    RR.print();   
    RR.print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");   
    RR.update("presider","10100");   
    RR.print();   
    RR.print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");   
    RR.delete("presider");   
    RR.print();   
 }    
 }//end:)~

package test6;   
   
/\*\*  
 \* Title: 树参数  
 \* Description: 使用继承类，柳树就是树  
 \* Filename: osier.java  
 \*/   
class tree   
{   
/\*\*  
 \*<br>方法说明：树的树根  
 \*/   
  public void root()   
  {   
    String sSite = "土壤中";   
    String sFunction = "吸收养份";   
    print("位置："+sSite);   
    print("功能："+sFunction);   
  }   
/\*\*  
 \*方法说明：树的树干  
 \*/   
  public void bolo()   
  {   
    String sSite = "地面";   
    String sFunction = "传递养份";   
    print("位置："+sSite);   
    print("功能："+sFunction);   
  }   
/\*\*  
 \*方法说明：树的树枝  
 \*/   
  public void branch()   
  {   
    String sSite = "树干上";   
    String sFunction = "传递养份";   
    print("位置："+sSite);   
    print("功能："+sFunction);   
  }   
/\*\*  
 \*方法说明：树的叶子  
 \*/   
  public void leaf()   
  {   
    String sSite = "树梢";   
    String sFunction = "光合作用";   
    String sColor = "绿色";

    print("位置："+sSite);   
    print("功能："+sFunction);   
    print("颜色："+sColor);   
  }   
/\*\*  
 \*方法说明：显示信息  
 \*输入参数：Object oPara 显示的信息  
 \*/   
  public void print(Object oPara)   
  {   
    System.out.println(oPara);   
  }   
/\*\*  
 \*方法说明：主方法：  
 \*/   
  public static void  main(String[] arges)   
  {   
    tree t = new tree();   
    t.print("描述一棵树：");   
    t.print("树根：");   
    t.root();   
    t.print("树干：");   
    t.bolo();   
    t.print("树枝：");   
    t.branch();   
    t.print("树叶：");   
    t.leaf();   
  }   
}   
/\*\*  
 \* Title: 柳树参数  
 \* Description: 描述柳树的参数  
 \*/   
class osier extends tree   
{   
 /\*\*  
 \*方法说明：过载树的树叶  
 \*/   
  public void leaf()   
  {   
    super.leaf();   
    String sShape = "长形";   
    super.print("形状："+sShape);   
  }   
  /\*\*  
 \*方法说明：扩展树的花  
 \*/   
  public void flower()   
  {   
    print("哈哈，柳树没有花！！");   
  }   
/\*\*  
 \*方法说明：主方法  
 \*/   
  public static void  main(String[] args)   
  {   
    osier o = new osier();   
    o.print("柳树树根：");   
    o.root();   
    o.print("柳树树干：");   
    o.bolo();   
    o.print("柳树树枝：");   
    o.branch();   
    o.print("柳树树叶：");   
    o.leaf();   
    o.print("柳树花：");   
    o.flower();   
  }   
}

package test7;   
   
/\*\*  
 \*  Title:  接口和抽象函数   
 \*  Description: 演示继承抽象函数和实现接口   
 \*  Filename: newPlay.java  
 \*/   
    
//接口    
interface player   
{   
 int flag = 1;   
 void play();//播放   
 void pause();//暂停   
 void stop();//停止   
}//end :)   
   
//抽象类   
abstract class playing   
{   
 public void display(Object oPara)   
 {   
   System.out.println(oPara);     
 }   
 abstract void winRun();   
}//end :)   
   
//继承了playing抽象类和实现类player接口   
public class newPlay extends playing implements player   
{   
  public void play()   
  {   
    display("newPlay.play()");//这里只是演示，去掉了代码。   
  }   
  public void pause()   
  {   
     display("newPlay.pause()");//这里只是演示，去掉了代码。   
  }   
  public void stop()   
  {   
    display("newPlay.stop()");//这里只是演示，去掉了代码。   
  }   
  void winRun()   
  {   
    display("newPlay.winRun()");//这里只是演示，去掉了代码。   
  }   
  public static void main(String[] args)   
  {   
    newPlay p = new newPlay();   
    p.play();   
    p.pause();   
    p.stop();   
    p.winRun();   
  }   
}//end :)

package test8.com;   
   
/\*\*  
 \* Title: 标识符  
 \* Description: 演示标识符对类的访问控制  
 \* Filename:  
 \*/   
public class classDemo1 {   
    // 公有方法   
    public void mechod1() {   
        System.out.println("这是一个公有的方法！任何类都可以访问。");   
    }   
   
    // 授保护的方法   
    protected void mechod2() {   
        System.out.println("这是一个受到保护的方法！只有子类可以访问。");   
    }   
   
    // 私有的方法   
    private void mechod3() {   
        System.out.println("这是一个私有的方法！只有类本身才可以访问。");   
    }   
   
    public static void main(String[] args) {   
        classDemo1 d = new classDemo1();   
        d.mechod1();   
        d.mechod2();   
        d.mechod3();   
    }   
}

package test8.com;   
/\*\*  
 \* Title: 标识符  
 \* Description: 演示标识符对类的访问控制  
 \* Filename:   
 \*/   
public class classPlay   
{   
  public static void main(String[] args){   
    classDemo1 d = new classDemo1();   
    d.mechod1();   
    d.mechod2();   
    //d.mechod3();   
  }   
}

package test8.net;   
   
import test8.com.classDemo1;   
/\*\*  
 \* Title: 标识符  
 \* Description: 演示标识符对类的访问控制  
 \* Filename:   
 \*/   
public class classPlay   
{   
  public static void main(String[] args){   
    classDemo1 d = new classDemo1();   
    d.mechod1();   
  //d.mechod2();   
  //d.mechod3();   
  }   
}

package test9;   
   
/\*\*  
 \* Title: 捕获异常和实现自己的异常  
 \* Description: 通过继承Exception类来实现自己的异常类。并使用try－catch来捕获这个异常。  
 \* Filename:  
 \*/   
class MyException extends Exception {   
    private static final long serialVersionUID = 1L;   
   
    public MyException() {   
    }   
   
    public MyException(String msg) {   
        super(msg);   
    }   
   
    public MyException(String msg, int x) {   
        super(msg);   
        i = x;   
    }   
   
    public int val() {   
        return i;   
    }   
   
    private int i;   
}   
   
public class DemoException {   
    /\*\*  
     \*方法说明：使用MyException类中默认的构造器  
     \*/   
    public static void a() throws MyException {   
        System.out.println("Throwing MyException from a()");   
        throw new MyException();   
    }   
   
    /\*\*  
     \*方法说明：使用MyException类中带信息的构造器  
     \*/   
    public static void b() throws MyException {   
        System.out.println("Throwing MyException from b()");   
        throw new MyException("Originated in b()");   
    }   
   
    /\*\*  
     \*方法说明：使用了MyException中有编码的构造器  
     \*/   
    public static void c() throws MyException {   
        System.out.println("Throwing MyException from c()");   
        throw new MyException("Originated in c()", 47);   
    }   
   
    public static void main(String[] args) {   
        try {   
            a();   
        } catch (MyException e) {   
            e.getMessage();   
        }   
        try {   
            b();   
        } catch (MyException e) {   
            e.toString();   
        }   
        try {   
            c();   
        } catch (MyException e) {   
            e.printStackTrace();   
            System.out.println("error code: " + e.val());   
        }   
    }   
} // end :)

package test10;   
   
import javax.swing.\*;   
import java.awt.\*;   
   
/\*\*  
 \* Title: 创建自己的窗体   
 \* Description:   
 \* Filename:mainFrame.java  
 \*/   
public class mainFrame extends JFrame {   
   
    private static final long serialVersionUID = 1L;   
   
    /\*\*  
     \*方法说明：构造器，通过传递参数来完成窗体的绘制。   
     \*输入参数：String sTitle 窗体标题   
     \*输入参数：int iWidth 窗体的宽度  
     \*输入参数：int iHeight 窗体的高度 返回类型：  
     \*/   
    public mainFrame(String sTitle, int iWidth, int iHeight) {   
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();// 获取屏幕尺寸   
        ImageIcon ii = new ImageIcon("middle.gif");   
        setTitle(sTitle);// 设置窗体标题   
        setIconImage(ii.getImage());// 设置窗体的图标   
        setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);// 设置但关闭窗体时退出程序   
        setSize(iWidth, iHeight);// 设置窗体大小   
        int w = getSize().width;// 获取窗体宽度   
        int h = getSize().height;// 获取窗体高度   
        System.out.println("窗体宽：" + w + " 窗体高：" + h);   
        int x = (dim.width - w) / 2;   
        int y = (dim.height - h) / 2;   
        setLocation(x, y);// 将窗体移到屏幕中间   
        setVisible(true);// 显示窗体   
    }   
   
    public static void main(String[] args) {   
        JFrame.setDefaultLookAndFeelDecorated(true);// 使用最新的SWING外观   
        new mainFrame("main Frame Demo", 400, 300);   
    }   
}