**章节练习题及面试题**

# 分支结构：

## 写出结果

class Demo{

public static void main(String[] args){

int m=0,n=3;

if(m>0)

if(n>2)

System.out.println("A");

else

System.out.println("B");

}

}

//answer:没有结果

## 实现对三个整数进行排序，输出时按照从小到大的顺序输出。

## 编写程序，从键盘接收整数参数。如果该数为1-7，打印对应的星期值，否则打印“非法参数”。

|  |
| --- |
| import java.util.Scanner;  class TestSwitch{  public static void main(String[] args){  Scanner input = new Scanner(System.in);  System.out.print("请输入星期值：");  int week = input.nextInt();    switch(week){  case 1:  System.out.println("星期一：Monday");  break;  case 2:  System.out.println("星期二：Tuesday");  break;  case 3:  System.out.println("星期三：Wednesday");  break;  case 4:  System.out.println("星期四：Thursday");  break;  case 5:  System.out.println("星期五：Friday");  break;  case 6:  System.out.println("星期六：Saturday");  break;  case 7:  System.out.println("星期天：Sunday");  break;  default:  System.out.println("非法星期值");  break;  }    }  } |

## 从键盘分别输入年、月、日，判断这一天是当年的第几天

|  |
| --- |
| import java.util.Scanner;  class TestDaysOfYear{  public static void main(String[] args){  Scanner input = new Scanner(System.in);    System.out.print("年：");  int year = input.nextInt();    System.out.print("月：");  int month = input.nextInt();    System.out.print("日：");  int day = input.nextInt();    int days = day;    //加前面几个月的满月天数  switch(month){  case 12:  //前面11个月的总天数  //days += 第11月的天数;  days += 30;  case 11:  //前面10个月的总天数  //days += 第10月的天数;  days += 31;  case 10:  days += 30;//九月  case 9:  days += 31;//八月  case 8:  days += 31;//七月  case 7:  days += 30;//六月  case 6:  days += 31;//五月  case 5:  days += 30;//四月  case 4:  days += 31;//三月  case 3:  days += 28;//二月  /\*if(闰年){  days++;  }  \*/  if(year % 4 ==0 && year % 100 != 0 || year%400==0){  days++;  }  case 2:  days += 31;//一月  }    System.out.println(year + "年" + month + "月" + day + "日是这一年的第" + days + "天");    /\*  int days = 0;  switch(month){  case 1:  days = day;  break;  case 2:  days = 31 + day;  break;  case 3:  //days = 31 + 二月的天数 + day;  days = 31 + 28 + day;  break;  case 4:  //days = 31 + 二月的天数 + 31 + day;  days = 31 + 28 + 31 + day;  break;  ....  }    if(闰年 && month >2){  days++;  }  \*/    }  } |

## switch是否能作用在byte上，是否能作用在long上，是否能作用在String上

答：switch（expr1）中，expr1是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是

int、 short、 char 或者 byte。long不能作用于swtich.JDK1.7新加入了String类型。

## 编写程序，判断给定的某个年份是否是闰年

闰年的判断规则如下：

（1）若某个年份能被4整除但不能被100整除，则是闰年。

（2）若某个年份能被400整除，则也是闰年。

提示：

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

## 要求用户输入两个数a和b，如果a能被b整除或者a加b大于1000，则输出a；否则输出b。

## 使用条件结构实现，如果用户名等于字符‘青’，密码等于数字‘123’，就输出“欢迎你，青”，否则就输出“对不起，你不是青”。

提示：先声明两个变量，一个是char型的，用来存放用户名，一个是int型的，用来存放密码。

## 编写程序：从键盘上读入一个学生成绩，存放在变量score中，根据score的值输出其对应的成绩等级：

score>=90 等级：A

70=<score<90 等级：B

60=<score<70 等级：C

score<60 等级：D

|  |
| --- |
| 建议使用if...else  import java.util.Scanner;  class Exam1{  public static void main(String[] args){  Scanner input = new Scanner(System.in);  System.out.print("请输入学生的成绩：");  int score = input.nextInt();  if(score>=90){  System.out.println("等级：A");  }else if(score>=70){  System.out.println("等级：B");  }else if(score>=60){  System.out.println("等级：C");  }else{  System.out.println("等级：D");  }    /\*  if(score>=90){  System.out.println("等级：A");  }else if(score<90 && score>=70){  System.out.println("等级：B");  }else if(score<70 && score>=60){  System.out.println("等级：C");  }else{  System.out.println("等级：D");  }  \*/  }  } |

## 根据指定月份，打印该月份所属的季节。

3,4,5 春季 6,7,8 夏季 9,10,11 秋季 12, 1, 2 冬季

|  |
| --- |
| import java.util.Scanner;  class TestSeason{  public static void main(String[] args){  Scanner input = new Scanner(System.in);  System.out.println("请输入当前月份：");  int month = input.nextInt();    switch(month){  case 3:  case 4:  case 5:  System.out.println("春季");  break;  case 6:  case 7:  case 8:  System.out.println("夏季");  break;  case 9:  case 10:  case 11:  System.out.println("秋季");  break;  case 12:  case 1:  case 2:  System.out.println("冬季");  break;  default:  System.out.println("非法月份");  }  }  } |

## 求ax2+bx+c=0方程的根。

求ax2+bx+c=0方程的根。a,b,c分别为函数的参数，

如果a≠0，那么：

1. 当b2-4ac>0，则有两个解；
2. 当b2-4ac=0，则有一个解；
3. 当b2-4ac<0，则无解；

如果a=0,b≠0，那么，

提示1：Math.sqrt(num); sqrt指平方根

例如：

求x2-4x+1=0方程的根

求x2-2x+1=0方程的根

|  |
| --- |
| import java.util.Scanner;  class Exer5{  public static void main(String[] args){  Scanner input = new Scanner(System.in);    System.out.println("一元二次方程：ax^2+bx+c=0");  System.out.print("请输入参数a：");  double a = input.nextDouble();    System.out.print("请输入参数b：");  double b = input.nextDouble();    System.out.print("请输入参数c：");  double c = input.nextDouble();  if(a!=0){  double temp = b\*b - 4\*a\*c;  if(temp==0){  double x = -b/(2\*a);  System.out.println("该方程是一元二次方法，有两个相同解：" + x);  }else if(temp>0){  double sqrt = Math.sqrt(temp);  double x1 = (-b+ sqrt)/(2\*a);  double x2 = (-b- sqrt)/(2\*a);  System.out.println("该方程是一元二次方法，两个不同解：" + x1 +"," + x2);  }else{  System.out.println("该方程是一元二次方法，在实数范围内无解！");  }  }else{  if(b!=0){  double x = -c/b;  System.out.println("该方程是一元一次方程，有一个解：" + x);  }else{  System.out.println("不是方程，是一个等式");  if(c == 0){  System.out.println("等式成立");  }else{  System.out.println("等式不成立");  }  }  }  }  } |

## 彩票游戏

假设你想开发一个玩彩票的游戏，程序随机地产生一个两位数的彩票，提示用户输入一个两位数，然后按照下面的规则判定用户是否能赢。

1)如果用户输入的数匹配彩票的实际顺序，奖金10 000美元。

2)如果用户输入的所有数字匹配彩票的所有数字，但顺序不一致，奖金 3 000美元。

3)如果用户输入的一个数字仅满足顺序情况下匹配彩票的一个数字，奖金1 000美元。

4)如果用户输入的一个数字仅满足非顺序情况下匹配彩票的一个数字，奖金500美元。

5)如果用户输入的数字没有匹配任何一个数字，则彩票作废。

提示：使用Math.random() 产生随机数

Math.random() 产生[0,1)范围的随机值

Math.random() \* 90：[0,90)

Math.random() \* 90 + 10：[10,100) 即得到 [10,99]

使用(int)(Math.random() \* 90 + 10)产生一个两位数的随机数。

|  |
| --- |
| import java.util.Scanner;  class TestCaiPiao{  public static void main(String[] args){  //1、随机产生一个两位数  //System.out.println(Math.random());//产生[0,1)  int number = (int)(Math.random()\*90 + 10);//得到[10,99]，即[10,100)  //System.out.println(number);    int numberShi = number/10;  int numberGe = number%10;    //2、用户输入一个两位数  Scanner input = new Scanner(System.in);  System.out.print("请输入一个两位数：");  int guess = input.nextInt();    int guessShi = guess/10;  int guessGe = guess%10;    if(number == guess){  System.out.println("奖金10 000美元");  }else if(numberShi == guessGe && numberGe == guessShi){  System.out.println("奖金3 000美元");  }else if(numberShi==guessShi || numberGe == guessGe){  System.out.println("奖金1 000美元");  }else if(numberShi==guessGe || numberGe == guessShi){  System.out.println("奖金500美元");  }else{  System.out.println("没中奖");  }    System.out.println("中奖号码是：" + number);  }  } |

## 赌数游戏

提供三个1-6的随机数，作为掷骰子得到的点数。如果各个点数相同，则为豹子。

如果三个骰子点数和，小于或等于9，则为“小”。

如果三个骰子点数和，大于9，则为“大”。

用户随机选择：押大、押小或者押豹子。通过判断，输出客户是否押正确。

|  |
| --- |
| import java.util.Scanner;  class GuessNumber{  public static void main(String[] args){  //1、产生一个[1-6]的数  int a = (int)(Math.random()\*6+1);  //System.out.println(a);  int b = (int)(Math.random()\*6+1);  //System.out.println(b);  int c = (int)(Math.random()\*6+1);  //System.out.println(c);    /\*  a = 1;  b = 1;  c = 1;  \*/    String result = "";  if(a==b && b==c){  result = "豹子";  }else if(a+b+c <=9 ){  result = "小";  }else{  result = “大”;  }    //2、用户输入猜的结果  Scanner input = new Scanner(System.in);  System.out.print("请押宝，买定离手：(选择：大、小、豹子)");  String guess = input.next();    //3、判断结果  if(result.equals(guess)){  System.out.println("猜对了");  }else{  System.out.println("猜错了");  }  }  } |

## 生肖问题

编写一个程序，为一个给定的年份找出其对应的中国生肖。

中国的生肖基于12年一个周期，每年用一个动物代表：

rat（鼠）、ox（牛）、tiger（虎）、rabbit（兔）、dragon（龙）、snake（蛇）、

horse（马）、sheep（羊）、monkey（候）、rooster（鸡）、dog（狗）、pig（猪）。

提示：2019年：猪 2019 % 12 == 3

|  |
| --- |
| import java.util.Scanner;  class ShengXiaoTest{  public static void main(String[] args){  Scanner input = new Scanner(System.in);  System.out.print("请输入年份：");  int year = input.nextInt();    switch(year%12){  case 0:  System.out.println("猴年");  break;  case 1:  System.out.println("鸡年");  break;  case 2:  System.out.println("狗年");  break;  case 3:  System.out.println("猪年");  break;  case 4:  System.out.println("鼠年");  break;  case 5:  System.out.println("牛年");  break;  case 6:  System.out.println("虎年");  break;  case 7:  System.out.println("兔年");  break;  case 8:  System.out.println("龙年");  break;  case 9:  System.out.println("蛇年");  break;  case 10:  System.out.println("马年");  break;  case 11:  System.out.println("羊年");  break;    }  }  } |

## switch是否能作用在byte上，是否能作用在long上，是否能作用在String上？

|  |
| --- |
| 答案一：switch可以作用在byte上，不能作用在long上，JDK1.7之后可以作用在String上。 |
| 答案二：switch支持的类型byte,short,int,char，JDK1.5之后支持枚举，JDK1.7之后支持String类型。 |

## 下面程序片段的输出结果是？

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **int** a = 3;  **int** b = 1;  **if**(a = b){  System.***out***.println("Equal");  }**else**{  System.***out***.println("Not Equal");  }  } |
| 答案：编译不通过 |

# 循环结构：

## What is the result when you compile and run the following code?

public class Test{

public void method(){

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

System.out.print(i);

}

System.out.print(i);

}

}

A. 0122

B. 0123

C. compile error

D. none of these

答案：C

## 打印1-100之间13的倍数，使用for循环

## 使用双重循环打印20 \* 8的矩形，使用for循环实现

## 用for循环计算1000以内偶数的和

## 执行如下代码后，c的值是多少？

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **int** a = 0;  **int** c = 0;  **do** {  --c;  a = a - 1;  } **while** (a >= 0);  System.***out***.println("c = " + c);  } |
| 答案：c = -1 |

## 以下代码的运行结果？

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **int** i=10;  **while**(i>0){  i = i +1;  **if**(i==10){  **break**;  }  }  System.***out***.println("i=" + i);  } |
| 答案一：是一个负数，因为i一直累加会超过int的存储范围 |
| 答案二：死循环 |

## 修正如下代码

下面是一段程序，目的是输出10个=，但是不小心代码写错了，现在需要修改代码，使得程序完成功能，但是只能“增加”或“修改”其中“一个”字符，很明显，将i--改为i++，可以完成功能，但是需要修改“两个”字符，所以并不是一个正确的答案？

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **int** n=10;  **for** (**int** i = 0; i < n; i--) {  System.***out***.println("=");  }  } |
| i<n修改为-i<n |

## 1)输入长和宽，输出长方形，如：输入4和3, 将输出如下图形

####

####

####

## 2)输入高度，输出直角三角形。如：输入4, 将输出如下图形

#

##

###

####

## 3)输入高度，输出倒直角三角形。如：输入4, 将输出如下图形

####

###

##

#

## 打印九九乘法表

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **for**(**int** i =1;i<=9;i++){  **for**(**int** j=1;j<=i;j++){  System.***out***.print(j+"\*"+i+"="+(i\*j) + "\t");  }  System.***out***.println();  }  } |

## 3000米长的绳子，每天减一半。问多少天这个绳子会小于5米？不考虑小数。

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **int** day = 0;  **for** (**int** x = 3000; x >= 5; x /= 2) {  day++;  }  System.***out***.println("day=" + day);  /\*  \* 方法二：  \* day = 0;  \* for(int x=3000; x>=5; day++) {  \* x = x/2;  \* }  \* System.out.println(day);  \*/  } |

## 打印1-100之间非13的倍数，使用continue语句

## 用循环控制语句打印输出：1+3+5+...+99=?的结果

|  |
| --- |
| 答案一：  **public** **static** **void** main(String[] args) {  **int** sum = 0;  **for** (**int** i = 1; i <= 99; i+=2) {  sum += i;  }  System.***out***.println("sum = " + sum);  } |
| 答案二：  **public** **static** **void** main(String[] args) {  **int** sum = 0;  **for** (**int** i = 1; i < 100; i++) {  **if** (i % 2 != 0) {  sum += i;  }  }  System.***out***.println("sum = " + sum);  } |

# 混合结构练习

## 写出结果

public class Demo{

public static void main(String []args){

int i = 0, j = 5;

tp: for (;;){

i++;

for(;;){

if(i > j--)

break tp;

}

}

System.out.println("i = " + i + ", j = "+ j); //i=1,j=-1;

}

}

## 输出从1到100之间所有不能被3整除的数;并输出这些整数的和

## 猜数字游戏

随机生成一个100以内的数，猜数字游戏：

从键盘输入数，如果大了提示，大了，如果小了，提示小了，如果对了，就不再猜了，并统计一共猜了多少次？

提示：随机数

import java.util.Random;

Random rand = new Random();

int num= rand.nextInt(100);

|  |
| --- |
| import java.util.Random;  import java.util.Scanner;  public class TestDoWhileLoop{  public static void main(String[] args){  //1、随机产生一个100以内的整数  Random rand = new Random();  //int num = rand.nextInt();//产生的是任意大小的整数  int num = rand.nextInt(100);//产生[0,100)的整数  System.out.println(num);    //2、键盘输入  Scanner input = new Scanner(System.in);    //声明变量  int guess;  int count = 0;  do{  //循环体至少执行一次  System.out.print("请输入一个整数：");  guess = input.nextInt();//为变量赋值    count++;//输入一次，计数一次    if(guess>num){  System.out.println("大了");  }else if(guess < num){  System.out.println("小了");  }else{  System.out.println("猜对了");  }  }while(guess != num);    System.out.println("一共猜了：" + count + "次");  }    } |

|  |
| --- |
| public static void main(String[] args) {  Random rand = new Random();  int num= rand.nextInt(100);  Scanner input = new Scanner(System.in);  int count =0 ;  do{  count++;  System.out.println("请猜：");  int temp = input.nextInt();  if(temp<num){  System.out.println("小了");  continue;  }  if(temp>num){  System.out.println("大了");  continue;  }  if(temp == num){  break;  }  }while(true);  System.out.println("总共猜了"+count+"次");  } |

## 打印1-100之间数，其中3、5、7的倍数不打印（continue）

|  |
| --- |
| class TestContinue {  public static void main(String[] args) {  //打印1-100之间数，其中3、5、7的倍数不打印  for(int i=1; i<=100; i++){  if(i%3==0 | i%5==0 | i%7==0){  //跳过下面的打印语句，提前进入下一次循环，即i++语句  //break;//结束循环  continue;  }    //下面这部分循环体语句，有些情况下需要跳过  System.out.println(i);  }  System.out.println("over");  }  } |

## 一个数如果恰好等于它的因子之和，这个数就称为"完数"。（因子：除去这个数本身的约数）

例如6=1＋2＋3.编程 找出1000以内的所有完数

|  |
| --- |
| **public** **static** **void** main(String[] args) {  System.***out***.println("1-1000之间的完数有：");  **for**(**int** i=1;i<=1000;i++){  **int** sum = 0;  **for**(**int** j=1;j<i;j++){  **if**(i%j==0){  sum +=j;  }  }  **if**(sum == i){  System.***out***.println(i);  }  }  } |

## 输入两个正整数m和n，求其最大公约数和最小公倍数

|  |
| --- |
| int m = 12, n = 28;  //获取m和n的较大值  int max = (m > n)? m : n;  //获取m和n的较小值  int min = (m < n)? m : n;  //求m和n的最大公约数  for(int i = min;i >= 1;i--){  if( m % i == 0 && n % i == 0){  System.out.println("m和n的最大公约数是：" + i);  break;  }  }  //求m和n的最小公倍数  for(int i = max;i <= m \* n;i++){  if( i % m == 0 && i % n == 0){  System.out.println("m和n的最小公倍数是：" + i);  break;  }  } |

## 根据指定月份，打印该月份所属的季节

分别使用if-else if-else语句和switch-case语句：

3,4,5 春季 6,7,8 夏季 9,10,11 秋季 12, 1, 2 冬季

[第一种]

if(x==3 || x==4 || x==5)

System.out.println(x+"春季");

else if(x==6 || x==7 || x==8)

System.out.println(x+"夏季");

else if(x==9 || x==10 || x==11)

System.out.println(x+"秋季");

else if(x==12 || x==1 || x==2)

System.out.println(x+"冬季");

else

System.out.println(x+"月份不存在");

[第二种]

if(x>12 || x<1)

System.out.println(x+"月份不存在");

else if(x>=3 && x<=5)

System.out.println(x+"春季");

else if(x>=6 && x<=8)

System.out.println(x+"夏季");

else if(x>=9 && x<=11)

System.out.println(x+"秋季");

else

System.out.println(x+"冬季");

[第三种]

public static void main(String[] args) {

int x = 4;

switch(x){

case 3:

case 4:

case 5:

System.out.println(x+"春季");

break;

case 6:

case 7:

case 8:

System.out.println(x+"夏季");

break;

case 9:

case 10:

case 11:

System.out.println(x+"秋季");

break;

case 12:

case 1:

case 2:

System.out.println(x+"冬季");

break;

default:

System.out.println("nono");

}

## 已知学生成绩以100分为满分，共分5个等级：A,B,C,D,E

90～100为等级A，

80～89为等级B，

70～79为等级C，

60～69为等级D，

0～59为等级E。

要求定义一个成绩变量，当成绩变化时，可直接知道该成绩对应的等级。

例如：当成绩为100时，该学生的等级是A。

|  |
| --- |
| **class** LevelDemo{  //定义一功能，通过给定分数，获取该分数对应的等级。  **public** **static** **void** main(String[] args){  **int** num = 89;  **if**(num>=90 && num<=100)  System.***out***.println("level = A");  **else** **if**(num>=80 && num<=89)  System.***out***.println("level = B");  **else** **if**(num>=70 && num<=79)  System.***out***.println("level = C");  **else** **if**(num>=60 && num<=69)  System.***out***.println("level = D");  **else**  System.***out***.println("level = E");    }  } |

## 打印自然数的个数

1）打印1~100之间 6的倍数的个数

2）求出1～100之间，既是3又是7的倍数的自然数出现的次数？

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **int** count1 = 0,count2 = 0;  **for** (**int** x = 1; x <= 100; x++) {  **if** (x % 6 == 0){  count1++;  }  **if**(x % 3 == 0 && x % 7 == 0){  count2++;  }    }  System.***out***.println("count1=" + count1);  System.***out***.println("count2=" + count2);  } |

## 求调和级数中从第多少项开始值大于10

调和级数的第n项形式为：1+1/2+1/3+…+1/n

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **double** sum = 0.0;  **int** i = 1;  **while** (**true**) {  sum += 1.0 / i;  **if** (sum > 10) {  **break**;  }  i++;  }  System.***out***.println(i);  } |

## 打印如下的图形：三角形

**\***

**\* \* \***

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

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

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

**\* \* \***

**\***

|  |
| --- |
| **for** (**int** i = 0; i < 7; i++) {  **if** (i < 4) {  **for** (**int** j = 0; j < 2 \* i + 1; j++) {  System.***out***.print("\* ");  }  System.***out***.println();  } **else** {  **for** (**int** k = 0; k < 13 - 2 \* i; k++) {  System.***out***.print("\* ");  }  System.***out***.println();  }  } |

【拓展】

## 打印如下的图形：菱形1

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

|  |
| --- |
| // 上半部分  **for** (**int** i = 0; i < 5; i++) {  // 输出“-”  **for** (**int** j = 0; j < 4 - i; j++) {  System.***out***.print(" ");  }  // 输出“\* ”  **for** (**int** k = 0; k < i + 1; k++) {  System.***out***.print("\* ");  }  System.***out***.println();  }  // 下半部分  **for** (**int** i = 0; i < 4; i++) {  **for** (**int** j = 0; j < i + 1; j++) {  System.***out***.print(" ");  }  **for** (**int** k = 0; k < 4 - i; k++) {  System.***out***.print("\* ");  }  System.***out***.println();  } |

## 打印如下的图形：菱形2

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

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

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

\*\* \*\*

\* \*

\*\* \*\*

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

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

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

## 打印如下的图形：心形

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

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

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

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

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

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

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

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

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

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

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

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

\*\*\*

\*

|  |
| --- |
| **class** Heart {  **public** **static** **void** main(String[] args) {  // 分三个大部分 上中下  **for** (**int** i = 0, k = 0; i < 14; i++) {// 打印行  // 上部分 上分为 四个部分  **if** (i < 3) {  **for** (**int** j = 0; j < 5 - 2 \* i; j++) {// 1、空心  System.***out***.print(" ");  }  **if** (i == 2) {// 2、\*  **for** (**int** j = 0; j < 6 + 4 \* i - 1; j++) {  System.***out***.print("\*");  }  **for** (**int** j = 0; j < 7 - 4 \* i + 2; j++) {// 3、空心  System.***out***.print(" ");  }  **for** (**int** j = 0; j < 6 + 4 \* i - 1; j++) {// 4、\*  System.***out***.print("\*");  }  } **else** {  **for** (**int** j = 0; j < 6 + 4 \* i; j++) {// 2、\*  System.***out***.print("\*");  }  **for** (**int** j = 0; j < 7 - 4 \* i; j++) {// 3、空心  System.***out***.print(" ");  }  **for** (**int** j = 0; j < 6 + 4 \* i; j++) {// 4、\*  System.***out***.print("\*");  }  }  } **else** **if** (i < 6) {// 中间  **for** (**int** j = 0; j < 29; j++) {  System.***out***.print("\*");  }  } **else** {// 下部分 6  **if** (i == 13) {  **for** (**int** j = 0; j < 2 \* (i - 6); j++) {// 打印空格  System.***out***.print(" ");  }  System.***out***.print("\*");  } **else** {  **for** (**int** j = 0; j < 2 \* (i - 6) + 1; j++) {// 打印空格  System.***out***.print(" ");  }  **for** (**int** j = 1; j < 28 - 4 \* k; j++) {  System.***out***.print("\*");  }  k++;  }  }  System.***out***.println();// 换行  }  }  } |

## 编写程序，打印100-200之间的质数

## 一个数如果恰好等于它的因子之和，这个数就称为"完数"

（因子：除去这个数本身正的约数）

例如6=1＋2＋3.编程 找出1000以内的所有完数

|  |
| --- |
| **public** **class** WanShu {  **static** **int** *count*;  **public** **static** **void** main(String[] args) {  **for** (**int** i = 1; i <= 1000; i++) {  **int** factor = 0;  **for** (**int** j = 1; j < i; j++) {  **if** (i % j == 0)  factor += j;  }  **if** (factor == i) {  System.***out***.println(i);  *count*++;  }  }  System.***out***.println("1-1000之间的完数个数为：" + *count*);  }  } |

## 写一个程序，找出4位数的所有吸血鬼的数字

例如：1260=21\*60

1827=21\*87

|  |
| --- |
| **public** **class** Test1 {  **public** **static** **void** main(String[] args) {  **for** (**int** num = 1001; num < 10000; num++) {  *math*(num);  }  }  **public** **static** **void** math(**int** num) {  **int**[] temp1 = **new** **int**[2];  **int**[] temp2 = **new** **int**[2];  **int** a = num / 1000;  **int** b = num / 100 % 10;  **int** c = num / 10 % 10;  **int** d = num % 10;  **int**[] data = { a, b, c, d };  **for** (**int** i = 0; i < data.length; i++) {  **for** (**int** j = 0; j < data.length; j++) {  **if** (i == j) {  **continue**;  }  temp1[0] = data[i];  temp1[1] = data[j];  **for** (**int** m = 0; m < data.length; m++) {  **if** (m != i && m != j) {  temp2[0] = data[m];  **for** (**int** n = 0; n < data.length; n++) {  **if** (n != i && n != j && n != m) {  temp2[1] = data[n];  *multi*(data, temp1, temp2);  }  }  }  }  }  }  }  **public** **static** **int** toInt(**int**[] temp) {  **int** m = 0;  **int**[] temp1 = **new** **int**[temp.length];  **for** (**int** i = 0; i < temp.length; i++) {  temp1[i] = temp[i] \* (**int**) Math.*pow*(10, temp.length - 1 - i);  }  **for** (**int** i = 0; i < temp1.length; i++) {  m += temp1[i];  }  **return** m;  }  **public** **static** **void** multi(**int**[] temp, **int**[] temp1, **int**[] temp2) {  **int** i = *toInt*(temp1);  **int** j = *toInt*(temp2);  **int** k = *toInt*(temp);  **if** (k == i \* j) {  System.***out***.println(k + "=" + i + "\*" + j);  }  }  } |

-------------------------------------------------------

/\*

## 输出所有的水仙花数

所谓水仙花数是指一个3位数，其各个位上数字立方和等于其本身。

例如： 153 = 1\*1\*1 + 3\*3\*3 + 5\*5\*5

|  |
| --- |
| **class** ShuiXianHua {  **public** **static** **void** main(String[] args) {  **for** (**int** i = 100; i < 1000; i++) {// 实现所有的三位数的一个遍历  **int** j1 = 0;  **int** j2 = 0;  **int** j3 = 0;  j1 = i / 100;// 百位  j2 = (i - 100 \* j1) / 10;// 十位  j3 = i - 100 \* j1 - 10 \* j2;// 个位  **if** (i == j1 \* j1 \* j1 + j2 \* j2 \* j2 + j3 \* j3 \* j3) {  System.***out***.println("此数值为满足条件的水仙花数:" + i);  }  }  }  } |

## 山上有一口缸可以装50升水，现在有15升水。老和尚叫小和尚下山挑水，每次可以挑5升。问：小和尚要挑几次水才可以把水缸挑满？通过编程解决这个问题。

提示：

(1) 用整型变量water表示水缸里的水“int water = 15;”。

(2) 用整型变量l表示小和尚下山挑水的次数“int l = 0;”。

(3) 分析循环条件（水少于50升），循环操作（水增加5升，挑水次数增加1）。

(4) 套用while循环（或do-while循环）写出代码。

## 生成13位条形码

Ean-13码规则：第十三位数字是前十二位数字经过计算得到的校验码。

例如：690123456789

计算其校验码的过程为：

① 前十二位的奇数位和6+0+2+4+6+8=26

② 前十二位的偶数位和9+1+3+5+7+9=34

③ 将奇数和与偶数和的三倍相加26+34\*3=128

④ 取结果的个位数：128的个位数为8

⑤ 用10减去这个个位数10-8=2

所以校验码为2

（注：如果取结果的个位数为0，那么校验码不是为10（10-0=10），而是0）

实现上述代码的功能，计算验证码，输入12位条码，返回带验证码的条码。

例：输入：692223361219输出：6922233612192

## 实现判断一个4位整数，统计出此整数里面包含多少个偶数，多少个奇数的功能

## 开发一款软件，根据公式（身高-108）\*2=体重，可以有10斤左右的浮动。来观察测试者体重是否合适。

## 有3个整数，给出提示信息：

能否创建三角形；两边之和大于第三边 三个条件都要写

如果能构建三角形，提示是直角三角形还是等边三角形等腰三角形还是普通三角形；

最后输出三角形面积；

# 其它

## 在JAVA中，如何跳出当前的多重嵌套循环？

答：用break或return 的方式。