第 4 章　循环结构

引言

本章学习可反复执行部分语句块的循环结构，循环结构包括 while、for、do-while 语句，以及可以退出循环的 break 语句和 continue 语句，本章最后对两种常用编程方法——穷举法和递推法进行介绍。

4.1　while 循环

循环结构（repetition structure）用来描述一定条件下重复执行某段语句的情况，被重复执行的语句称之为**循环体**（body of loop），循环结构包括 while 语句、do-while 语句以及 for 语句。本节介绍 while 语句。

while 语句的语法格式如下：

```
while (条件表达式)
{
    循环体语句;
}
```

while 语句的含义为当条件表达式的值为 true 时，执行循环体语句，然后再去判断条件表达式，如为 true，继续执行循环体语句，反复判断表达式值、执行循环体语句，一直到条件表达式值为 false，则执行循环体后面的语句。

while 语句执行过程如图 4-1 所示。

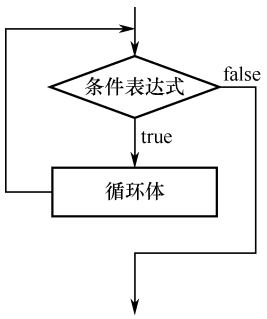

图 4-1　while 语句执行过程图

对于 while 循环语句，有下列注意事项：

①while 为小写的关键字，while 表示“当”的含义；

②条件表达式的值为 boolean 类型的值，仅能为 true 或者 false，因此条件表达式一般为 boolean 类型变量、关系表达式或逻辑表达式；

③循环体语句需放入到大括号内，可以有多条语句，若循环体语句为一条语句，大括号可以省略，但这种情况下推荐使用大括号；

④循环不能永远执行下去，因此在循环体中需要有改变循环条件表达式结果的语句，或者退出循环的语句；

⑤while 语句需先判断条件表达式为 true 时再执行循环体，因此循环体有可能一次也不被执行；

⑥若在括号后面直接带分号，如 while（true）;，表示循环体为空语句，即没有可执行的循环体。

【例 1】　编写程序输出 10 次 hello world。

思路：要反复输出 10 次 hello world，因此可以使用循环语句进行输出，循环条件是输出次数小于 10，因此需要记录循环的次数，循环前设置循环次数 i = 0，当完成一次循环后，i++，循环次数加一。

```
public class App4_1 {
```

```
        public static void main(String[] args) {
            int i=0;//循环次数为 0
            while(i<10){
               System.out.println("hello world");
               i++;//循环次数加 1
            }
        }
    }
```

【例 2】 求 1+2+3+…+100 的值。

思路: 编写程序求 1+2+3+…+100，首先设置变量 sum 存放和的结果，sum=0，然后利用 sum = sum + 1; sum = sum+2; ……; sum = sum + 100;将数据存入 sum 中，由此可见程序需要反复求和，因此需要使用 while 循环，while 循环体为 sum = sum + i; i=1,2,3,…,100;，循环执行条件为 i<=100，i 在完成一次求和后，i 值要加 1 即 i++。

```
    public class App4_2 {
        public static void main(String[] args) {
            int sum =0;                 //求和变量 sum 初值为 0
            int i = 1;
            while(i<=100){
               sum = sum + i;           //求和
               i++;
            }
            System.out.printf("sum = %d", sum);
        }
    }
```

通过【例 1】和【例 2】可以看出，使用循环语句，一般要确定反复执行的循环体语句和循环的执行条件表达式。由于循环需要有限次数循环，在循环体中应有可以改变条件表达式结果的语句，或者能够退出循环的语句。

总结【例 1】、【例 2】，若循环次数已知，while 循环使用方式如下：

设置循环条件的初值;

```
    while(条件表达式)
     {
       循环主体语句;
       改变循环条件;
    }
```

【例 3】 已知 N!=1*2*3*…*N，求大于 500 的最小阶乘的数 N。

思路: 已知 N!=1*2*3*...*N，求阶乘，定义变量 result=1 存放阶乘值，然后通过 result =result * 1;result = result * 2;result = result *3;result = result *4;……; result = result *n;，求 n 的阶乘，则反复执行 result = result *i;i++的循环体，循环的条件应为阶乘结果 result<=500。

```
     public class App4_3 {
        public static void main(String[] args) {
          int  i =0;
          int  result = 1;
          while(result < 500) {
                i++;
                result = result * i;
          }
          System.out.printf("大于 500 最小阶乘是%d,i=%d",result,i);
        }
    }
```

4.2 do-while 循环

do-while 语句的语法格式如下：

```
do{
    <循环体>;
} while (<条件表达式>)
```

do-while 语句的含义为：先执行循环体语句，然后判断条件表达式的值，当条件表达式的值为 true 时，则重复执行循环体语句，然后继续判断条件表达式的值，反复如此，一直到条件表达式的值为 false，退出循环，执行循环后面的语句。

do-while 循环的执行过程如图 4-2 所示。

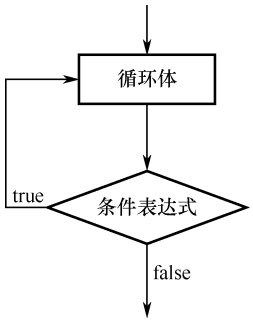

图 4-2　do-while 运行过程图

使用 do-while 语句需注意下列事项：

①do-while 循环先执行循环体，再判断条件表达式的值，因此，循环体至少要被执行一次，do-while 循环的关键字有 do 和 while，while 后面没有分号；

②循环体语句若有多条，则必须放在大括号内，若循环体语句仅有一条，可以省略大括号，但推荐放入大括号内；

③使用循环语句，循环次数应为有限次，因此在循环体中应该有使循环趋于结束的语句，否则循环将永远进行下去，形成死循环。

【例 4】　输入一个整数，将其反转并输出。例如，输入 123，反转为 321，然后输出。

思路：将键盘输入整数存入变量 x，假设 x=123，对 x 反转放入变量 y 中，y=0;，首先得到 x 的最低位 t = x%10，将 t 存入 y 中 y= y*10+ t，x 去掉最低位 x= x/10; 反复执行上面语句，一直到 x=0，表示到处理结束 x 的每一位。

```
import  java.util.Scanner;
public class App4_4 {
    public static void main(String[] args) {
        int x;                  // x 用来存放由键盘输入的正整数
        int y = 0;              // 反转后变量
        int t;
        Scanner s = new Scanner(System.in);
        System.out.println("请输入一个正整数:");
        x = s.nextInt();
        // 下面用 do-while 循环结构进行反转输出
        do {
            t = x % 10;         // 除以 10 取余数输出
            y = y * 10 + t;
            x /= 10;            // 将 x 刷新为除以 10 的商
        } while (x != 0);       // 如 x(商数)为 0 则结束循环
        System.out.printf("正整数反转输出:%d", y);
    }
}
```

4.3　for 循环

for 语句是目前比较常用的循环语句，它的语法格式为：

```
for（表达式 1;表达式 2;表达式 3）{
循环体;
}
```

for 语句的含义为：

（1）执行表达式 1；

（2）判断表达式 2 的值，如果表达式 2 的值为 true，则去执行循环体，如果为假就退出 for 语句循环，执行第（5）步；

（3）执行表达式 3；

（4）转到第（2）步；

（5）结束循环，执行 for 语句下面的一个语句。

for 语句是先判断表达式 2 的值后再执行，如果不满足判断条件，循环体可能一次都不能执行。

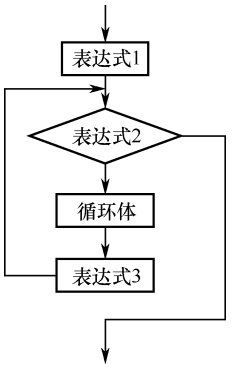

图 4-3　for 语句执行过程图

for 语句的执行过程如图 4-3 所示：

使用 for 语句需要注意以下事项：

①在 for 语句中括号中有 3 个表达式，表达式之间用分号隔开；

②在 3 个表达式中，表达式 1 仅执行一次，一般是循环变量的初始化，表达式 2 的值为 boolean 类型，一般为逻辑或关系表达式，表达式 3 一般为循环变量的增值；

③对于 3 个表达式，表达式 1 和表达式 3 都可以省略，但表达式 2 不可以省略，而且必须以分号隔开，如 for(;true;)System.out.println();；

④若有多条循环体语句，则必须放在大括号内，若仅有一条循环体语句，可以省略大括号，这种情况下不推荐省略大括号；

⑤若在 for 括号后面直接带分号，如 for(;true);，表示循环体为空语句，即没有可执行的循环体。

【例 5】　求 1 到 10 的倒数之和。

思路：定义求和变量 sum=0，然后将 1～10 倒数加到 sum 中，即 sum = sum+1.0/1，sum= sum=sum +1.0/2，…，sum=sum+1.0/10，因此可以看出需要使用循环语句，循环条件为 i=1～10，循环体为 sum=sum+1.0/i，最后输出 sum。

```
public class App4_5 {
    public static void main(String[] args) {
        int n = 1;
        double sum = 0;
        for(int i =1 ;i<=10;i++) {          // i<=10 时，累加求和；否则结束循环
            sum += 1.0/i;                   // 将自然数 n 倒数的值加到 sum 中
        }
        System.out.println("1+1/2+...+1/10=" + sum); // 输出和
    }
}
```

注意，【例 5】中，由于求倒数之和，求和变量 sum 定义为 double 类型变量，并且在求和时使用 sum+=1.0/i。

4.4　循环嵌套和编程方法

4.4.1　循环嵌套

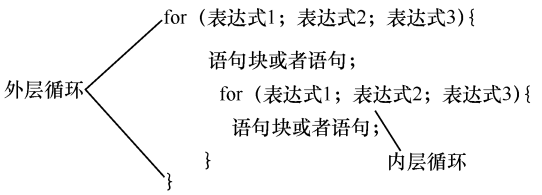

图 4-4　for 语句二层嵌套图

一个循环结构内可以包含另一个循环，称为**循环嵌套**，又称多重循环。常用的循环嵌套是二重循环，外层循环称为**外循环**，内层循环称为**内循环**。图 4-4 展示了两个 for 循环语句的二重嵌套。

在循环嵌套中，三种循环语句可以互相嵌套，并且层次不限，但是不能循环交叉。

【例 6】　输出如图 4-5 所示的九九乘法表。

```
1×1=1
1×2=2   2×2=4
1×3=3   2×3=6   3×3=9
1×4=4   2×4=8   3×4=12  4×4=16
1×5=5   2×5=10  3×5=15  4×5=20  5×5=25
1×6=6   2×6=12  3×6=18  4×6=24  5×6=30  6×6=36
1×7=7   2×7=14  3×7=21  4×7=28  5×7=35  6×7=42  7×7=49
1×8=8   2×8=16  3×8=24  4×8=32  5×8=40  6×8=48  7×8=56  8×8=64
1×9=9   2×9=18  3×9=27  4×9=36  5×9=45  6×9=54  7×9=63  8×9=72  9×9=81
```

图 4-5　九九乘法表

思路：对于每一行，要反复输出每一行数据，因此首先按行循环，循环九次（i=1，2，…，9）；在每一行，反复输出如 1*2=2 这样的运算式，所以在行中也要循环输出运算式 i*j=（j=1，2，…，i），完成输出。

```
public class App4_6{
   public static void main(String[] args) {
      for (int i = 1; i <= 9; i++) {        //循环输出每行的数据
         for (int j = 1; j <= i; j++) {     //对每一行，循环输出每一个运算式
            if (j <= i) {
               System.out.print(" "+i + "X" + j + "=" + i * j + "\t");
//输出一个运算式
            }
         }
         System.out.println("");           // 换行
      }
   }
}
```

对于循环嵌套，要注意循环的执行过程，对于外循环每执行一次，内循环需执行完全部内循环语句。

4.4.2　编程方法

【例 7】　用 0～9 这十位数字可以组成多少无重复的三位数？

思路：设置计数变量的值为 0，三位数范围从 100～999，对于每一个三位数 num=100，101，…，999，取出它的个位，十位，百位，然后判断这三位是否不同，如果不同，则计数变量+1，最后输出计数变量。

```
public class App4_7 {
    public static void main(String[] args) {
        int gw = 0;       //个位
        int sw = 0;       //百位
        int bw = 0;       //十位
        int count  = 0;   //计数变量，统计满足条件的数的个数
        for(int i=100 ;i<999;i++){
            bw = i/100;
            sw = i/10%10;
            gw = i%10;
            if(bw != gw&&bw != sw&&sw != gw )
                System.out.printf("%d ",i);
                count++;
        }
        System.out.printf("\n 满足条件的数有%d 个", count);
    }
}
```

【例 7】典型使用了“穷举法”，**穷举法**是一种常用的编程方法，这种方法需要一一列举出该问题的所有可能解，并在对逐一列举的可能解的过程中，检验每个可能解是否为问题真正的解，若是，则采纳这种解，否则抛弃它，对于所列举的值，不能遗漏也不能重复。

【例 8】　某旅行团有男人、女人和小孩共 30 人，在一家小饭店吃饭，该饭店按人头收费，

每个男人收 3 元，每个女人收 2 元，每个小孩收 1 元，共收取 50 元，男人、女人和小孩各有多少人？共有多少组解？

思路：使用穷举法解决问题，男人最少 1 个，最多 16 个；女人最少一个，最多 24 个；小孩最少一个，最多 48 个。然后对男人、女人、小孩的每一种组合情况，进行判断，是否满足情况。

```
public class App4_8 {
    public static void main(String[] args) {
        int man;
        int women;
        int child;
        int count = 0;//记录满足要求的解的个数
        for(man = 1;man< 17 ;man++)
            for(women =1 ;women< 25 ;women++)
                for(child = 1; child< 50 ;child ++)
                    if(man *3 + women * 2 + child * 1 == 50){
                        count ++;//计数加 1
                        System.out.printf("男人有%d 个，女人有%d 个，小孩有%d 个
\n",man,women,child);
                    }
        System.out.println("可能解决方法总数为："+count);
    }
}
```

【例 9】 斐波那契数列的第 1、2 项分别为 1、1，以后各项的值均是其前两项之和，求前 30 项斐波那契数。

思路：假设 f1、f2、f3 分别是第一个、第二个、第三个数，有 f1=1，f2=1，f3=f1+f2。对于后面的数，若让 f1=f2，f2=f3，f3 = f2 + f3，则可以求出后面的数，反复如此，即可求出前 30 项斐波那契数。

```
public class App4_9 {
    public static void main(String[] args) {
        long   f1 = 1;                      //第一项
        long   f2 = 1;                          //第二项
        long   f3;                              //第三项
        System.out.printf("%d %d ",f1,f2);
        for(int i=3;i<=30;i++){
              f3 = f1 + f2                      //求第三项
              f1 = f2;                          //第二项变成第一项
              f2 = f3;                          //第三项变成第二项
            System.out.printf("%d ", f3);
            if(i%5==0)
                System.out.println();      //每行 5 个换行
        }
    }
}
```

【例 9】使用了“递推法”来解决问题，**递推法**是指从初值出发，归纳出新值和旧值的关系，直到求出所需值为止。新值的求出依赖于旧值，不知道旧值，无法推导出新值，斐波那契数列求值、数学上的递推公式都是这一类的问题。

4.5 break 和 continue

4.5.1 break 语句

break 语句的基本语法格式为：

```
break;
```

break 语句的含义表示“中断”，即中断当前程序块的执行，继续执行程序块下面的语句。

break 语句只能用于退出循环（for、while、do-while）语句或者 switch 语句。当循环体中

存在 break 语句时，break 的作用是中断正在执行的最近一层循环体，执行循环体后的语句。对于 switch 语句，break 的作用是中断执行某个 case 语句，执行 switch 后的语句。

【例 10】　输入一个整数，判断该数是否是素数。

思路：若一个数是素数，则这个数只能被自己和 1 整除，如 3，5 等。输入的整数假设为 num，判断 num 是否为素数，则用 i=2、3，…，num−1 去除 num，如果有一个数能整除 num，则表示该数不为素数，若都不能整除 num，则表示该数是素数。

```
public class App4_10 {
    public static void main(String[] args) {
        Scanner  input = new Scanner(System.in);
        System.out.println("请输入整数：");
        int n = input.nextInt();
        int i;
        //判断 2…n 能否被 n 整除
        for( i = 2;i<n;i++)
            //如果 n 能被 i 整除，则不是质数，跳出循环
            if(n%i == 0)
                break;
        if(i<n)//break 退出，表示被某个数整除
            System.out.println(n +"不是质数");
        else
            System.out.println(n +"是质数");
    }
}
```

若 break 用于嵌套循环，则 break 只能退出其所在层的循环。

【例 11】　查看程序，了解 break 用法。

```
public class App4_11 {
    public static void main(String[] args) {
        int i,j;
        for(i=1;i<4;++i){
        for(j=2;j<5;++j)
              break; //break 语句只能终止离该语句最近的循环
             System.out.println("这会被输出 4 次吗?");
           }
    }
}
```

4.5.2　continue 语句

continue 语句的基本格式为：

```
continue;
```

continue 语句的中文含义为“继续”，常用于循环语句中，表示结束循环体中 continue 其后语句的执行，并返回循环语句的开头执行下一次循环。

【例 12】　打印 1～100 内，能被 5 整除的数。

思路：循环 i=1 ~ 100 内的每个数，若 i 不能被 5 整除，则打印 i。

```
public class App4_12 {
    public static void main(String[] args) {
        for(int i = 1; i<100;i++){
            //整除 5，则跳到循环开始，执行下一次循环
            if(i%5 ==0)
                continue;
            System.out.printf("%d ",i);
        }
    }
}
```

注意 break 和 continue 虽然都能改变循环语句的执行，但两者的作用是不同的，break 是中

断该层的循环的执行，跳出循环，执行循环后面的语句，而 continue 仅结束本次循环体语句的执行，返回循环语句开头执行下一次循环，而不是结束整个循环。例如下列程序段：

```
int x=0;
while(x++ < 10){
 if(x == 3){
  break;
 }
 System.out.println ("x="+x);
}
```

结果输出:

```
x=1
x=2
```

当 x =3 就退出 while 循环。

若以上程序段 break 语句改为 continue，结果则不同。

```
int x=0;
while(x++ < 10){
if(x == 3){
continue;
}
System.out.println ("x="+x);
}
```

结果输出:

```
x=1
x=2
x=4
x=5
x=6
x=7
x=8
x=9
x=10
```

当 x=3 时，continue 语句仅仅结束本次运行，返回循环最上面，执行下一次循环。

4.6 循环示例

【例 13】 计算机随机生成 10 道运算数 10 以内的加法题，用户输入每道题的运算结果，计算机判断用户输入答案是否正确，如果正确，给出提示信息“恭喜，答对了”，如果错误，给出提示信息“很遗憾答错了”，然后显示正确答案，完成 10 道题目后，显示用户答对题目的个数。

4.6.1 for 循环实现实例

思路：对每一道题，需要完成以下任务：生成两个随机数来组成加法运算式；将运算式输出到屏幕；获得用户的控制台输入的答案，对用户输入的答案和题目的正确答案进行比较判断是否正确，如果正确，答对题目个数加 1。10 道题目都需要这样做，因此需要循环以上任务 10 次，循环完成后，输出答对题目个数。

```
import java.util.Scanner;
import java.util.Random;
public class App4_13 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Random random = new Random();       // 生成随机数类
        int a = 0, b = 0;
        int count = 0;                      // 统计正确结果运算
        for (int i = 1; i <= 10; i++) {
```

```
                // 生成两个运算数 a、b 显示运算式
                a = random.nextInt(10);
                b = random.nextInt(10);
                System.out.println("请输入以下算式的计算结果：");
                System.out.printf("%d + %d =", a, b);
                // 得到用户输入的运算结果
                int result = input.nextInt();
                if (result == (a + b)) {       //判断结果是否正确
                    count++;
                    System.out.println("恭喜，答对了！");
                } else
                 System.out.println("很遗憾答错了！本题的正确答案是：”" + (a + b));
            }
            System.out.println("本次总共答对了" + count + "道题！");
        }
    }
```

4.6.2　while 循环语句实现实例

将【例 13】使用 while 循环语句完成，代码如下：

```
    import java.util.Scanner;
    import java.util.Random;
    class App4_14 {
        public static void main(String[] args) {
            Scanner input = new Scanner(System.in);
            Random random = new Random();           // 生成随机数类
            int a = 0, b = 0;
            int count = 0;                          // 统计正确结果运算
            int i = 0;                              // 循环次数
            while (i < 10) {
               // 生成两个运算数 a、b 显示运算式
                a = random.nextInt(10);
                b = random.nextInt(10);
                System.out.println("请输入以下算式的计算结果：");
                System.out.printf("%d + %d =", a, b);
                // 得到用户输入的运算结果
                int result = input.nextInt();
                if (result == (a + b)) {
                    count++;
                    System.out.println("恭喜，答对了！");
                } else
                    System.out.println("很遗憾答错了！本题的正确答案是" + (a + b));
                i++;                                // 循环次数加 1
            }
            System.out.println("本次总共答对了" + count + "道题！");
        }
    }
```

关 键 术 语

循环结构 repetition structure　　for 循环语句 for repetition statement
while 循环语句 while loop statement
do-while 循环语句 do-while repetition statement　　循环体 body of loop　　循环 loop
嵌套结构 nested control statements　　循环语句 loop statement

本 章 小 结

Java 的循环控制语句主要有 3 种，while 循环、do-while 循环和 for 循环语句。

while 循环语句和 for 循环语句是先判断循环条件，如循环条件为 true，则执行循环体，否则就退出循环，因此循环体可能一次也不执行。

for 循环经常适用于循环次数已知循环，while 语句经常适用于循环次数未知的循环。

do-while 循环先执行循环体，再判断条件表达式的值，因此，循环体至少要被执行一次。

循环语句可以嵌套，一个循环语句的循环体可以为另一个循环语句。

在循环嵌套中，三种循环语句可以互相嵌套，并且层次不限，但是不能循环交叉。

break 语句表示退出最内一层的循环，执行循环下一条语句。

continue 语句表示中断这一次的循环，去执行下一次循环。

复 习 题

一、选择题

1．下列语句段执行后，x 的值是（　　）。

```
int x=2;
do{
x+=x;
}while(x<17);
```

A．4　　B．16　　C．32　　D．256

2．下列语句执行后，c 的值是（　　）。

```
char c='\0';
for(c='a';c<'z';c+=3) {
  if(c>='e')
    break;
}
```

A．’e’　　B．’d’　　C．’f’　　D．’g’

3．阅读下面的程序：

```
public class  Test{
    public static void main(String[] args){
         for(int i=0;i<10;i++){
          if(i==3)
             continue;
             System.out.print(i);
        }
    }
}
```

程序运行后的输出是（　　）

A．0123　　B．012　　C．0123456789　　D．012456789

4．（　　）语句将至少执行一次循环语句体，且能够继续执行下去直至其循环条件变为 false 时为止。

A．while　　B．if　　C．do…while　　D．if…else

5．当 while 语句中的条件语句不会变为 false 时，会导致出现（　　）循环。

A．不确定　　B．未定义　　C．嵌套　　D．无限

6．认真阅读下面的程序：

```
public class  Test1{
    public static void main(String[] args){
         for(int i=0;i<5;i++)
```

```
                System.out.print(i+1);
            System.out.println(i);
        }
    }
```

上述程序运行后的结果是（　　）。

A．123456　　B．123455　　C．123450　　D．编译错误

7．下面程序段的运行结果是（　　）。

```
    int n=0;
    while(n++<=2); System.out.println (n);
```

A．2　　B．3　　C．4　　D．有语法错

8．运行以下程序段，其正确的运行结果是（　　）。

```
int x=-1;
do{
   x=x*x;
  }
while(!x);
```

A．是死循环　　B．循环执行两次　　C．循环执行一次　　D．有语法错误

二、简答题

1．while 语句中的条件表达式的值是什么类型？

2．for 循环语句可以代替 while 语句的作用吗？

3．break 和 continue 的区别在哪？

三、程序题

1．查找下列代码段的错误，并修改。

（1）

```
int x=1,total;
while(x<=10){
    total += x;
  x ++;
}
```

（2）

```
while(x<=100)
    total = total +x;
++x;
```

（3）

```
int y = 5;
    while (y > 0){
        System.out.println(y);
        ++y;
```

（4）

```
int y = 4;
while(y<0);
System.out.println(y);
    y++;
}
```

2．查找并修改下列代码段的错误。

（1）

```
int y=5;
do{
    System.out.println(y);
}while(y>0);
```

（2）

```
for(int y=3,y<10,y++)
```

```
            System.out.println(y+1);
```

（3）

```
        for(;;)
            System.out.println(4-2);
```

（4）

```
        for(k=0.1;k!=1.0;k+=0.1)
            System.out.println(k);
```

3．阅读下列代码，给出输出结果。

（1）

```
        public class Printing {
            public static void main(String[] args) {
                for(int i = 1; i<= 10;i++){
                    for(int j= 1;j<= 5 ;j++)
                        System.out.print("$");
                    System.out.println();
                }
            }
        }
```

（2）

```
        public class Test {
            public static void main(String[] args) {
                for(int  i = 1; i<= 5;i++){
                    for(int  j= 1;j<= 3 ;j++){
                        for(int  k=1; k<=4;k++)
                            System.out.print("$");
                        System.out.println();
                    }
                    System.out.println();
                }
        }
        }
```

（3）

```
        public class BreakTest {
            public static void main(String[] args) {
                for(int i=1;i<=10;i++){
                    if(i == 5)
                        break;
                    System.out.printf("%d",i);
                }
                System.out.printf("退出循环时候 i= %d", i);
        }
        }
```

（4）

```
        public class ContinueTest {
            public static void main(String[] args) {
                for(int i=1;i<=10;i++){
                    if(i == 5)
                        continue;
                    System.out.printf("%d",i);
                }
                System.out.printf("使用 continue , i=%d", i);
        }
        }
```

4．分别使用三种循环，求 100 以内奇数的乘积。

5．输入 10 个整数，求出 10 个数的平均值、最大值和最小值。

6．编写一个程序，输出斐波那契数列的前 20 项，并求前 20 项之和。斐波那契数列前几项为：1，1，2，3，5，8，13…

7．编写程序，求下列公式的值。

（1）e = 1+ 1/1!+1/2!+1/3!+…

（2）e^x = 1+ x/1!+x^2/2!+x^3/3!+…

8．使用循环将 1～100 内的所有素数输出到屏幕。

9．编写程序使用循环输出下列值。

N	10*N	100*N	1000*N
1	10	100	1000
2	20	200	2000
3	30	300	3000
4	40	400	4000
5	50	500	5000

10．有父子二人，已知父亲年龄不大于 90 岁，儿子年龄不大于 50 岁。10 年前父亲的年龄是儿子的 4 倍，10 年后父亲的年龄是儿子年龄的整数倍，求父子的年龄。

11．设计猜数游戏，游戏随机给出一个 0～99（包括 0，99）的数字，然后用户猜是什么数字。用户可以随便猜一个数字，游戏会提示太小还是太大，从而缩小结果范围。经过几次猜测与提示后，最终推出答案。在游戏过程中，记录用户最终猜对时所需要的次数，游戏结束后公布结果。

12．编写一个程序，有 1、2、3、4、5 个数字，计算能组成多少个互不相同且无重复数字的三位数，并且显示该数。