循环结构

一 内容回顾（列举前一天重点难点内容）

程序结构之选择结构
1.选择结构包括两种：分别为if语句和switch语句
2.if语句有四种使用形式
    第一种，只有if语句
```
  if(布尔表达式){
    }
```
    第二种 双分支if语句（有if和else语句）
```
  if(布尔表达式){
    }
    else{
  }
```
    第三种 多分支if语句（有if else if else）
```
  if(布尔表达式){
    }
    else if(布尔表达式){
  }
    else{
  }
```
    第四种 嵌套
```
  if(布尔表达式){
    if(布尔表达式){
        }
        else{
    }
    }
    else{
  }
```
3.switch语句形式
```
  switch(变量){
      case 值1:
            语句块;
        break;
        case 值2:
            语句块;
        break;
  }
```
4.switch语句的用法：
  1）根据变量的值，来寻找case的值,如果找到，执行该case下的语句，直到碰到break为止！如果没有break,则会顺序执行后面的语句。
  2）如果变量的值，不与任意一条case的值相等，则会执行default后的语句。default的位置是任意的，并且是可有可无的。
  3）变量的类型，可以是：char byte short int 枚举 String(jdk1.7)
  4）case的值必须是确定的、固定的值（常量）,不能是取值范围。
5.if和switch的使用场景
 1）如果对具体的个数的数值进行判断，用if可以，用switch也可以，建议用switch。

因为switch会把所有的备选答案加载进入内存当中，选择的效率就会更高。
2）如果要对数据的区间进行判断时，用if语句。
3）如果表达式的结果是boolean类型的，毫无疑问用if语句

二 教学目标

1.掌握循环的使用场景
2.掌握for、while、do-while循环语句的使用
3.掌握break和continue语句的使用
4.掌握多重循环的使用
5.掌握三种循环结构的区别

三 教学导读

3.1 为什么需要循环结构

到目前为止，我们学习了顺序结构和选择结构，但当我们想重复做一件事时，这两种结构显然是不能胜任的。比如：

场景1：设置闹钟每天早上6点，准时响起
场景2：哈雷彗星每76年回归一次地球
场景3：每7天为一周，每4周为一月，每12月为一年
场景4：每天都要工作、学习、吃饭、睡觉、打游戏
场景5：打印输出 1-100的所有数字
。。。。。。
等等
这时，我们就需要循环结构！

3.1 循环结构的概念

循环结构是指在程序中需要反复执行某个功能而设置的一种程序结构。它由循环体中的条件，判断继续执行某个功能还是退出循环。根据判断条件,循环结构又可细分为以下两种形式:先判断后执行的循环结构和先执行后判断的循环结构.循环结构可以看成是一个条件判断语句和一个向回转向语句的组合。

3.3 循环结构三要素

循环结构的三个要素：循环变量、循环体和循环终止条件

四 教学内容

4.1. for循环(会)

4.1.1 for循环的基本语法

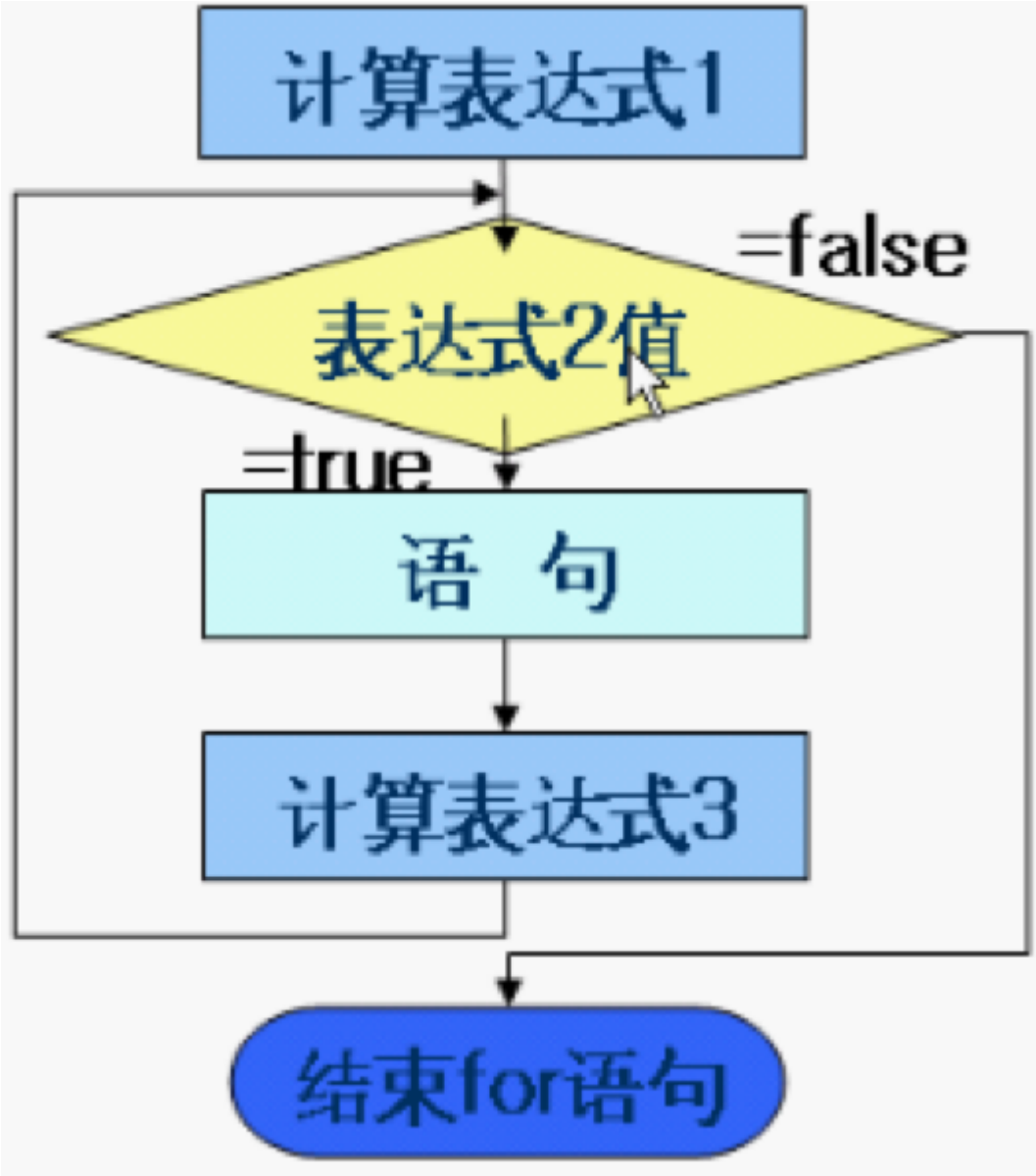
for语句由关键字 for 小括号 大括号 以及相关语句组成
基本格式如下

```
for(表达式1;表达式2;表达式3){
  循环体
}
```

表达式1:循环变量初始化语句 比如 int i = 0;
表达式2:循环终止的判断条件语句，要求为布尔表达式，也就是结果为真或假值 比如 i < 10;
表达式3:循环改变的控制条件语句 比如 i++
循环体:循环要执行的语句
三个表达式之间用分号分隔

4.1.2 for循环的执行过程

流程说明:
第一步:执行表达式1(也就是执行循环变量初始化语句)
第二步:执行表达式2(也就是执行循环终止的判断条件语句,看其结果是true还是false
如果是 false，循环结束。
如果是 true，继续执行第三步。
第三步:执行循环体语句（也就是执行 大括号中的实际代码）
第四步:执行表达式3（也就是执行循环改变的控制条件语句，使循环变量的值发生改变）
第五步:回到第二步再次执行第二到第五步（直到第二步的表达式结果为false，循环结束）




```
//实操代码
for(int i=0; i< 10; i++){
  System.out.println("当前循环变量的值是" + i);
}
/*
第一步：执行 int i=0;
第二步：执行 i<10;(此时 i值为0) 结果为true,所以程序继续执行第三步
第三步：执行 System.out.println("当前循环变量的值是" + i); 打印输出
第四步：执行 i++ (执行后 i值为 1)
第五步：执行 (回到第二步执行) i<10;(此时 i值为1) 表达式结果依然为 true,所以程序继续 (执行三到五步，直到某次第二步的语句结果为false ,程序结束)
*/
```

课上练习一

打印输出3次helloworld

```
for(int i = 0;i<3;i++){
    System.out.println("Hello World!");
}
```

课上练习二

打印100以内 能被4整除不到能被7整除的数据，每行打印6个

```
int count = 0;
for(int i = 1; i<=100; i++){
    if(i%4==0 && i%7!=0){
        System.out.print(i+"\t");
        count++;// 6
        if(count%6==0){
            System.out.print("\n");
        }
    }
}
```

4.1.3 for循环的特殊形式

```
//第一种，将表达式一省略，放到循环前面
int i=0;
for(;i<10;i++){
    System.out.println("当前循环变量的值是" + i);
}
//第二种，将表达式3省略，放到循环体中
int i=0;
for(;i<10;){
    System.out.println("当前循环变量的值是" + i);
  i++;
}
//第三种，将表达式1，2，3全部省略 此种为死循环，一般需要在循环体中给出结束条件
for( ; ; ){
  System.out.println("当前循环变量的值是" + i);
}
```

4.1.4 for循环随堂练习

课上练习一

打印1到100的所有整数

```
//注意 正常循环变量初值一般为0，这里从1开始，结束条件是否包括等于，需要根据实际需求决定
for(int i = 1 ; i <= 100 ; i++){
  System.out.println("整数值:" + i);
}
```

课上练习二

计算1到100的所有整数和

```
//注意 正常循环变量初值一般为0，这里从1开始，结束条件是否包括等于，需要根据实际需求决定
int sum = 0;
for(int i = 1 ; i <= 100 ; i++){
   sum += i;
}
System.out.println("1到100的所有整数和为:" + sum);
```

课上练习三

统计1到100之间分别能被3整除或5整除或同时被3和5整除的数字个数并打印

```
//记录能被3整除的数字个数
int threeCount = 0;
//记录能被3整除的所有数字
String threeStr = "";
//记录能被5整除的数字个数
int fiveCount = 0;
//记录能被3整除的所有数字
String fiveStr = "";
//记录能同时被3和5整除的数字个数
int threeFiveCount = 0;
//记录能同时被3和5整除的所有数字
String threeFiveStr = "";

for(int i =1 ; i <= 100 ; i++){
  if(i%3==0 && i%5==0){
    threeFiveStr += i + " ";
    threeFiveCount++;
  }
  else if(i%3==0){
    threeStr += i + " ";
    threeCount++;
  }
  else if(i%5==0){
    fiveStr += i + " ";
    fiveCount++;
  }
}
System.out.println("1到100之间能被3的数字个数为:" + threeCount);
System.out.println("分别是:" + threeStr);
System.out.println("1到100之间能被5的数字个数为:" + fiveCount);
System.out.println("分别是:" + fiveStr);
System.out.println("1到100之间同时被3和5整除的数字个数为:" + threeFiveCount);
System.out.println("分别是:" + threeFiveStr);
```

课上练习四

打印九九乘法表

```
//九九乘法表 共有45个结果，所以需要循环45次 第一行输出一个结果，第二行输出二个结果，...第九行输出九个结果
//用来记录当前行号
int row = 1;
//用来记录当前列号（也就是当前行的第几个结果）
int col = 1;
for (int i = 0; i <45 ; i++) {
    //输出row行的第col个结果,不换行
    System.out.print(col +"*"+ row+"=" + (col*row));
    //同一行中多个结果之间的分隔符
    System.out.print("\t");
    //如果行号和列号相等，说明第row行已打印完成
    if(row ==col){
      //打印换行符
       System.out.println();
      //列号重置
       col =1;
      //行号加1
       row++;
    }
    else{
      //列号加1
       col++;
    }
}
```

课上练习五

判断指定的数字是否是质数并输出

```
//用来记录是否是质数的布尔变量 false 就是质数
boolean flag = false;
//需要判断的数字，实际可由用户通过键盘输入
int checkNumber =100;
//循环判断当前数字是否能被1和它本身外的数字整除
for(int i=2; i<checkNumber;i++){
    //如果能被整除，说明不是质数
    if(checkNumber%i==0){
        //设置标质不是质数
        flag=true;
        //跳出循环，也就是只要发生过整除，后续就没必要再判断了
        break;
    }
}
//根据标记变量的真假，直接输出结果
System.out.println("整数"+checkNumber + (flag?"不是质数":"是质数") );
```

4.2. while循环(会)

4.2.1. while循环的基本语法

while语句由关键字 while 小括号 大括号 以及相关语句组成
基本格式如下：

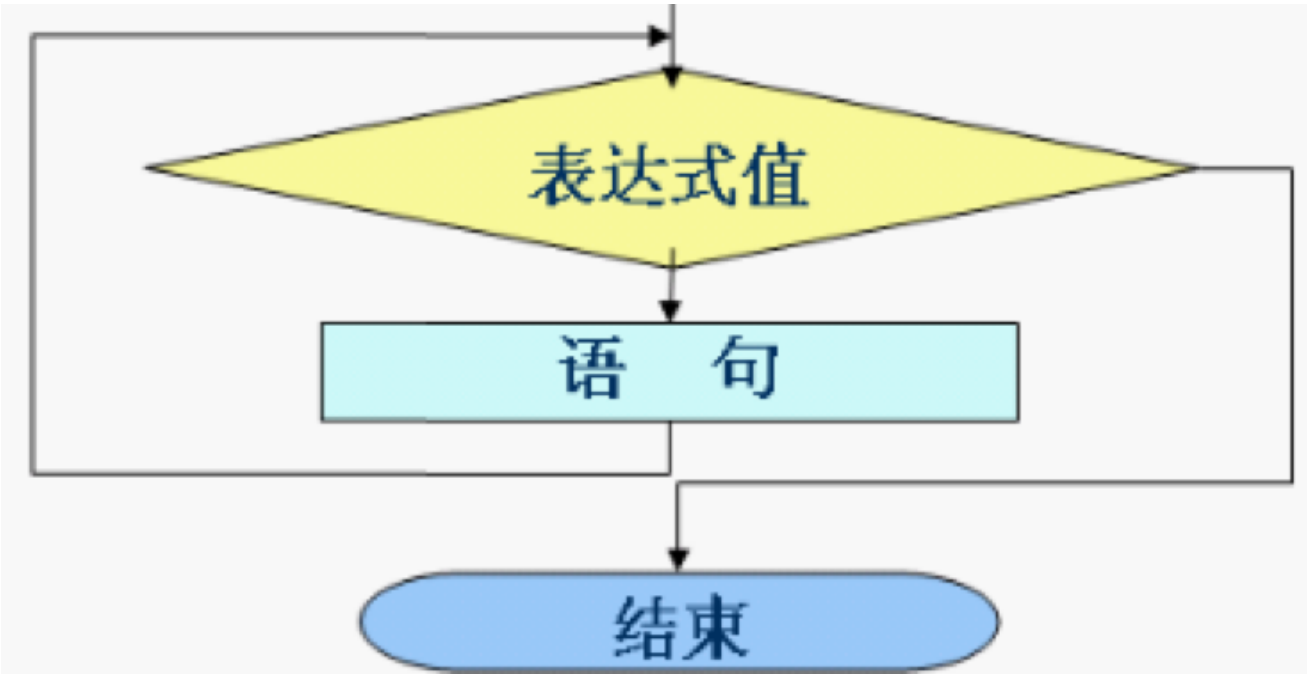
```
while(条件表达式){
        循环体
}
```

条件表达式：循环终止的判断条件语句（相当于for循环的 表达式2），要求为布尔表达式，也就是结果为真或假值 比如 i < 10;
循环体:n行循环要执行的语句
循环变量：和for循环相比，while循环缺少循环变量声明，初始值以及控制循环变量改变的语句，所以需要在循环外声明循环变量并设置初值，在循环体中，增加控制循环变量改变的语句，比如：

```
//声明循环变量并赋值
int i=0;
while(i<10){
    //代表实际要执行的循环体代码
        System.out.println("当前的i值为:" + i);
    //控制循环变量改变的语句
    i++;
}
```

4.2.2. while循环的执行过程

流程说明：
第一步:执行条件表达式(也就是执行循环终止的判断条件语句,看其结果是true还是false，如果是 false，循环结束，如果是true 继续执行)
第二步:执行循环体语句（也就是执行 大括号中的实际代码）
第三步:回到第一步再次执行 第一到第三步，直到某次第一步表达式结果为false



4.2.3 while循环的注意事项

```
1.while循环本身没有循环变量声明和初始化的组成部份，所以如果有这个需要，应在while循环前声明循环变量并赋值
2.while循环本身也没有控制循环终止的判断条件语句部分，所以需要在循环体中增加相应的控制语句，否则容易死循环
3.如果不确定循环次数的情况下，我们优先选择while循环
```

4.2.4 while循环随堂练习

课上练习一

打印输出5次 我爱编程，我爱学习大数据

```
// 初始化部分
int count = 0;
// 2循环条件
while(count<5){
    //3循环体
    System.out.println("我爱编程，我爱学习大数据");
    //4更新循环变量
    count++;
}
```

课上练习二

求 10 的阶乘

```
int sum = 1;
int i = 1;
while(i<=10){
    sum=sum*i;
    i++;
}
System.out.println("10的阶乘"+sum);
```

课上练习三

求1-100的和

```
//1初始化变量
int i=1;
int sum=0;//保存和
//2循环条件
while(i<=100){
  sum=sum+i;//sum+=i;
  i++;
}
System.out.println("1-100的和是:"+sum);
```

课上练习四

求 100以内的 偶数的和

```
int z=2;
int sum=0;
while(z<=100){
  sum=sum+z;
  z+=2;
}
System.out.println("1-100的偶数的和是:"+sum);

或
int z=1;
int sum=0;
while(z<=100){
  if(z%2==0){
    sum=sum+z;
  }
  z++;
}
System.out.println("1-100的偶数的和是:"+sum);
```

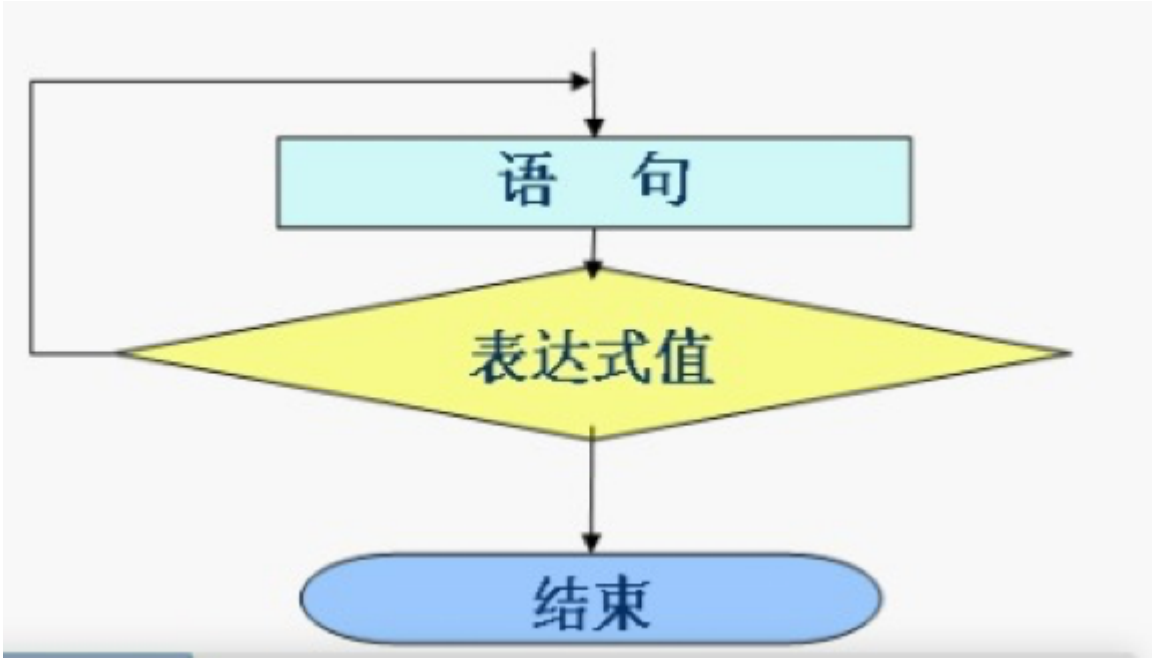
4.3. do-while循环(会)

4.3.1 do-while循环的基本语法

```
do-while语句由  do关键字 大括号 while关键字 小括号号和相关语句组成
基本格式如下:
do{
        循环体
}while(条件表达式);
```

4.3.2 do-while循环的执行过程

```
流程说明:
第一步:执行循环体语句 (也就是执行 大括号中的实际代码)
第二步:执行条件表达式(也就是执行循环终止的判断条件语句,看其结果是true还是false, 如果是 false, 循环结束,
如果是true 继续执行)
第三步:回到第一步再次执行 第一到第三步, 直到某次第一步表达式结果为false
```



4.3.3 do-while循环的特点和注意事项

1.do-while循环为先执行后判断，也就是先执行一次循环体中的代码，然后再检查条件表达式，所以do-while循环至少会执行一次
2.其它特点和while循环一样

4.3.4 do-while循环的随堂练习

课上练习一

打印三次helloworld

```
// 1 初始化部分
int i = 0;
do{
   // 2 循环体
   System.out.println("Hello World!");
   // 4 循环变量变化部分
    i++;
}while(i<3);// 3 循环条件
```

课上练习二

用do/while实现打印100以内的奇数

```
int j = 1;
do{
/*if(j%2==1){
    System.out.println(j);
}
j++;*/
    System.out.println(j);
    j+=2;
}while(j<100);
```

课上练习三

100 以内能够被3整除 但是不能被5整除的数打印输出

```
int z = 3;
do{
    if(z%3==0 && z%5!=0){
        System.out.println(z);
    }
    z++;
}while(z<=100);
```

4.3.5 while 和 do-while的区别

```
1.while 先执行循环条件，然后在执行循环体，一句话：先判断，再执行
2.do/while 先执行循环体  然后在执行循环条件，一句话：先执行，再判断
当第一次 就不满足循环条件的情况下 while循环不能执行循环体，do while 可以执行一次
```

4.3.6 几种循环的比较

```
1. 对于同一问题，三种循环可相互替代。
2. 循环次数确定的情况优先选用for循环，循环次数不确定的情况，通常选用while和do-while循环。
3. 要防止无限循环--死循环。
```

4.4. break和continue的使用(会)

4.4.1 break语句

```
1.作用：break语句用于终止某个语句块的执行
2.用法：如果是循环中，作用为跳出所在的循环，如果是在switch语句中，则为跳出所在的switch语句
3.其它说明：使用Lable标签实现跳出指定的循环
```

4.4.2 break语句随堂练习

课上练习一

跳出break语句所在的循环

```
for (int i = 1; i < 3 ; i++ ){
    for (int j = 1; j< 5 ;j++ ){
            if(j == 2){
                break;// 跳出内层循环，进行下一次外层循环
            }
            System.out.println(i+" "+j);
    }
}
```

课上练习二

跳出break语句所在的switch(根据输入的月份，输出它对应的天数，忽略闰年的情况下)

```
import java.util.Scanner;
public class Demo9{
    public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.println("请输入一个月份:");
        int month=input.nextInt();
        switch(month){
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                System.out.println(month +"月有31天");
                break;
            case 2:
                System.out.println(month +"月有28天");
                break;
            case 4:
            case 6:
            case 9:
            case 11:
                System.out.println(month +"月有30天");
                break;
            default:
                System.out.println("非地球的月份");
                break;
        }
        input.close();

    }
}
```

课上练习三

使用Label标签实现跳出指定的循环

```
out : for (int i = 1; i <3 ; i++ ){//定义一个标签out
            for (int j = 1;j<3 ;j++ ){
                if(j == 2){
                    break out;// 可以指定跳出的循环
                }
                System.out.println(i+" "+j);
            }
        }
```

4.4.3 continue语句

```
1.作用：跳过本次循环，执行下一次循环（如果有多重循环，默认继续执行离自己最近的循环）提前终止本次循环
2.使用：只能在循环结构中使用
3.其它说明：使用Label标签改变继续执行的循环
```

4.4.4 continue的随堂练习

课上练习一

当 j==2时，跳过本次循环，不进行打印输出

```
    for (int j = 1;j<4 ;j++ ){
         if(j==2){
                continue;
            }
            System.out.println("i="+i + "  j="+j);
    }
    System.out.println("Hello World!");
```

课上练习二

当 j==2时，直接跳转到out标签处继续执行

```
out: for (int i = 1;i<4 ;i++ ){
            for (int j = 1;j<4 ;j++ ){
                if(j==2){
                    continue out ;
                }
                System.out.println("i="+i + "  j="+j);
            }
        }
        System.out.println("Hello World!");
```

4.5. 多重循环(会)

4.5.1 多重循环的基本概念

1.概念：多重循环就是指 在循环内嵌套其它循环，和选择语句嵌套类似，嵌套几层就是几重循环，最常见的为双重和三重
2.说明：外层循环执行一次，它的内层循环执行一轮（也就是内循环正常循环一遍结束）

4.5.2 多重循环的执行流程

第一步:从外层循环开始执行 先执行一次进入循环体（里面包括内层循环）
第二步:遇到内层循环，开始正常执行，直到内循环循环结束
第三步:继续外循环的下一次循环（回到第一步）

4.5.3 多重循环的随堂练习（主要为双重循环）

课上练习一

用双重循环实现九九乘法表的打印

```
for (int i = 1; i <=9 ; i++) {
      for (int j = 1; j <=i ; j++) {
           System.out.print(j +"*"+ i+"=" + (i*j));
           System.out.print("\t");
      }
      System.out.println();
}
```

课上练习二

打印直角三角形

```
for (int i = 1;i<=5 ;i++ ){
    // 1 2 3 4 5
    for (int j = 1;j<=i ;j++ ){
        System.out.print("*");
  }
    System.out.println();
}
```

课上练习三

打印等腰三角形

```
for (int i = 1;i<=5 ;i++ ){
    //空格
    for(int k=1;k<=5-i;k++){
        System.out.print(" ");
    }
    // 1 2 3 4 5
    for (int j = 1;j<=i*2-1 ;j++ ){
        System.out.print("*");
    }
    System.out.println();
}
```

课上练习四

课堂练习2:键盘上输入三个数值，分表表示三角形的三条边长。需要先判断三条边长是否可以构成三角形。如果可以打印输出周长和面积，如果不能，需要键盘再次输入，最多3次

```
        Scanner scan = new Scanner(System.in);
        int count = 0;
        while (count < 3) {
            System.out.println("请输入第一个边长：");
            double a = scan.nextDouble();
            System.out.println("请输入第二个边长：");
            double b = scan.nextDouble();
            System.out.println("请输入第三个边长：");
            double c = scan.nextDouble();
            if (a + b > c && b + c > a && a + c > b) {
                double zhouchang = a + b + c;
                double p = zhouchang / 2;
                double area = Math.sqrt(p * (p - a) * (p - b) * (p - c));
                System.out.println("周长：" + zhouchang + ",面积：" + area);
                break;
            } else {
                System.out.println("您输入的三条边长无法构成三角形，请重新输入：");
            }
            count++;
        }
        if (count == 3) {
            System.out.println("输入连续3次错误数据，程序结束");
        }
        System.out.println("程序结束!");
```

五 实战应用

六 教学总结

6.1 课程重点

1. 掌握for循环的语法
2. 掌握while,dowhile循环的语法
3. 三种循环的比较
   对于同一问题，三种循环可相互替代
   for循环功能强于while，do-while.但若不是明显地给出循环变量初终值(或修改条件),则可以用while 或do-while.以增强程序的结构化和可读性。
   要防止无限循环--死循环。
4. 掌握三种循环的基本用法
5.掌握break,continue的使用

6.2 课程难点

实现双层或多层循环
解决办法:多写循环练习,过程中学着自己分析,在必要时通过观看答案解析理清思路.当我们对循环的语法理解更加清晰时,问题就会慢慢的变得简单起来.

七 课后作业

作业题:

1.将一个整数数位翻转
   如：整数 56123
   返回结果为整数： 32165

2：一个5位数，判断它是不是回文数，例如12321是回文数
3.打印下列图形

```
    *
   * *
  * * *
 * * * *
* * * * *
```

4.写出程序结果

```
public class Demo      tp:相当于给当前的循环起了一个别名.主要作用:实现在内层循环结束外层循环
{
        public static void main(String []args)
        {
                int i = 0, j = 5;
                tp: for (;;)
                {
                        i++;
                        cc:for(;;)
                        {
                                if(i > j--)
                                        break tp;
                        }
                }
```

```
                System.out.println("i = " + i + ", j = "+ j);
        }
}
```

5. 图形打印（每一个图形都需要单独设计一个方法）（注：这是四个题）

```
*****               *********
 *****               *******
  *****               *****
   *****               ***
    *****               *
```

6. 在屏幕上输出以下内容：

```
12345
1234
123
12
1
```

7. 计算10个99相加后的值并输出。
8. 计算从1加到100的值并输出。
9. 计算10的阶乘（1x2x3x4x5x6x7x8x9x10）。
10. 计算2的20次方。
11. 计算从1到1000以内所有奇数的和并输出。 //对2取余 i%2 == 1
12. 计算从1到1000以内所有能被3或者17整除的数的和并输出。// i % 3 == 0 || i%17==0
13. 计算从1到1000以内所有能同时被3，5和7整除的数的和并输出。
14. 计算1到100以内能被7或者3整除但不能同时被这两者整除的数的个数。 (i%7==0 || i%3==0) && ! (i%7==0 && i%3==0)
15. 计算1到100以内能被7整除但不是偶数的数的个数。
16. 判断n是不是一个质数（质数是只能被1和它自身整除的数）。
17. 将某个8位的整数所有位的数值加在一起并输出。

八 解决方案

8.1 应用场景

8.2 核心面试题

1. 什么时候用for循环，什么时候用while循环
2. while循环和do-while循环的区别
3. break、continue、return的区别
   break:应用在switch和循环中，作用跳出（终止）语句块
   continue:应用在循环中，作用结束本次循环，继续下一次循环。
   return :用在方法中作用返回结果，结束方法。