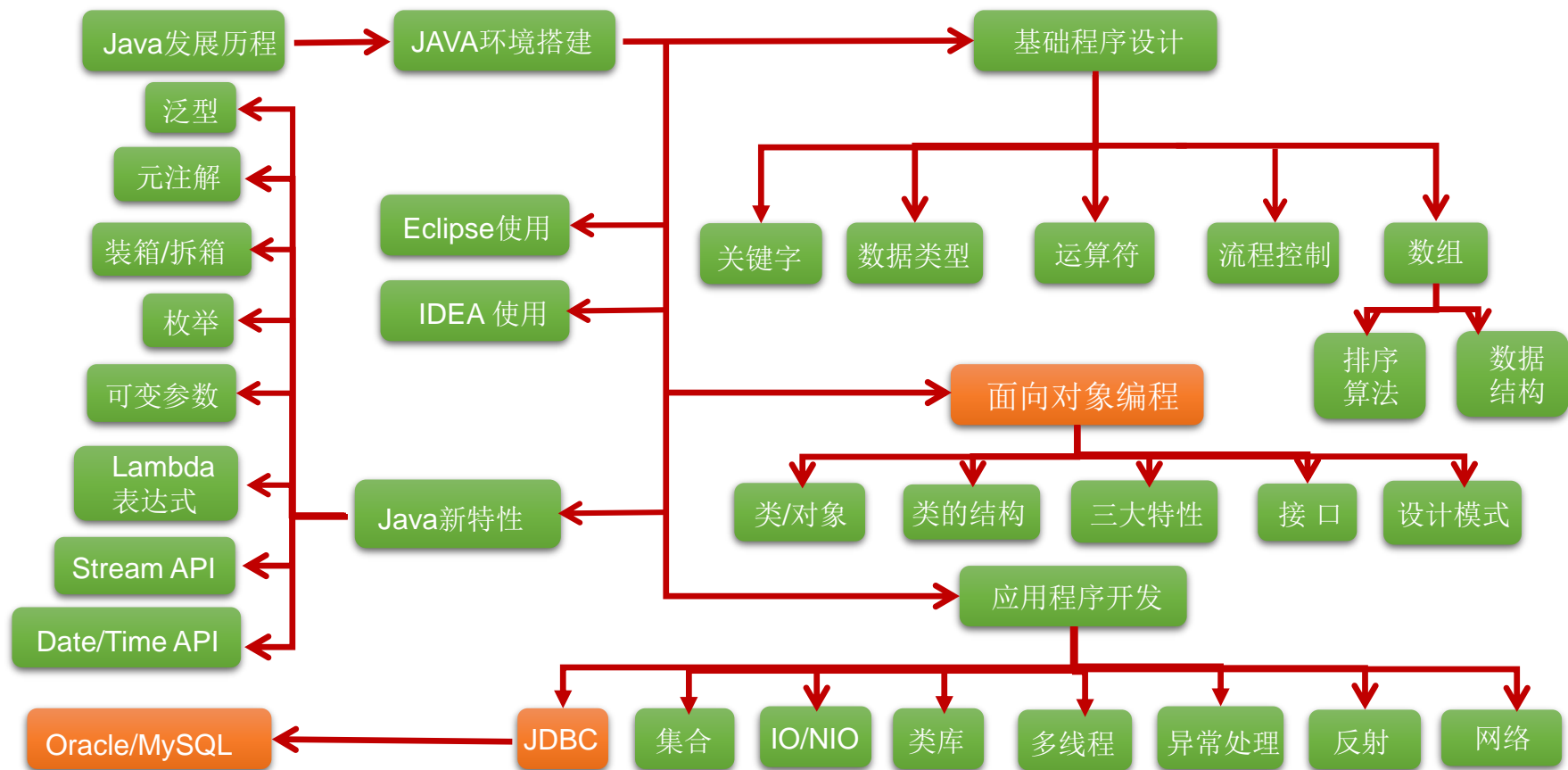




第2章

Java基本语法(下): 程序流程控制

讲师：宋红康
新浪微博：尚硅谷-宋红康





2-5 程序流程控制



- 流程控制语句是用来控制程序中各语句执行顺序的语句，可以把语句组合成能完成一定功能的小逻辑模块。
- 其流程控制方式采用结构化程序设计中规定的三种基本流程结构，即：
 - 顺序结构
 - 分支结构
 - 循环结构



塑料软包装生产流程



成交



设计



制版



印刷



复合



熟化



制袋



成品



质检



入库



出货





●顺序结构

➤程序从上到下逐行地执行，中间没有任何判断和跳转。

●分支结构

➤根据条件，选择性地执行某段代码。

➤有if...else和switch-case两种分支语句。

●循环结构

➤根据循环条件，重复性的执行某段代码。

➤有while、do...while、for三种循环语句。

➤注：JDK1.5提供了foreach循环，方便的遍历集合、数组元素。



2-5-1 顺序结构



2.5.1 顺序结构

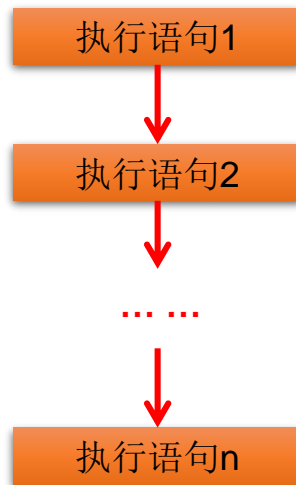
●顺序结构

Java中定义成员变量时采用合法的**前向引用**。如：

```
public class Test{  
    int num1 = 12;  
    int num2 = num1 + 2;  
}
```

错误形式：

```
public class Test{  
    int num2 = num1 + 2;  
    int num1 = 12;  
}
```



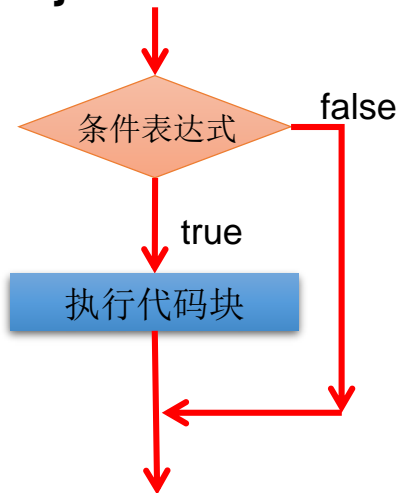


2-5-2 分支语句1: if-else结构

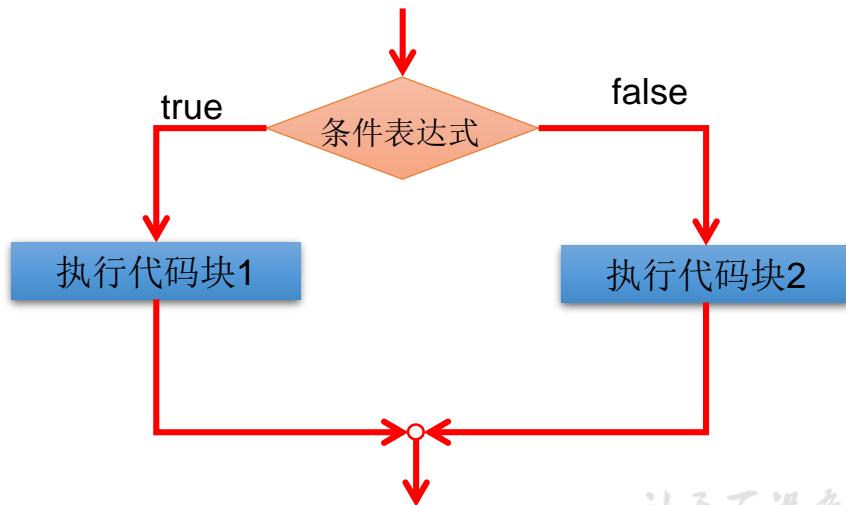


if语句三种格式：

1. if(条件表达式){
 执行代码块;
}



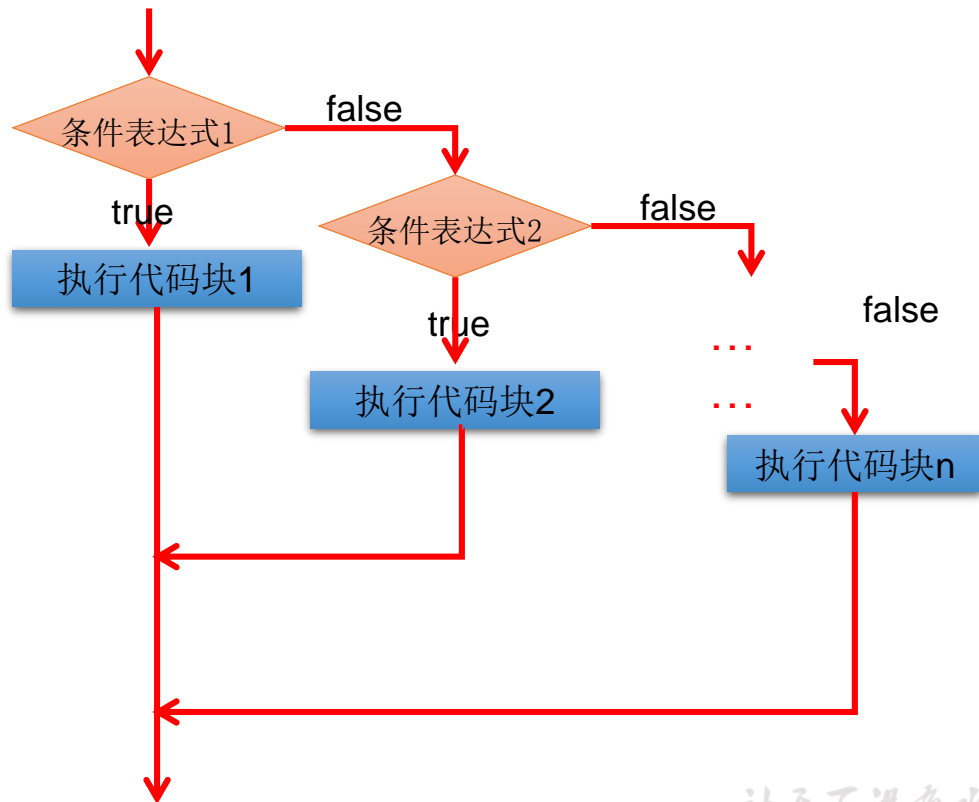
2. if(条件表达式){
 执行代码块1;
}
else{
 执行代码块2;
}





2.5.2 程序流程控制：if-else结构

```
3. if(条件表达式1){  
    执行代码块1;  
}  
else if (条件表达式2){  
    执行代码块2;  
}  
.....  
else{  
    执行代码块n;  
}
```





分支结构：if-else使用说明

- 条件表达式必须是布尔表达式（关系表达式或逻辑表达式）、布尔变量
- 语句块只有一条执行语句时，一对{}可以省略，但建议保留
- if-else语句结构，根据需要可以嵌套使用
- 当if-else结构是“多选一”时，最后的else是可选的，根据需要可以省略
- 当多个条件是“互斥”关系时，条件判断语句及执行语句间顺序无所谓
当多个条件是“包含”关系时，“小上大下 / 子上父下”



if-else语句应用举例

```
public class AgeTest{
    public static void main(String args[]){
        int age = 75;
        if (age< 0) {
            System.out.println("不可能！");
        } else if (age>250) {
            System.out.println("是个妖怪！");
        } else {
            System.out.println("人家芳龄 " + age + ",马马乎乎啦！");
        }
    }
}
```



if语句例题1

岳小鹏参加Java考试，他和父亲岳不群达成承诺：
如果：

成绩为100分时，奖励一辆BMW；

成绩为(80, 99]时，奖励一台iphone xs max；

当成绩为[60,80]时，奖励一个 iPad；

其它时，什么奖励也没有。

请从键盘输入岳小鹏的期末成绩，并加以判断



if语句例题2

- 编写程序：由键盘输入三个整数分别存入变量num1、num2、num3，对它们进行排序(使用 if-else if-else),并且从小到大输出。



if语句练习1

1)对下列代码，若有输出，指出输出结果。

```
int x = 4;
int y = 1;
if (x > 2) {
    if (y > 2)
        System.out.println(x + y);
    System.out.println("atguigu");
} else
    System.out.println("x is " + x);
```

2)

```
boolean b = true;
//如果写成if(b=false)能编译通过吗？如果能，结果是？
if(b == false)
    System.out.println("a");
else if(b)
    System.out.println("b");
else if(!b)
    System.out.println("c");
else
    System.out.println("d");
```




if语句练习2

- 1) 编写程序，声明2个int型变量并赋值。判断两数之和，如果大于等于50，打印“hello world!”
- 2) 编写程序，声明2个double型变量并赋值。判断第一个数大于10.0，且第2个数小于20.0，打印两数之和。否则，打印两数的乘积。
- 3) 我家的狗5岁了，5岁的狗相当于人类多大呢？其实，狗的前两年每一年相当于人类的10.5岁，之后每增加一年就增加四岁。那么5岁的狗相当于人类多少年龄呢？应该是： $10.5 + 10.5 + 4 + 4 + 4 = 33$ 岁。

编写一个程序，获取用户输入的狗的年龄，通过程序显示其相当于人类的年龄。如果用户输入负数，请显示一个提示信息。



if语句练习3

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`(int)(Math.random() * 90 + 10)`产生随机数。

`Math.random() : [0,1) * 90 → [0,90) + 10 → [10,100) → [10,99]`



if语句练习4

大家都知道，男大当婚，女大当嫁。那么女方家长要嫁女儿，当然要提出一定的条件：高：180cm以上；富：财富1千万以上；帅：是。

- 如果这三个条件同时满足，则：“我一定要嫁给他!!!”
- 如果三个条件有为真的情况，则：“嫁吧，比上不足，比下有余。”
- 如果三个条件都不满足，则：“不嫁！”

提示：

```
Sysout("身高: (cm))  
scanner.nextInt();  
Sysout("财富: (千万))  
scanner.nextDouble();  
Sysout("帅否: (true/false)) (是/否)  
scanner.nextBoolean(); scanner.next(); "是".equals(str)
```

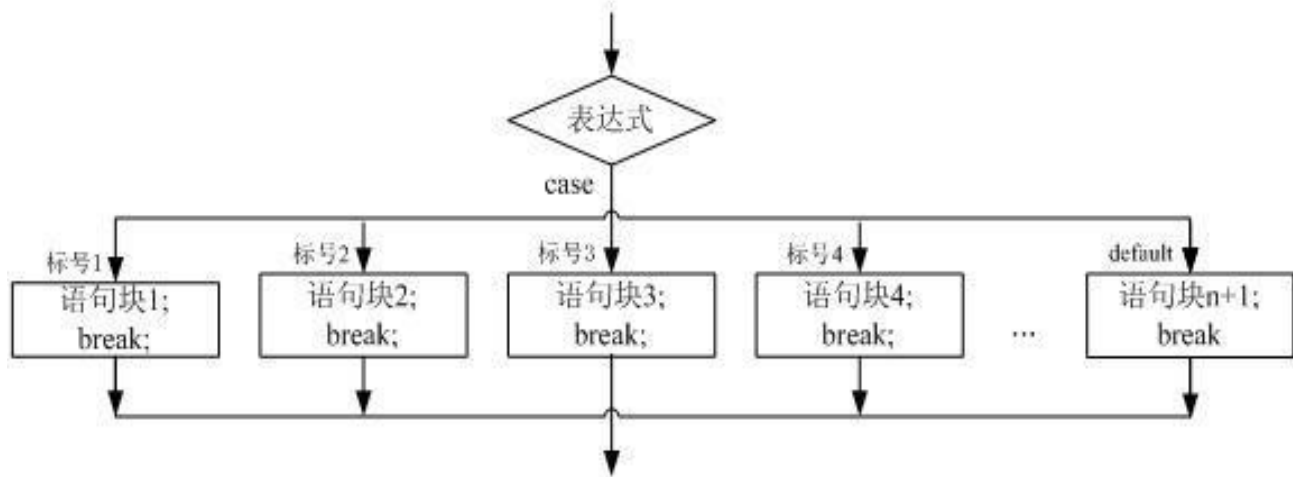


2-5-3 分支语句2: switch-case结构



2.5.3 程序流程控制：switch-case结构

```
switch(表达式){  
case 常量1:  
    语句1;  
    // break;  
case 常量2:  
    语句2;  
    // break;  
... ..  
case 常量N:  
    语句N;  
    // break;  
default:  
    语句;  
    // break;  
}
```





switch语句应用举例

```
public class SwitchTest {  
    public static void main(String args[]) {  
        int i = 1;  
        switch (i) {  
            case 0:  
                System.out.println("zero");  
                break;  
            case 1:  
                System.out.println("one");  
                break;  
            default:  
                System.out.println("default");  
                break;  
        }  
    }  
}
```



switch语句应用举例

```
String season = "summer";
switch (season) {
    case "spring":
        System.out.println("春暖花开");
        break;
    case "summer":
        System.out.println("夏日炎炎");
        break;
    case "autumn":
        System.out.println("秋高气爽");
        break;
    case "winter":
        System.out.println("冬雪皑皑");
        break;
    default:
        System.out.println("季节输入有误");
        break;
}
```



switch语句有关规则

- switch(表达式)中表达式的值必须是下述几种类型之一：**byte, short, char, int, 枚举 (jdk 5.0), String (jdk 7.0);**
- case子句中的值必须是**常量**，**不能是变量名或不确定的表达式值**；
像case > 18这种是不行的
- 同一个switch语句，所有case子句中的常量值互不相同；
- break语句用来在执行完一个case分支后使程序跳出switch语句块；如果没有break，程序会顺序执行到switch结尾
- default子句是**可任选的**。同时，位置也是灵活的。**当没有匹配的case时，执行default**

注意，default语句执行完后，如果没有break，依然会继续往下执行



例 题

1.使用 switch 把小写类型的 char型转为大写。只转换 a, b, c, d, e. 其它的输出 “other”。

提示：String word = scan.next(); char c = word.charAt(0); switch(c){}

2.对学生成绩大于60分的，输出“合格”。低于60分的，输出“不合格”。

3.根据用于指定月份，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

4. 编写程序：从键盘上输入2019年的“month”和“day”，要求通过程序输出输入的日期为2019年的第几天。



switch语句练习1

从键盘分别输入年、月、日，判断这一天是当年的第几天

注：判断一年是否是闰年的标准：

- 1) 可以被4整除，但不可被100整除
或
- 2) 可以被400整除



switch语句练习2

- 使用switch语句改写下列if语句：

```
int a = 3;  
int x = 100;  
if(a==1)  
    x+=5;  
else if(a==2)  
    x+=10;  
else if(a==3)  
    x+=16;  
else  
    x+=34;
```



switch语句练习3

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

方式一：使用if-else

方式二：使用switch-case: score / 10: 0 - 10



switch和if语句的对比

if和switch语句很像，具体什么场景下，应用哪个语句呢？

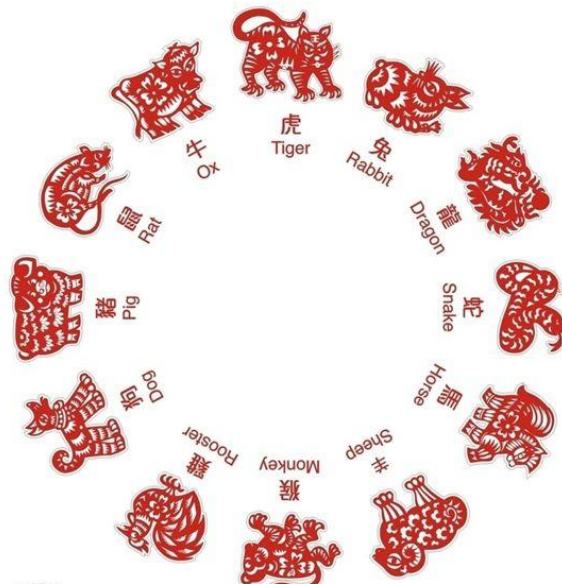
- 如果判断的具体数值不多，而且符合byte、short、char、int、String、枚举等几种类型。虽然两个语句都可以使用，建议使用switch语句。因为效率稍高。
- 其他情况：对区间判断，对结果为boolean类型判断，使用if，if的使用范围更广。也就是说，使用switch-case的，都可以改写为if-else。反之不成立。



switch语句练习4

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、ox、tiger、rabbit、dragon、snake、horse、sheep、monkey、rooster、dog、pig。

提示：2019年：猪 $2019 \% 12 == 3$





2-5-4 循环结构

- 循环结构

- 在某些条件满足的情况下，反复执行特定代码的功能

- 循环语句分类

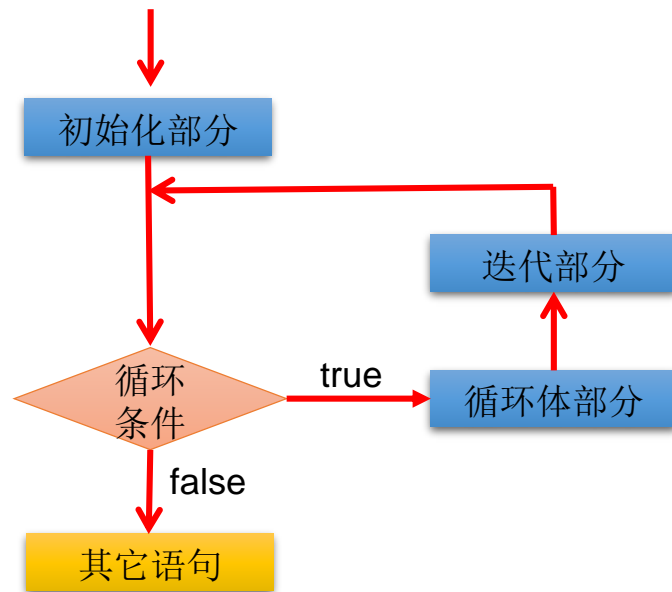
- for 循环

- while 循环

- do-while 循环

●循环语句的四个组成部分

- 初始化部分(`init_statement`)
- 循环条件部分(`test_exp`)
- 循环体部分(`body_statement`)
- 迭代部分(`alter_statement`)





2-5-4 循环结构1： for循环



- 语法格式

```
for (①初始化部分; ②循环条件部分; ④迭代部分) {  
    ③循环体部分;  
}
```

- 执行过程：

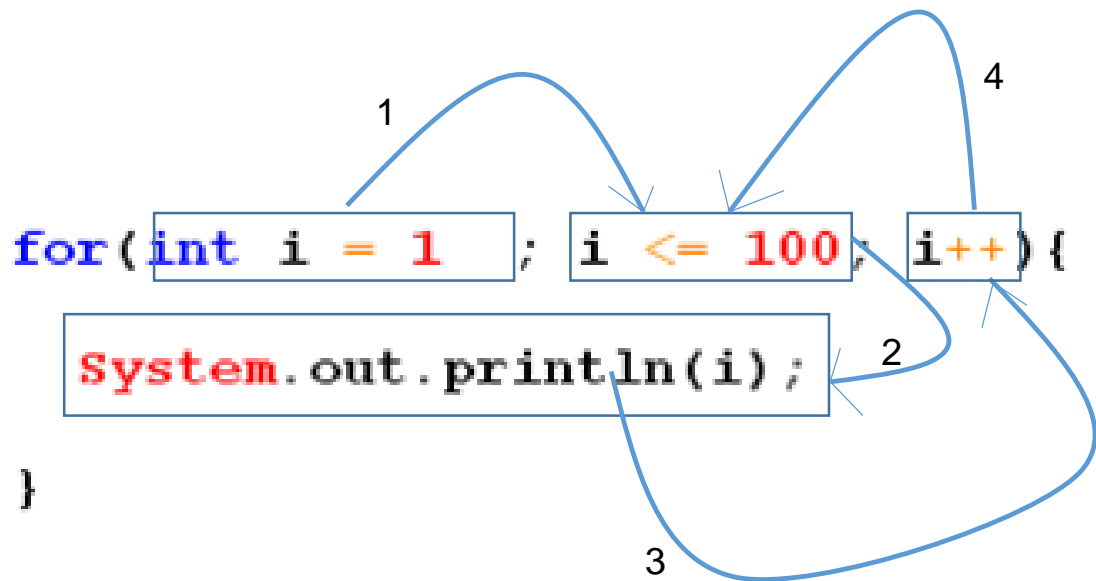
①-②-③-④-②-③-④-②-③-④-.....-②

- 说明：

- ②循环条件部分为boolean类型表达式，当值为false时，退出循环
- ①初始化部分可以声明多个变量，但必须是同一个类型，用逗号分隔
- ④可以有多个变量更新，用逗号分隔



for 循环执行演示





●应用举例

```
public class ForLoop {  
    public static void main(String args[]) {  
        int result = 0;  
        for (int i = 1; i <= 100; i++) {  
            result += i;  
        }  
        System.out.println("result=" + result);  
    }  
}
```



for语句例题1

编写程序从1循环到150，并在每行打印一个值，另外在每个3的倍数行上打印出“foo”，在每个5的倍数行上打印“biz”，在每个7的倍数行上打印输出“baz”。

```
1
2
3 foo
4
5 biz
6 foo
7 baz
8
9 foo
10 biz
11
12 foo
13
14 baz
15 foo biz
```

```
100 biz
101
102 foo
103
104
105 foo biz baz
106
107
108 foo
109
110 biz
111 foo
112 baz
113
114 foo
```



for语句例题2

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。

比如：12和20的最大公约数是4，最小公倍数是60。

说明：**break**关键字的使用



for语句练习

- 1.打印1~100之间所有奇数的和
- 2.打印1~100之间所有是7的倍数的整数的个数及总和（体会设置计数器的思想）
- 3.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： $153 = 1*1*1 + 3*3*3 + 5*5*5$



2-5-5 循环结构2: while循环



●语法格式

①初始化部分

while(②循环条件部分) {

③循环体部分;

④迭代部分;

}

●执行过程:

①-②-③-④-②-③-④-②-③-④-...-②

说明:

➤注意不要忘记声明④迭代部分。否则，循环将不能结束，变成死循环。

➤for循环和while循环可以相互转换



● 应用举例

```
public class WhileLoop {  
    public static void main(String args[]) {  
        int result = 0;  
        int i = 1;  
        while (i <= 100) {  
            result += i;  
            i++;  
        }  
        System.out.println("result=" + result);  
    }  
}
```



2-5-6 循环结构3： do-while循环



●语法格式

①初始化部分;

do{

③循环体部分

④迭代部分

}while(②循环条件部分);

●执行过程:

①-③-④-②-③-④-②-③-④-...②

●说明:

do-while循环至少执行一次循环体。



- 应用举例

```
public class DoWhileLoop {  
    public static void main(String args[]) {  
        int result = 0, i = 1;  
        do {  
            result += i;  
            i++;  
        } while (i <= 100);  
        System.out.println("result=" + result);  
    }  
}
```



循环语句综合例题

题目：

从键盘读入个数不确定的整数，并判断读入的正数和负数的个数，输入为0时结束程序。

最简单“无限”循环格式：**while(true)** , **for(;;)**,无限循环存在的原因是并不知道循环多少次，需要根据循环体内部某些条件，来控制循环的结束。

```
• class PositiveNegative {  
•     public static void main(String[] args) {  
•         Scanner scanner = new Scanner(System.in);  
•         int positiveNumber = 0; //统计正数的个数  
•         int negativeNumber = 0; //统计负数的个数  
•         for(;;){ //while(true){  
•             System.out.println("请输入一个整数: ");  
•             int z = scanner.nextInt();  
•             if(z>0)  
•                 positiveNumber++;  
•             else if(z<0)  
•                 negativeNumber++;  
•             else  
•                 break;  
•         }  
•         System.out.println("正数的个数为: "+ positiveNumber);  
•         System.out.println("负数的个数为: "+ negativeNumber); } }
```




2-5-7 嵌套循环



嵌套循环(多重循环)

- 将一个循环放在另一个循环体内，就形成了嵌套循环。其中，for ,while ,do...while均可以作为外层循环或内层循环。
- 实质上，嵌套循环就是把内层循环当成外层循环的循环体。当只有内层循环的循环条件为false时，才会完全跳出内层循环，才可结束外层的当次循环，开始下一轮的循环。
- 设外层循环次数为m次，内层为n次，则内层循环体实际上需要执行m*n次。

例题：1) 九九乘法表
2) 100以内的所有质数



2-5-8 特殊关键字的使用： break、continue



特殊流程控制语句1

●break 语句

- break语句用于终止某个语句块的执行

```
{  
    .....  
    break;  
    .....  
}
```

- break语句出现在多层嵌套的语句块中时，可以通过标签指明要终止的是哪一层语句块

```
label1: { .....  
label2:   { .....  
label3:   { .....  
           { .....  
             break label2;  
             .....  
           }  
         }  
       }  
}
```



特殊流程控制语句1

●break 语句用法举例

```
public class BreakTest{  
    public static void main(String args[]){  
        for(int i = 0; i<10; i++){  
            if(i==3)  
                break;  
            System.out.println(" i =" + i);  
        }  
        System.out.println("Game Over!");  
    }  
}
```



特殊流程控制语句2

●continue 语句

- continue只能使用在循环结构中
- continue语句用于跳过其所在循环语句块的一次执行，继续下一次循环
- continue语句出现在多层嵌套的循环语句体中时，可以通过标签指明要跳过的是哪一层循环

●continue语句用法举例

```
public class ContinueTest {  
    public static void main(String args[]){  
        for (int i = 0; i < 100; i++) {  
            if (i%10==0)  
                continue;  
            System.out.println(i);  
        }  
    }  
}
```



附加：特殊流程控制语句3

- **return**：并非专门用于结束循环的，它的功能是结束一个方法。当一个方法执行到一个**return**语句时，这个方法将被结束。
- 与**break**和**continue**不同的是，**return**直接结束整个方法，不管这个**return**处于多少层循环之内



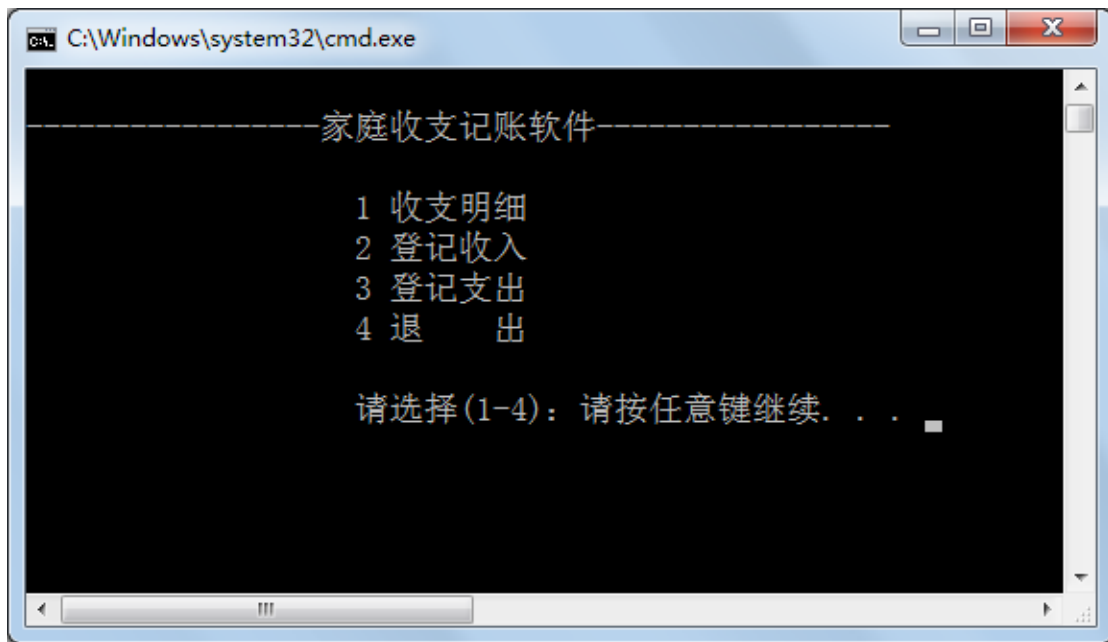
特殊流程控制语句说明

- break只能用于**switch语句**和**循环语句**中。
- continue 只能用于**循环语句**中。
- 二者功能类似，但continue是终止**本次**循环，break是终止**本层**循环。
- break、continue之后不能有其他的语句，因为程序永远不会执行其后的语句。
- 标号语句必须紧接在循环的头部。标号语句不能用在非循环语句的前面。
- 很多语言都有goto语句，goto语句可以随意将控制转移到程序中的任意一条语句上，然后执行它。但使程序容易出错。Java中的break和continue是不同于goto的。



综合练习

项目一：家庭收支记账软件



让天下没有难学的技术



尚硅谷