

语言基础第三天：

笔记：

1. 运算符：

- 算术：+、-、*、/、%、++、--
 - %：取模/取余，余数为0即为整除

```
System.out.println(5 % 2); //1, 商2余1
System.out.println(8 % 2); //0, 商4余0----整除
System.out.println(2 % 8); //2, 商0余2
```

- ++/--：自增1/自减1，可在变量前也可在变量后
 - 单独使用时，在前在后都一样

```
//++单独使用：
int a = 5, b = 5;
a++; //相当于a=a+1
++b; //相当于b=b+1
System.out.println(a); //6
System.out.println(b); //6

//--单独使用：
int a = 5, b = 5;
a--; //相当于a=a-1
--b; //相当于b=b-1
System.out.println(a); //4
System.out.println(b); //4
```

- 被使用时，在前在后不一样
 - 区别：先++，还是先使用a

```
//++被使用：
int a = 5, b = 5;
int c = a++; //1)记录下a++的值5 2)a自增1变为6 3)将记录的5赋值给c
int d = ++b; //1)记录下++b的值6 2)b自增1变为6 3)将记录的6赋值给d
System.out.println(a); //6
System.out.println(b); //6
System.out.println(c); //5
System.out.println(d); //6

//--被使用：
int a = 5, b = 5;
int c = a--; //a--的值为5，所以c就是5
int d = --b; //--b的值为4，所以d就是4
System.out.println(a); //4
System.out.println(b); //4
System.out.println(c); //5
System.out.println(d); //4
```

o 关系: >、<、>=、<=、==、!=

- 大于>, 小于<, 大于或等于>=, 小于或等于<=, 等于==, 不等于!=
- 关系运算的结果为boolean型, 关系成立则为true, 关系不成立则为false

```
int a = 5, b = 10, c = 5;
boolean b1 = a > b;
System.out.println(b1); //false
System.out.println(c < b); //true
System.out.println(b >= a); //true
System.out.println(a <= c); //true
System.out.println(a == c); //true
System.out.println(a != c); //false
System.out.println(a + c > 10); //false
System.out.println(a % 2 == 0); //false

System.out.println(a++ > 5); //false-----a自增1变为6
System.out.println(a++ > 5); //true-----a自增1变为7
```

o 逻辑: &&、||、!

- &&: 短路与(并且), 见false则false, 两边都为true才为true

若第1个条件为false, 则发生短路(后面的不执行了)

```
int a = 5, b = 10, c = 5;
boolean b1 = b >= a && b < c;
System.out.println(b1); //true&&false=false
System.out.println(b <= c && b > a); //false&&true=false
System.out.println(a == b && c > b); //false&&false=false
System.out.println(b != c && a < b); //true&&true=true

//演示短路:
boolean b3 = a > b && c++ > 2;
System.out.println(b3); //false
System.out.println(c); //5, 发生短路了(c++>2并没有被执行)
```

- ||: 短路或(或者), 见true则true, 两边都为false才为false

若第1个条件为true, 则发生短路(后面的不执行了)

```
System.out.println(b >= a || b < c); //true||false=true
System.out.println(b <= c || b > a); //false||true=true
System.out.println(b != c || a < b); //true||true=true
System.out.println(a == b || c > b); //false||false=false

//演示短路:
boolean b4 = a < b || c++ > 2;
System.out.println(b4); //true
System.out.println(c); //5, 发生短路了(c++>2并没有被执行)
```

- !: 逻辑非(取反)

```
boolean b2 = !(a > b); //!的优先级高, 所以需要将a>b括起来
System.out.println(b2); //!false=true
System.out.println(!(a < b)); //!true=false
```

- 逻辑运算建立在关系运算的基础上，所以逻辑运算的结果也是boolean型
- 赋值：=、+=、-=、*=、/=、%=
 - 简单赋值运算符:=
 - 扩展赋值运算符:+=、-=、*=、/=、%=
 - 扩展赋值自带强转功能

```
int a = 5;
a += 10; //相当于a=(int)(a+10)
System.out.println(a); //15
a *= 2; //相当于a=(int)(a*2)
System.out.println(a); //30
a /= 6; //相当于a=(int)(a/6)
System.out.println(a); //5

//小面试题:
short s = 5;
// s = s + 10; //编译错误, 需强转, 改为: s = (short) (s + 10);
s += 10; //相当于: s=(short)(s+10);
```

- 字符串连接:

- +:
- 若两边为数字，则做加法运算
- 若两边出现了字符串，则做字符串连接

任何类型的数据与字符串相连，结果都会变为字符串类型

```
int age = 38;
System.out.println("age="); //age=
System.out.println(age);    //38
System.out.println("age=" + age); //age=38
System.out.println("我的年龄是" + age); //我的年龄是38
System.out.println("我今年" + age + "岁了"); //我今年38岁了

String name = "WKJ"; //String是字符串数据类型
System.out.println("大家好, 我叫" + name); //大家好, 我叫WKJ
System.out.println("大家好, 我叫" + name + ", 今年" + age + "岁了"); //大家好, 我叫WKJ, 今年38岁了

System.out.println(10 + 20 + "" + 30); //3030-----String
System.out.println("" + 10 + 20 + 30); //102030----String
System.out.println(10 + 20 + 30 + ""); //60-----String
```

- 条件/三目: boolean?数1:数2

- 执行过程:
 - 整个表达式是有值的，它的值要么是?号后的数1，要么是:号后的数2
 - 计算boolean的值:
 - 若为true，则整个表达式的值为?号后的数1
 - 若为false，则整个表达式的值为:号后的数2

```
int num = 5;
int flag = num > 0 ? 1 : -1;
System.out.println(flag); //1

int a = 8, b = 55;
int max = a > b ? a : b;
System.out.println("max=" + max);
```

2. 分支结构：基于条件执行某语句

o if结构：1条路

1) 语法：

```
if(boolean){
    语句块-----有条件执行的语句
}
```

2) 执行过程：

判断boolean的值：

若为true，则执行语句块(if整个结束)

若为false，则if直接结束

```
//1) 偶数的判断：
int num = 6; //带数(6,5)
if (num % 2 == 0) {
    System.out.println(num + "是偶数");
}
System.out.println("继续执行...");

//2) 满500打8折：
double price = 300.0; //消费金额 带数(600.0,300.0)
if (price >= 500) { //满500
    price = price * 0.8; //打8折
}
System.out.println("最终结算金额为:" + price);
```

o if...else结构：2条路

1) 语法：

```
if(boolean){
    语句块1
}else{
    语句块2
}
```

2) 执行过程：

判断boolean的值：

若为true，则执行语句块1(整个结束)

若为false，则执行语句块2(整个结束)

3) 说明：

语句块1和语句块2，必走其中之一-----2选1

```
//1) 偶数、奇数的判断：
int num = 5; //带数(6,5)
if (num % 2 == 0) {
    System.out.println(num + "是偶数");
} else {
```

```

        System.out.println(num + "是奇数");
    }
    System.out.println("继续执行...");

    //2) 满500打8折, 不满500打9折:
    double price = 300.0;
    if (price >= 500) { //带数(600.0,300.0)
        price *= 0.8;
    } else {
        price *= 0.9;
    }
    System.out.println("最终结算金额为:" + price);

```

补充:

1. 任何复杂的程序逻辑都可以通过三种结构来实现:
 - 顺序结构: 从上往下逐行执行, 每句必走
 - 分支结构: 有条件的执行某语句, 并非每句必走
 - 循环结构: -----下次课讲
2. 明日单词:

```

1)Scanner/scan: 扫描仪
2)import: 引入、导入
3)System: 系统
4)in: 进入
5)new: 新的
6)nextInt: 下一个整数
7)nextDouble: 下一个浮点数
8)switch: 开关
9)case: 案例
10)break: 中断、退出
11)command: 命令
12)by: 通过
13)times: 次数
14)while: 当...的时候, 循环的一种
15)do: 做、干
16)math: 数学
17)random: 随机
18)guess/guessing: 猜
19)game: 游戏
20)level: 等级
21)count: 数量

```

扩展练习:

1. 声明两个整型变量a和b并分别赋值, 找到a和b中的最大值, 并输出。用两种方式实现: 条件运算符、if...else分支结构。

```

public class extra_1 {
    public static void main(String[] args) {
        int a = 1, b = 2;
        int max1 = a > b ? a : b;
        System.out.println(max1);

        int c = 3, d = 4;
    }
}

```

```

        int max2;
        if (c > d) {
            max2 = c;
        } else {
            max2 = d;
        }
        System.out.println(max2);
    }
}

```

2. 声明一个整型变量year并存储年份，判断其是平年还是闰年，若是闰年则输出"某某某年是闰年"，否则输出"某某某年是平年"。

```

public class extra_2 {
    public static void main(String[] args) {
        //闰年：能被4整除且不能被100整除，或能被400整除
        int year = 1999; //代入1999, 2020, 2000
        if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
            System.out.println(year + "是闰年");
        } else {
            System.out.println(year + "是平年");
        }
    }
}

```