## Day03

### 3-1 类型转换6条规则

* 基本数据类型中除了boolean类型以外，剩余7种类型可以进行相互转换
* 当取值范围小的类型赋给取值范围大的类型，构成自动类型转换也叫做隐式类型转换
  + byte short char int long float double
* 当取值范围大的类型不能直接给取值范围小的类型赋值，则必须进行强制类型转换也叫做显式类型转换，但是可能会存在精度丢失
* 当对byte short char类型赋值时，如果没有超出该类型取值范围则可以直接赋值
* 当对byte short char类型进行混合运算时，则先转为int类型然后再进行计算
* 当多种数据类型进行混合运算时，则选转为取值范围大的类型，然后再进行计算

### 3-2 算术运算符

* + - \* / %(求余数，取模) ++ --
* +
  + 当+两边的操作数都是数值类型，则做加法运算
  + 当+两边的操作数中有一个为非数值类型，则做连接,最终连接后的结果为String
* -
  + 当做减法运算时，必须保证两个操作数都是数值类型
* \*
* /
  + 除数不能为0
* %
* ++
  + ++表示自身加1
  + ++又分为：前置++、后置++
    - 前置++：将++编写在变量名称前面，先自身加1，然后再做其他运算

int a = 3;

int b = ++a; //a = 4 b = 4

* + - 后置++：将++编写在变量名称后面，先做其他运算，然后再自身加1

int x = 5;

int y = x++; //y = 5 x = 6

* --
  + --自身减1
  + --又分为前置、后置

### 3-3 连接运算符

* +
  + 作用：求和、连接

### 3-4 赋值运算符

* =
  + 将右侧的值赋给左侧的变量名称
* 扩展赋值运算符
  + += -= \*= /= %=
  + 当是扩展赋值运算符时，变量最终的数据类型没有发生改变

int x = 10;

//x += 0.5; //结果x = 10

//x = x + 0.5; //出现编译错误

### 3-5 关系运算符

* > < >= <= == !=
  + 关系运算符：最终结果为boolean类型的值
  + 优先级别：算术运算符 > 关系运算符 > 赋值运算符

### 3-6 逻辑运算符

* 逻辑与& 逻辑或| 逻辑异或^ 逻辑非!

短路与&& 短路或||

总结：逻辑运算符:最终结果为boolean类型的值

* 逻辑与&，表示并且
  + 当两个条件同时为true时，则结果为true,否则结果为false
* 逻辑或|，表示或者
  + 当两个条件中有一个为true时，则结果为true,否则为false
* 逻辑异或
  + 当两个条件的值不同时，则结果为true,否则为false

true ^ true ------- > false

true ^ false------- > true

fasle ^ false ------ > false

false ^ true ------ > true

* 逻辑非!
  + 表示对boolean类型的值进行取反
* 短路与&&，类似于逻辑与，都表示并且
  + 短路与、逻辑与运行结果都相同，但是执行过程可能不同，当使用短路与，并且第一个条件为false时，则结果直接为false
  + 建议使用短路与
* 短路或，类似于逻辑或，都表示或者
  + 短路或、逻辑或运行结果都相同，但是执行过程可能不同，当使用短路或，并且第一个条件为true时，则结果直接为true
  + 建议使用短路或

### 3-7 =与==、&与&&、|与||区别

* =与==
  + =属于赋值运算符，将右侧的值赋给左侧的变量名称
  + ==属于关系运算符，判断左右两边的值是否相等，结果为boolean类型
* &与&&
  + &是逻辑与， &&是短路与，都属于逻辑运算符，都表示并且，执行结果都相同
  + 当使用短路与，并且第一个条件为false时，则结果直接为false
* |与||
  + |是逻辑或，||是短路或，都属于逻辑运算符，都表示或者，结果都相同
  + 当使用短路或，并且第一个条件为true时，则结果直接为true

### 3-8 三目运算符

也叫做条件运算符

语法格式： 条件 ? 代码1 : 代码2

### 3-9 键盘接收用户输入

第一步：导入包，必须位于Java源文件中的第一行

import java.util.Scanner;

第二步：创建Scanner对象

Scanner input = new Scanner(System.in);

第三步：友好提示

System.out.print("请输入：");

第四步：键盘输入

int num = input.nextInt(); //注意：等待用户接收键盘输入值，当按回车键，则表示输入完毕，并将回车之前所输入的内容赋给左侧的变量

double score = input.nextDouble();

String name = input.next(); //注意：在Java中，键盘接收字符串类型时没有nextString(),而使用的是next()

boolean isFlag = input.nextBoolean();

String s = input.next(); //注意：在Java中，键盘不能直接接收字符类型，而是先接收字符串类型,然后在字符串中获取第0个位置中的值，则使用charAt(0)

char sex = s.charAt(0);