1.4 字符串的“+”操作

当“+”操作中出现字符串时，这个”+”是**字符串连接符**，而不是算术运算。

- **“itheima” + 666**

在”+”操作中，如果出现了字符串，就是连接运算符，否则就是算术运算。当连续进行“+”操作时，从左到右逐个执行。

- **1 + 99 + "年黑马"**

2.1 赋值运算符

符号	作用	说明
=	赋值	a=10，将10赋值给变量a
+=	加后赋值	a+=b，将a+b的值给a
-=	减后赋值	a-=b，将a-b的值给a
*=	乘后赋值	a*=b，将a×b的值给a
/=	除后赋值	a/=b，将a÷b的商给a
%=	取余后赋值	a%=b，将a÷b的余数给a

注意事项：

扩展的赋值运算符**隐含**了强制类型转换。

3.1 自增自减运算符

符号	作用	说明
++	自增	变量的值加1
--	自减	变量的值减1

注意事项：

- ++和--既可以放在变量的后边，也可以放在变量的前边。
- 单独使用的时候，++和--无论是放在变量的前边还是后边，结果是一样的。
- 参与操作的时候，如果放在变量的后边，先拿变量参与操作，后拿变量做++或者--。
  参与操作的时候，如果放在变量的前边，先拿变量做++或者--，后拿变量参与操作。

最常见的用法“++”“--”都是单独使用的

4.1 关系运算符

符号	说明
==	a==b，判断a和b的值是否相等，成立为true，不成立为false
!=	a!=b，判断a和b的值是否不相等，成立为true，不成立为false
>	a>b，判断a是否大于b，成立为true，不成立为false
>=	a>=b，判断a是否大于等于b，成立为true，不成立为false
<	a<b，判断a是否小于b，成立为true，不成立为false
<=	a<=b，判断a是否小于等于b，成立为true，不成立为false

注意事项：

关系运算符的**结果**都是**boolean**类型，要么是true，要么是false。

千万不要把“==”误写成“=”。

5.2 逻辑运算符

符号	作用	说明
&	逻辑与	a&b，a和b都是true，结果为true，否则为false
|	逻辑或	a|b，a和b都是false，结果为false，否则为true
^	逻辑异或	a^b，a和b结果不同为true，相同为false
!	逻辑非	!a，结果和a的结果正好相反

```
//& 有false则false
System.out.println((i > j) & (i > k)); //false & false
System.out.println((i < j) & (i > k)); //true & false
System.out.println((i > j) & (i < k)); //false & true
System.out.println((i < j) & (i < k)); //true & true
System.out.println("--------");

//| 有true则true
System.out.println((i > j) | (i > k)); //false | false
System.out.println((i < j) | (i > k)); //true | false
System.out.println((i > j) | (i < k)); //false | true
System.out.println((i < j) | (i < k)); //true | true
System.out.println("--------");

//^ 相同为false，不同为true
System.out.println((i > j) ^ (i > k)); //false ^ false
System.out.println((i < j) ^ (i > k)); //true ^ false
System.out.println((i > j) ^ (i < k)); //false ^ true
System.out.println((i < j) ^ (i < k)); //true ^ true
System.out.println("--------");

//!
System.out.println((i > j)); //false
System.out.println(!(i > j)); //!false
System.out.println(!!(i > j)); //!!false
System.out.println(!!!(i > j)); //!!!false
```

非“!”的表达式与其他的不同

5.3 短路逻辑运算符

符号	作用	说明
&&	短路与	作用和&相同，但是有短路效果
||	短路或	作用和|相同，但是有短路效果

注意事项：

- 逻辑与**&**，无论左边真假，右边都要执行。
  短路与**&&**，如果左边为真，右边执行；如果**左边为假，右边不执行。**
- 逻辑或**|**，无论左边真假，右边都要执行。
  短路或**||**，如果左边为假，右边执行；如果**左边为真，右边不执行。**

最常用的逻辑运算符：**&&，||，！**

```
//&&和&
//System.out.println((i++ > 100) & (j++ > 100)); //false & false
System.out.println((i++ > 100) && (j++ > 100)); //false && false
System.out.println("i:" + i);
System.out.println("j:" + j);
```

```
false
i:11
j:20

E:\>
```

1. 数据输入

1.2 Scanner使用的基本步骤

① 导包

```
import java.util.Scanner;
```

导包的动作必须出现在类定义的上边

② 创建对象

```
Scanner sc = new Scanner(System.in);
```

上面这个格式里面，只有sc是变量名，可以变，其他的都不允许变。

③ 接收数据

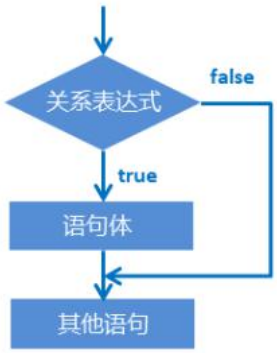
```
int i = sc.nextInt();
```

上面这个格式里面，只有i是变量名，可以变，其他的都不允许变。

2.1 if语句格式1

格式：

```
if (关系表达式) {
    语句体;
}
```



执行流程：

① 首先计算关系表达式的值

② 如果关系表达式的值为true就执行语句体

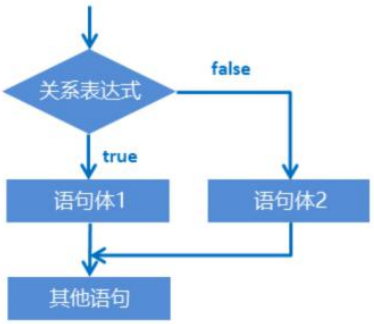
③ 如果关系表达式的值为false就不执行语句体

④ 继续执行后面的语句内容

2.2 if语句格式2

格式：

```
if (关系表达式) {
    语句体1;
} else {
    语句体2;
}
```



执行流程：

① 首先计算关系表达式的值

② 如果关系表达式的值为true就执行语句体1

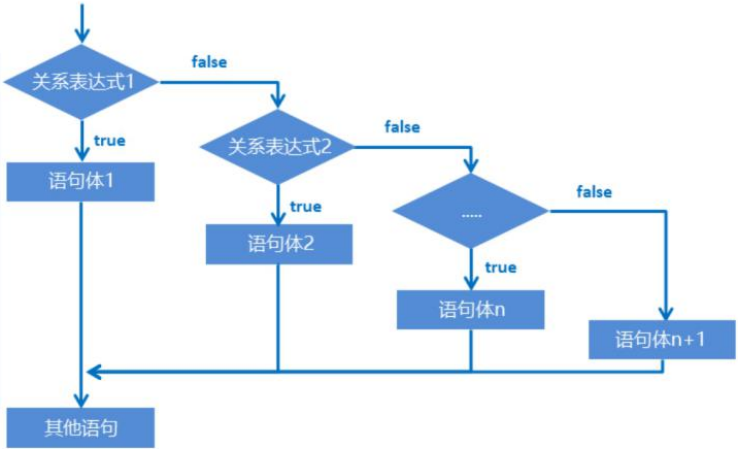
③ 如果关系表达式的值为false就执行语句体2

④ 继续执行后面的语句内容

2.3 if语句格式3

格式：

```
if (关系表达式1) {
    语句体1;
} else if (关系表达式2) {
    语句体2;
}
...
else {
    语句体n+1;
}
```



测试数据的时候一定要测试 正确数据、边界数据、错误数据，我们的程序才是健全的程序