

数据类型的转换：

//数据类型的转换

1, 隐式转换(自动转换):把所占字节少 的数据所占字节多的数据类型. 是安全的转换, 由系统自动执行

```
byte a1 = 10;
```

```
int a2 = a1;
```

//2. 显示转换(强制转换)把所占字节多 的数据所占字节少的数据类型. 是不安全的转换, 由由开发人

```
int d1 = 13456;
```

```
char d2 = (char) d1;
```

```
System.out.println("d2 = " + d2);
```

逻辑运算符

<=,>=,!=,

逻辑与算法符

//逻辑运算符

//逻辑与:&&并且

//规则:两侧同时为真才为真. 否则为假

```
int i = 3, j = 5, k = 7;
```

```
result = i < j && j == k;
```

```
System.out.println("result = " + result)
```

//逻辑或||;或者

```
result = j != k || j < k;
```

```
System.out.println("result = " + result)
```

//逻辑非:!

//规则:真变假, 假变真

```
result = !(i != j);
```

```
System.out.println("result = " + result)
```

//逻辑运算符的短路现象

//1. 逻辑与的短路:左侧为假, 右侧不参与运算

```
//逻辑运算符的短路现象
//1. 逻辑与的短路:左侧为假,右侧不参与运算
int c = 1;
result = i > j && ((c = 2) == 1);
System.out.println("result = " + result);
System.out.println("c = " + c);
//2. 逻辑或的短路:左侧为真,右侧不参与运算
```

```
result = i > j || ((c = 2) == 1);
System.out.println("result = " + result);
System.out.println("c = " + c);
```

程序的三大结构

```
//程序的三大结构
//1. 顺序结构:从main方法开始,从上到下,从左到右,依次执行
//2. 分支结构:程序执行到某个位置.进行条件判断,根据不同的结果,执行不同的语句
//3. 循环结构:程序反复执行某些代码,直到条件不满足
```

```
//运用:
//分支结构
//1. if语句
//2. switch
//3. 条件运算符
```

```
//if语句的第一种语句
/*if(条件表达式){
语句
}
```

```
*/
//执行顺序:先判断条件表达式,的结果. true, 执行, false, 跳过.
```

//if语句的第二种形式:

```
if(条件表达式){  
    语句一;  
}else{  
    语句二;  
}
```

//执行顺序,先计算条件表达式的结果,如果为 *true*,执行语句1, *false*,执行语句2

```
Scanner scanner = new Scanner(System.in);  
System.out.println("输入性别:");  
char gender = scanner.next().charAt(0);  
if (gender == 'f') {  
    System.out.println("女");  
}  
else {  
    System.out.println("男");  
}
```

条件运算符

//条件运算符

//条件表达式?表达式1:表达式2

//执行顺序:判断条件表达式的结果,如果为 *true*,返回表达式1的值; *false*,返回表达式2的值

```
Scanner scanner=new Scanner(System.in);  
System.out.println("输入两个数");  
int p = scanner.nextInt();  
int t = scanner.nextInt();  
int result2=p>t?p:t;  
System.out.println("result = " + result);
```