



北京大学
PEKING UNIVERSITY

信息科学技术学院

程序设计与算法(一)

李文新 郭炜



关系运算符、逻辑运算符和逻辑表达式

关系运算符

- 六种关系运算符用于数值的比较

相等 ==

不等 !=

大于 >

小于 <

大于等于 >=

小于等于 <=

- 比较的结果是bool类型，成立则为true, 反之为false

bool类型

- bool 类型变量只有两种取值，true或false

- false等价于0， true等价于非0整型值

```
int n = true, m = false;  
printf("%d,%d", n, m);    => 1,0
```

关系运算符

```
int main()
{
    int n1 = 4,  n2 = 5,  n3;
    n3 = ( n1 > n2 );    // n3 的值变为 0
    cout << n3 << ", "; // 输出 0,
    n3 = ( n1 < n2 );    // n3 的值变为非0值
    cout << n3 << ", "; // 输出 1,
    n3 = (n1 == 4);      // n3 的值变为非 0 值
    cout << n3 << ", "; // 输出 1,
    n3 = (n1 != 4);      // n3 的值变为0
    cout << n3 << ", "; // 输出 0,
    n3 = (n1 == 5);      // n3 的值变为0
    cout << n3 ;         // 输出 0,
    return 0;
}
```

逻辑运算符和逻辑表达式

逻辑运算符用于表达式的逻辑操作，有 **&&**, **||**, **!** 三种，操作的结果是**true**或**false**

● 与 **&&**

`exp1 && exp2` 当且仅当`exp1`和`exp2`的值都为真（或非0）时，结果为**true**

```
int n = 4;  
n > 4 && n < 5      => false  
n >= 2 && n < 5     => true  
5 && 0              => false  
4 && 1              => true
```

逻辑运算符和逻辑表达式

●或

||

exp1 || exp2 当且仅当exp1和exp2的值都为假（或0）时，结果为false

```
int n = 4;
```

```
n > 4 || n < 5    => true
```

```
n <= 2 || n > 5   => false
```

逻辑运算符和逻辑表达式

●非 !

! exp exp值为真(或非0), 结果为false, exp值为false(0), 结果为true

! (4 < 5) => false

!5 => false

!0 => true

逻辑运算符和逻辑表达式

逻辑表达式是**短路计算**的，即对逻辑表达式的计算，在整个表达式的值已经能够断定的时候即会停止

- `exp1 && exp2` : 如果已经算出表达式`exp1`为假，那么整个表达式的值肯定为假，于是表达式`exp2`就不需要再计算
- `exp1 || exp2` : 如果已经算出`exp1`为真，那么整个表达式必定为真，于是`exp2`也不必计算

逻辑运算符和逻辑表达式

```
#include <iostream>
using namespace std;
int main()
{
    int a = 0, b = 1;
    bool n = (a++) && (b++) ;    // b++不被计算
    cout << a << ", " << b << endl; //输出 1,1
    n = a++ && b++ ;    // a++和b++都要计算
    cout << a << ", " << b << endl; //输出 2,2
    n = a++ || b++ ;    //b++不被计算
    cout << a << ", " << b << endl; //输出 3,2
    return 0;
}
```