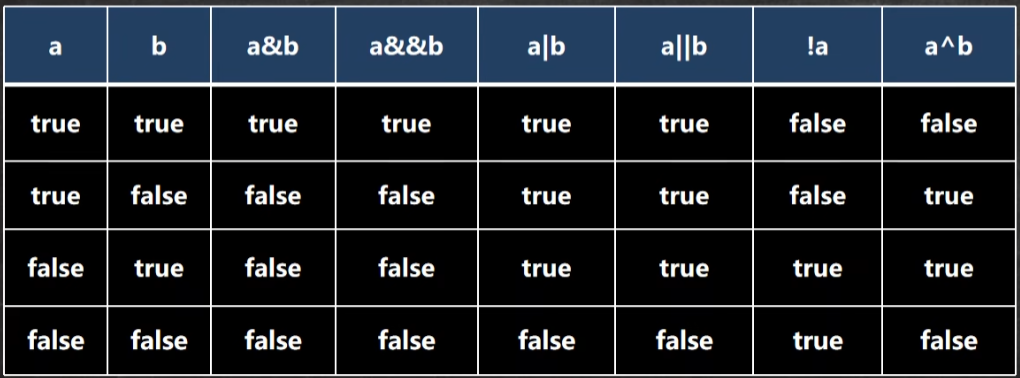
7.13.2024书：java核心技术卷一

今日： 今天学习逻辑（Logic）运算符 短路逻辑

逻辑运算符

用于连接多个条件（多个关系表达式），最终的结果也是一个boolean值



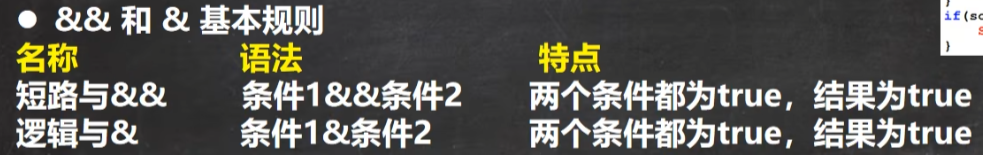
分为两组学习

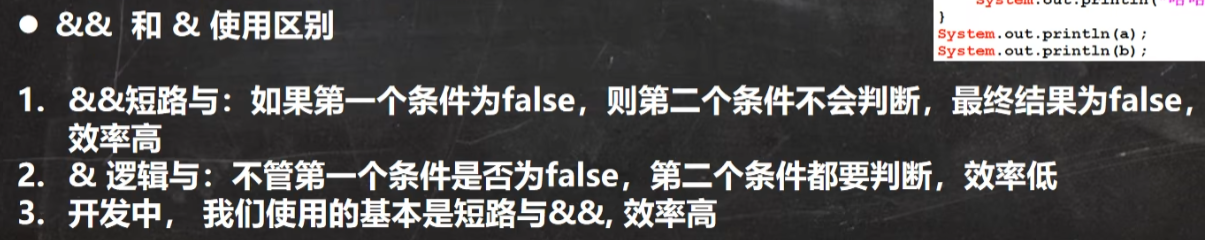
1. 短路与：&& ， 短路或: || ，取反: ！
2. 逻辑与：& ， 逻辑或: | ， ^： 逻辑异或

说明逻辑运算规则

1. a&b ：&叫逻辑与。规则： 当a和b同时为ture，则结果为ture，否则为false
2. a&&b ：&&叫短路与。规则：当a和b同时为ture，则结果为ture，否则为false
3. a|b ：|叫逻辑或。规则：当a和b，有一个为ture，则结果为ture，否则为false
4. a||b ：||叫短路或。规则：当a和b，有一个为ture，则结果为ture，否则为flase
5. !a ：!叫取反（非运算）。规则：当a为ture，则结果为false。当a为false，则结果为ture。
6. a^b ：^叫做逻辑异或。规则：当a和b不同时，结构为ture，否则为flase

（1）&& 和 & 的区别（ExerCise1）







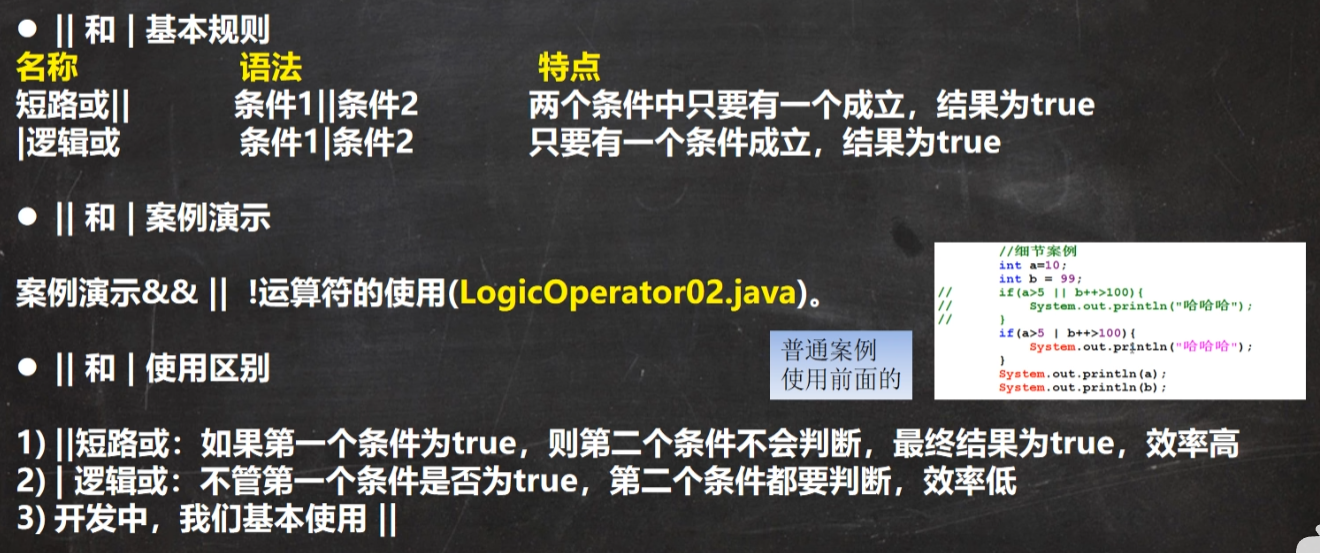
Logic（逻辑）Operator(运算符)

//演示逻辑运算符的使用

public class ExerCise1 {  
 //演示逻辑运算符的使用  
 public static void main(String[]args){  
 //&&短路与 和 &逻辑与 的演示  
 //&&的使用  
 int age = 20;  
 if(age < 30 && age > 10){  
 System.*out*.println("ok");  
 }  
 //&的使用  
 int age1 = 20;  
 if(age1 < 30 & age1 > 15){  
 System.*out*.println("ko");  
 }  
 //&和&&的区别  
 int a = 4;  
 int b = 5;  
 //对于&&短路与而言 如果第一个条件为false 后面的条件不在判断  
 if (a < 1 && ++b < 50){  
 System.*out*.println("ok666");//false未输出  
 }  
 //对于&逻辑与而言 如果第一个条件为false 后面的条件仍然会判断  
 System.*out*.println("a=" + a + "b=" + b);//4 5  
 if (a < 1 & ++b < 50){  
 System.*out*.println("ko666");//false未输出  
 }  
 System.*out*.println("a=" + a +"b=" + b);//4 6  
 }  
}



（2）||短路或 和 | 逻辑或的区别(ExerCise2)

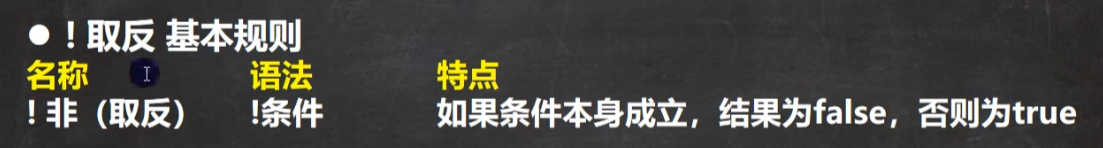




代码演示

public class ExerCise2 {  
 public static void main(String[] args){  
 //演示||短路或 和 | 逻辑或的演示  
 int a = 4;  
 int b = 5;  
 //对于||短路或而言 如果第一个条件为false 后面的条件不在判断 效率高  
 if (a < 6 || ++b < 50){  
 System.*out*.println("ok666");//输出  
 }  
 //对于|逻辑或而言 如果第一个条件为false 后面的条件仍然会判断  
 System.*out*.println("a=" + a + "b=" + b);//4 5  
 if (a < 6 | ++b < 50){  
 System.*out*.println("ko666");//输出  
 }  
 System.*out*.println("a=" + a +"b=" + b);//4 6  
 }  
}

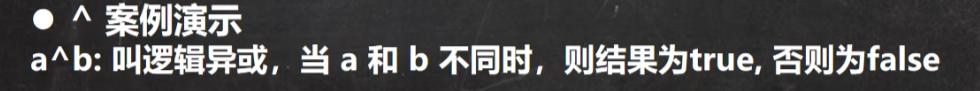
（3）！取反



代码举例

public class ExerCise3 {  
 public static void main (String[]args){  
 System.*out*.println(60 > 20);//ture  
 System.*out*.println(!(60 > 20));//false  
 }  
}

（4）^逻辑异或



//^逻辑与或 两边不同为ture 否则false  
 boolean a = (1 < 5) ^ (3 > 2);  
 System.*out*.println("a=" + a);//false