# 针对基本数据类型

无论什么情况，finally总会执行。

只要碰到return，就会将return时的值放到栈中。

try 或者 catch中有return语句时，就会将要return的数据放到栈中，等finally执行完毕后将栈中数据返回。其他地方无论怎么修改数据，返回的总是当时压栈时候的数据值。

|  |
| --- |
| public class Main {  private static int a=2;    public static void main(String[] args) {    System.out.println("test输出的结果 "+test());  System.out.println("main 自加后 a="+(a++));      }  private static int test(){  try {  a+=3;  System.out.println("try a="+a);  return a;  }catch (Exception e){    }finally {  ++a;  System.out.println("finally a="+a);  }  return a;    }  } |

执行结果：

|  |
| --- |
| try a=5  finally a=6  test输出的结果 5  main 自加后 a=6 |

到了test方法中执行顺序是：首先a+=3，然后输出"try a="+a。

这个时候不会执行try中的return。而是直接到了finally，执行++a，输出"finally a="+a

然后再返回到try中的return a。

这里有个疑问：

Finally中已经将a修改为6了，但是try返回的值为什么还是5.

返回值是存放再栈中的，当有return时就会将这个值压入到栈中，try中有return，就会将a的值放到栈中，但是此时a的值是5，所以在栈中a=5.所以返回的时候取的是栈中值5.

如果finally中有return a。那么返回的值就是finally中执行之后的结果了。

# 针对引用型

引用类型数据finally中的修改对任何一个地方的return都会产生影响。因为引用数据类型在堆中存放了内存地址，finally对值改变时还是指向同一个内存地址，只是地址的值变了。