

```

package com.principle.liskov;

public class Liskov {

    public static void main(String[] args) {
        A a=new A();
        System.out.println("11-3=" + a.func1(11, 3)); System.out.println("1-8=" +
a.func1(1, 8));

        System.out.println("-----");
        B b = new B();
        System.out.println("11-3=" + b.func1(11, 3));//这里本意是求出 11-3
        System.out.println("1-8=" + b.func1(1, 8));// 1-8 System.out.println("11+3+9="
+ b.func2(11, 3));
    }

}

// A 类
class A {
    // 返回两个数的差
    public int func1(int num1, int num2) {
        return num1 - num2;
    }
}

// B 类继承了 A
// 增加了一个新功能：完成两个数相加,然后和 9 求和
class B extends A {
    //这里, 重写了 A 类的方法, 可能是无意识
    public int func1(int a, int b) {
        return a + b;
    }

    public int func2(int a, int b) {
        return func1(a, b) + 9;
    }
}

```