**0628-Java编程语言基础--基础语法(上）**

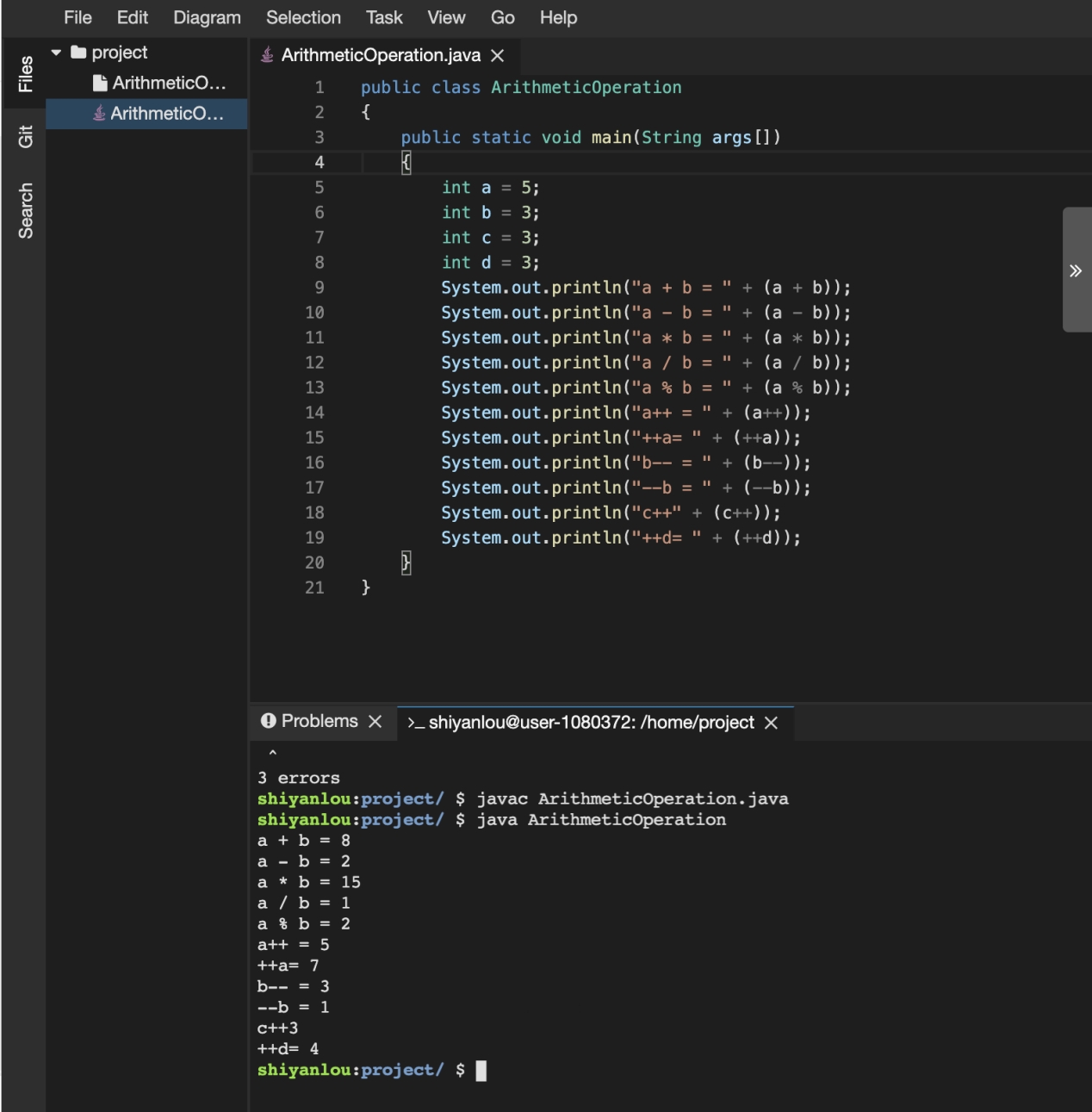
**本节主要学习内容从一个 HelloWorld 示例程序开始，主要学习变量、常量、数据类型、String、运算符、关键字和语句、方法以及 IDE 等Java基础语法，涉及的知识点包括变量、常量、数据结构、String、运算符、关键字和语句、方法等。**

**(接上节)**

**9.运算符**

**运算符顾名思义是一种符号，它是连接一个以上的操作符，实现某种功能的运算。**

**(1)算数运算符**



public class ArithmeticOperation

{

public static void main(String args[])

{

int a = 5;

int b = 3;

int c = 3;

int d = 3;

System.out.println("a + b = " + (a + b));

System.out.println("a - b = " + (a - b));

System.out.println("a \* b = " + (a \* b));

System.out.println("a / b = " + (a / b));

System.out.println("a % b = " + (a % b));

System.out.println("a++ = " + (a++));

System.out.println("++a= " + (++a));

System.out.println("b-- = " + (b--));

System.out.println("--b = " + (--b));

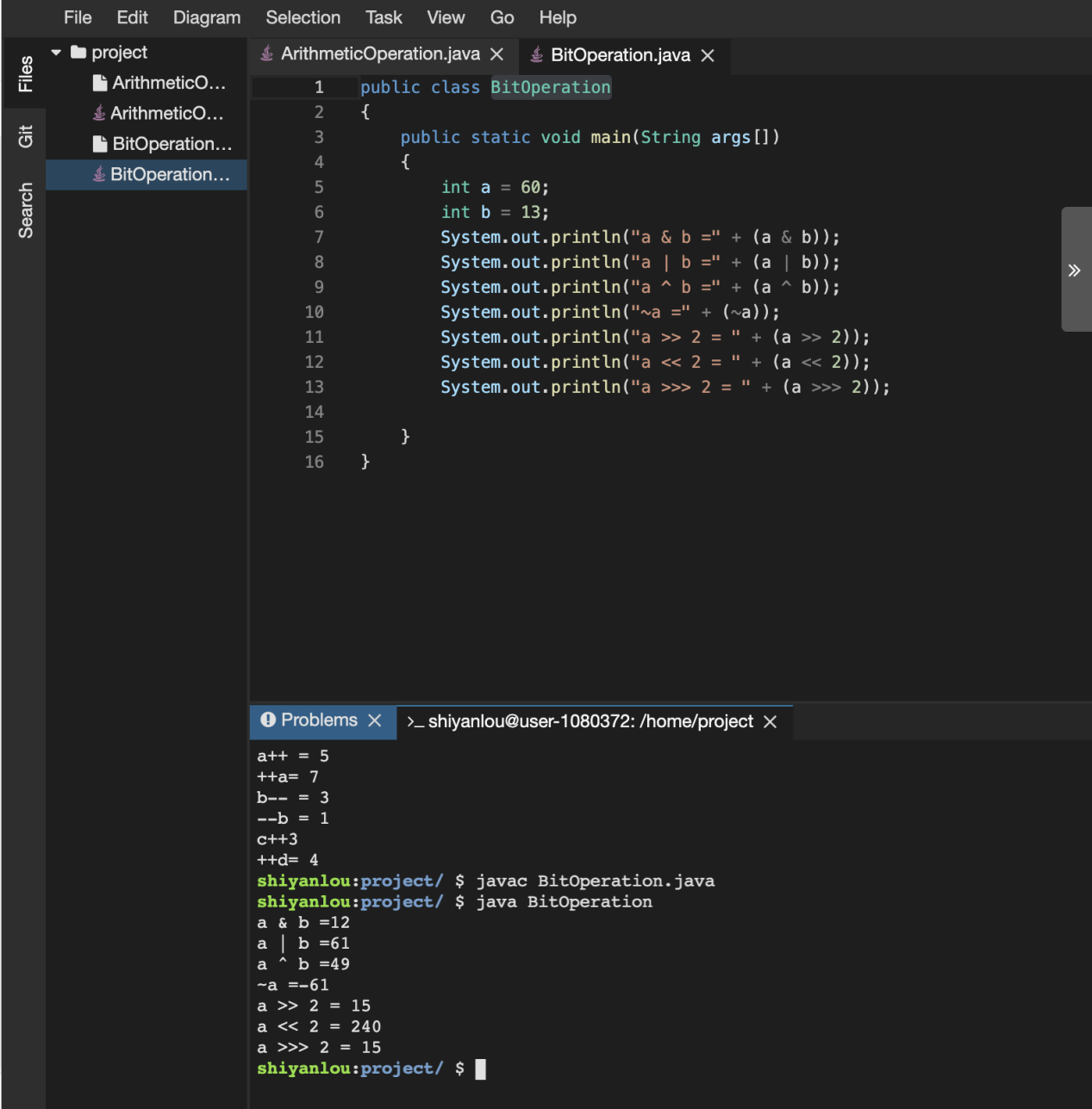
System.out.println("c++" + (c++));

System.out.println("++d= " + (++d));

}

}

**(2)位运算符**

****

public class BitOperation

{

public static void main(String args[])

{

int a = 60;

int b = 13;

System.out.println("a & b =" + (a & b));

System.out.println("a | b =" + (a | b));

System.out.println("a ^ b =" + (a ^ b));

System.out.println("~a =" + (~a));

System.out.println("a >> 2 = " + (a >> 2));

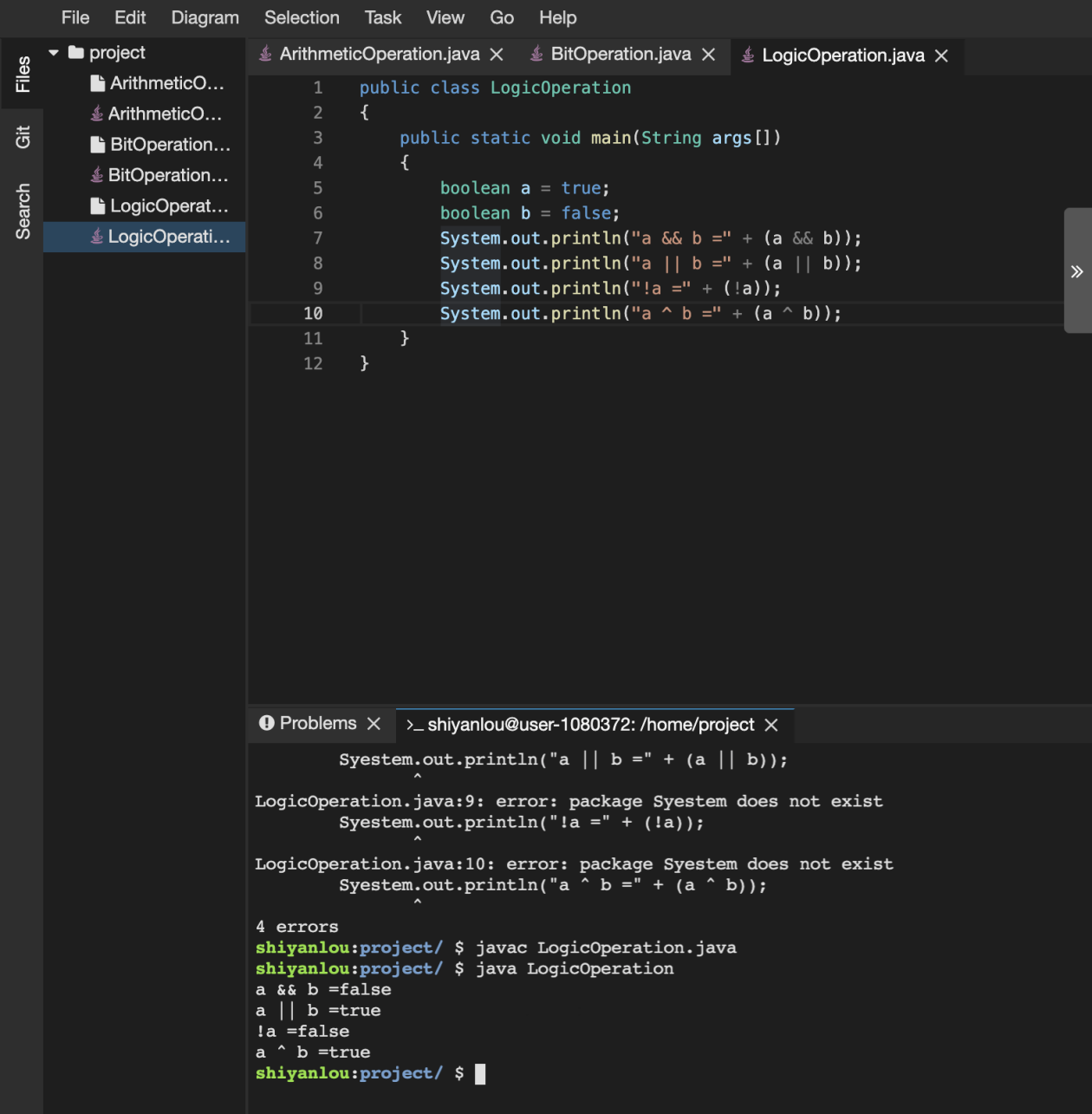
System.out.println("a << 2 = " + (a << 2));

System.out.println("a >>> 2 = " + (a >>> 2));

}

}

**(3)逻辑运算符**

****

public class LogicOperation

{

public static void main(String args[])

{

boolean a = true;

boolean b = false;

System.out.println("a && b =" + (a && b));

System.out.println("a || b =" + (a || b));

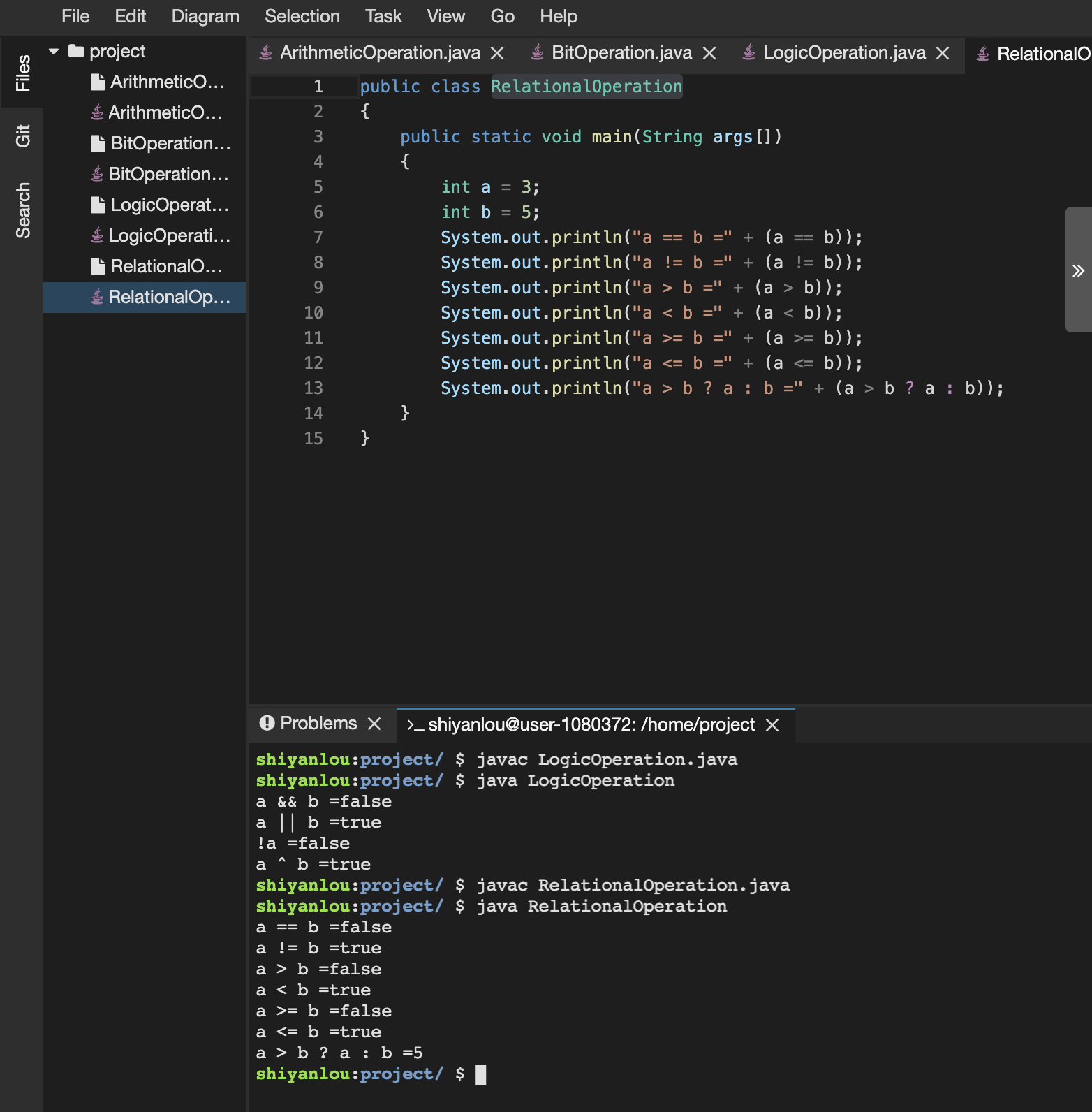
System.out.println("!a =" + (!a));

System.out.println("a ^ b =" + (a ^ b));

}

}

**(4)关系运算符**

****

public class RelationalOperation

{

public static void main(String args[])

{

int a = 3;

int b = 5;

System.out.println("a == b =" + (a == b));

System.out.println("a != b =" + (a != b));

System.out.println("a > b =" + (a > b));

System.out.println("a < b =" + (a < b));

System.out.println("a >= b =" + (a >= b));

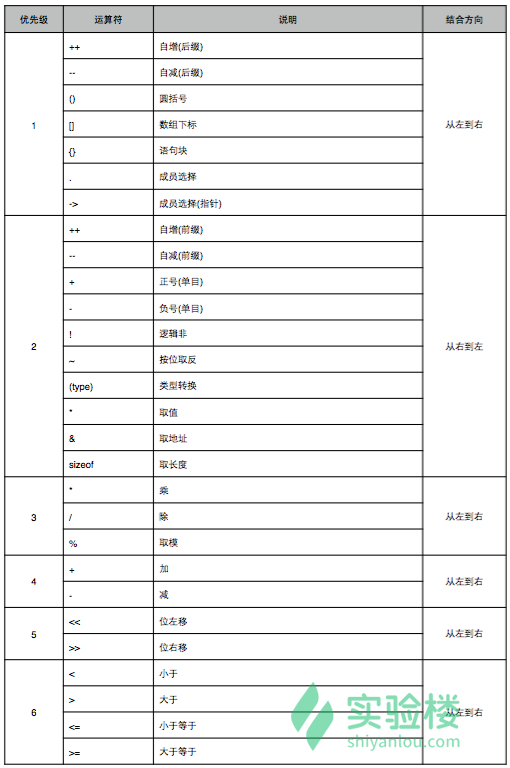
System.out.println("a <= b =" + (a <= b));

System.out.println("a > b ? a : b =" + (a > b ? a : b));

}

}

**(5)运算符优先级**





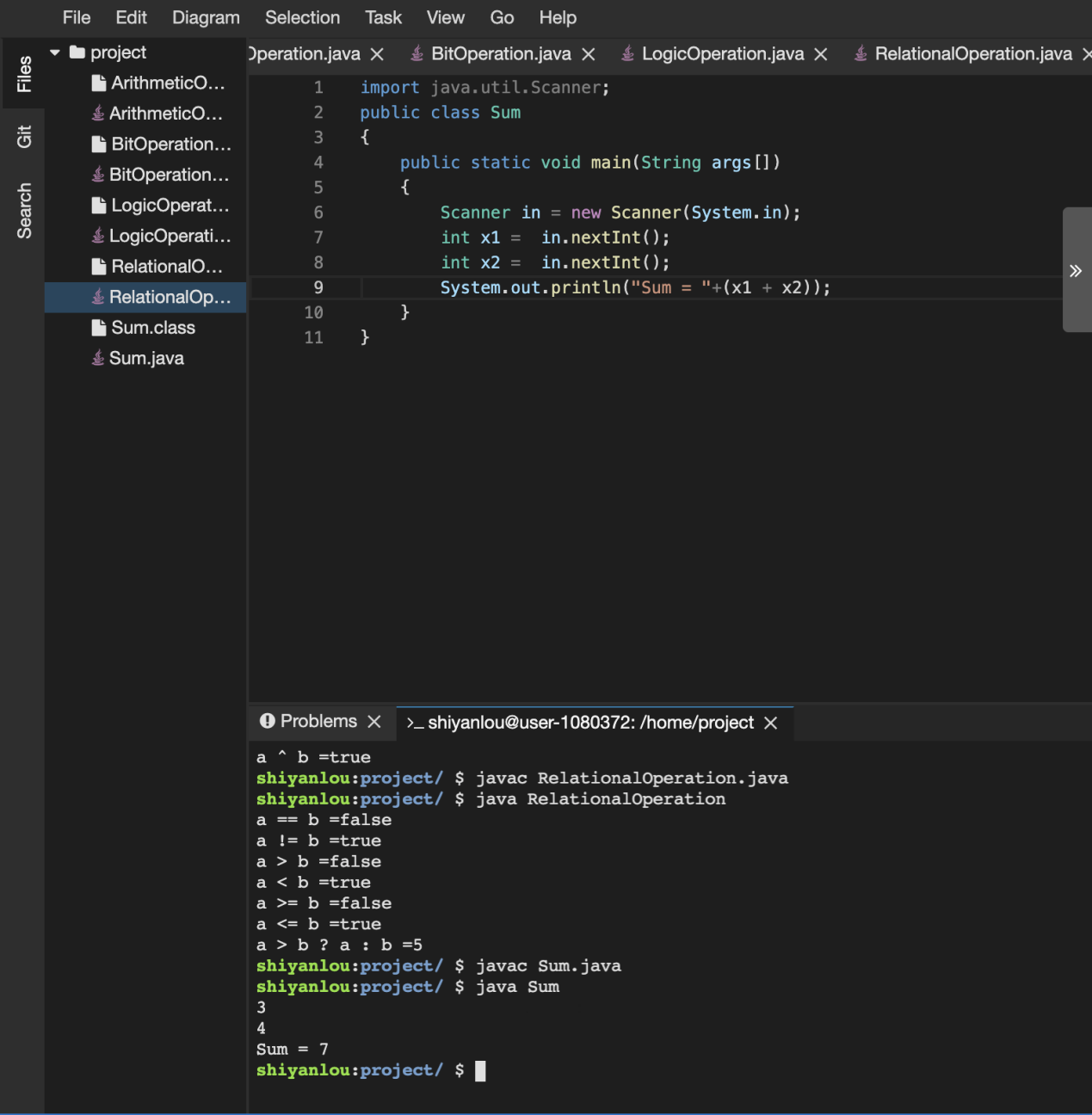
1. **练习--计算数字和**

**在/home/project/目录下新建文件Sum.java，你需要实现以下需求：**

**获取控制台输入的两个整型参数**

**输出两个整型参数和**

**比如输入 3 和 4 对应输出 7**

****

import java.util.Scanner;

public class Sum

{

public static void main(String args[])

{

Scanner in = new Scanner(System.in);

int x1 = in.nextInt();

int x2 = in.nextInt();

System.out.println("Sum = "+(x1 + x2));

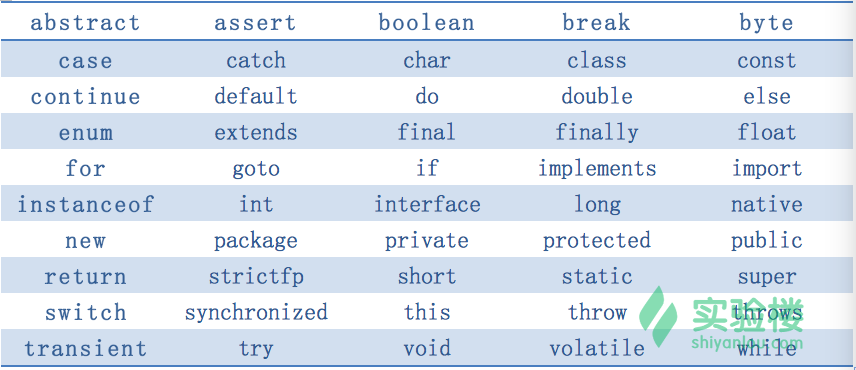
}

}

1. **关键字和语句**

**Java 的关键字对 java 的编译器有特殊的意义，他们用来表示一种数据类型，或者表示程序的结构等，关键字不能用作变量名、方法名、类名、包名。**

**Java 关键字有如下表所列，目前共有 50 个 Java 关键字，其中，"const"和"goto"这两个关键字在 Java 语言中并没有具体含义。**



1. **方法**

**(1)访问修饰符 返回值类型 方法名(参数列表)**

**{ 方法体 }**

**访问修饰符：代表方法允许被访问的权限范围， 可以是 public、protected、private 或者省略（default），其中 public 表示该方法可以被其他任何代码调用。**

**返回值类型：方法返回值的类型，如果方法不返回任何值，则返回值类型指定为 void (代表无类型)；如果方法具有返回值，则需要指定返回值的类型，并且在方法体中使用 return 语句返回值。**

**方法名：是方法的名字，必须使用合法的标识符。**

**参数列表：是传递给方法的参数列表，参数可以有多个，多个参数间以逗号隔开，每个参数由参数类型和参数名组成，以空格隔开。当方法被调用时，传递值给参数。这个值被称为实参或变量。参数列表是指方法的参数类型、顺序和参数的个数。参数是可选的，方法可以不包含任何参数。**

**方法体：方法体包含具体的语句，定义该方法的功能。**

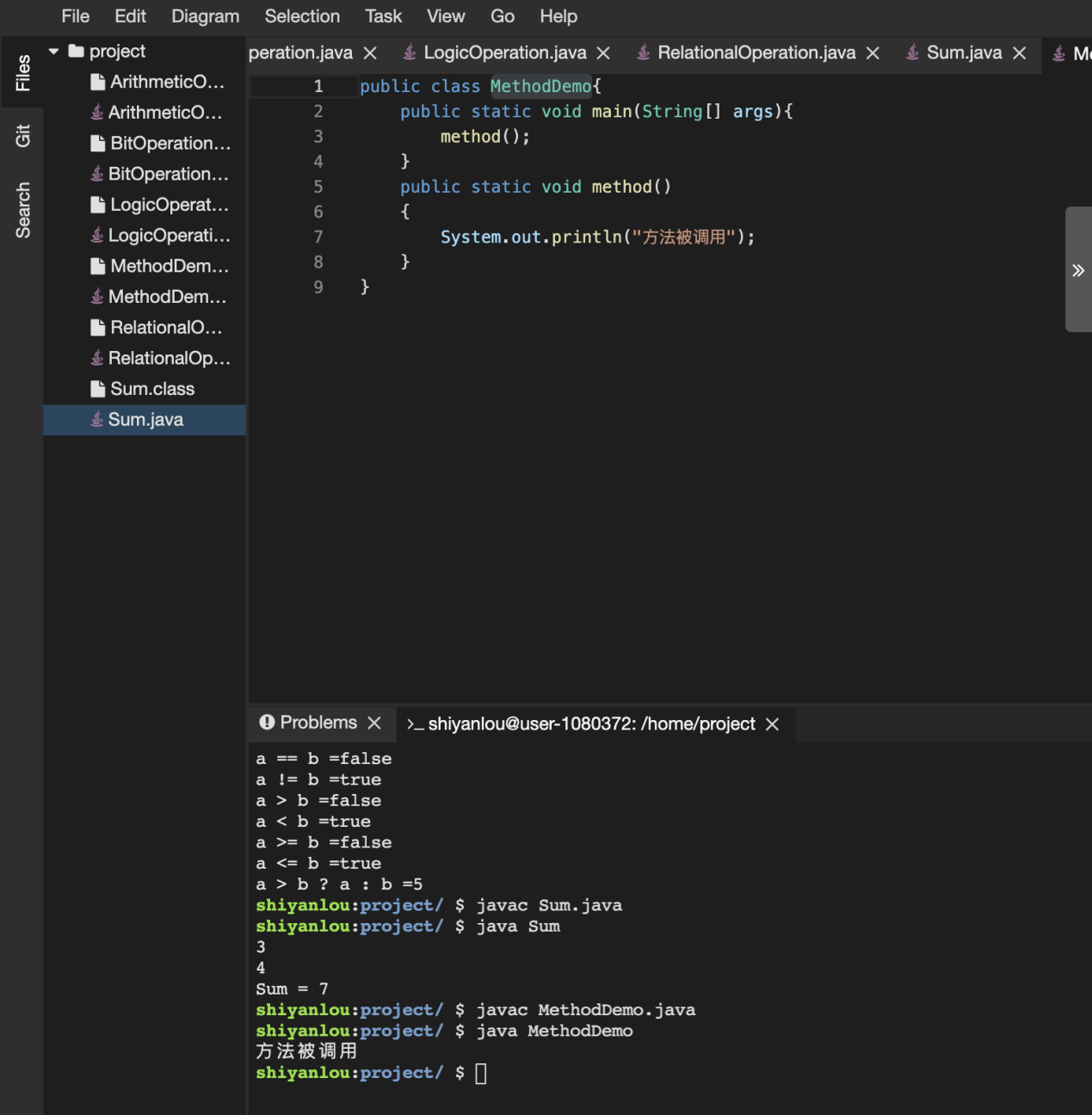
**根据方法是否带参、是否带返回值，可将方法分为四类：**

**无参无返回值方法**

**无参带返回值方法**

**带参无返回值方法**

**带参带返回值方法**

****

public class MethodDemo{

public static void main(String[] args)

{

method();

}

public static void method()

{

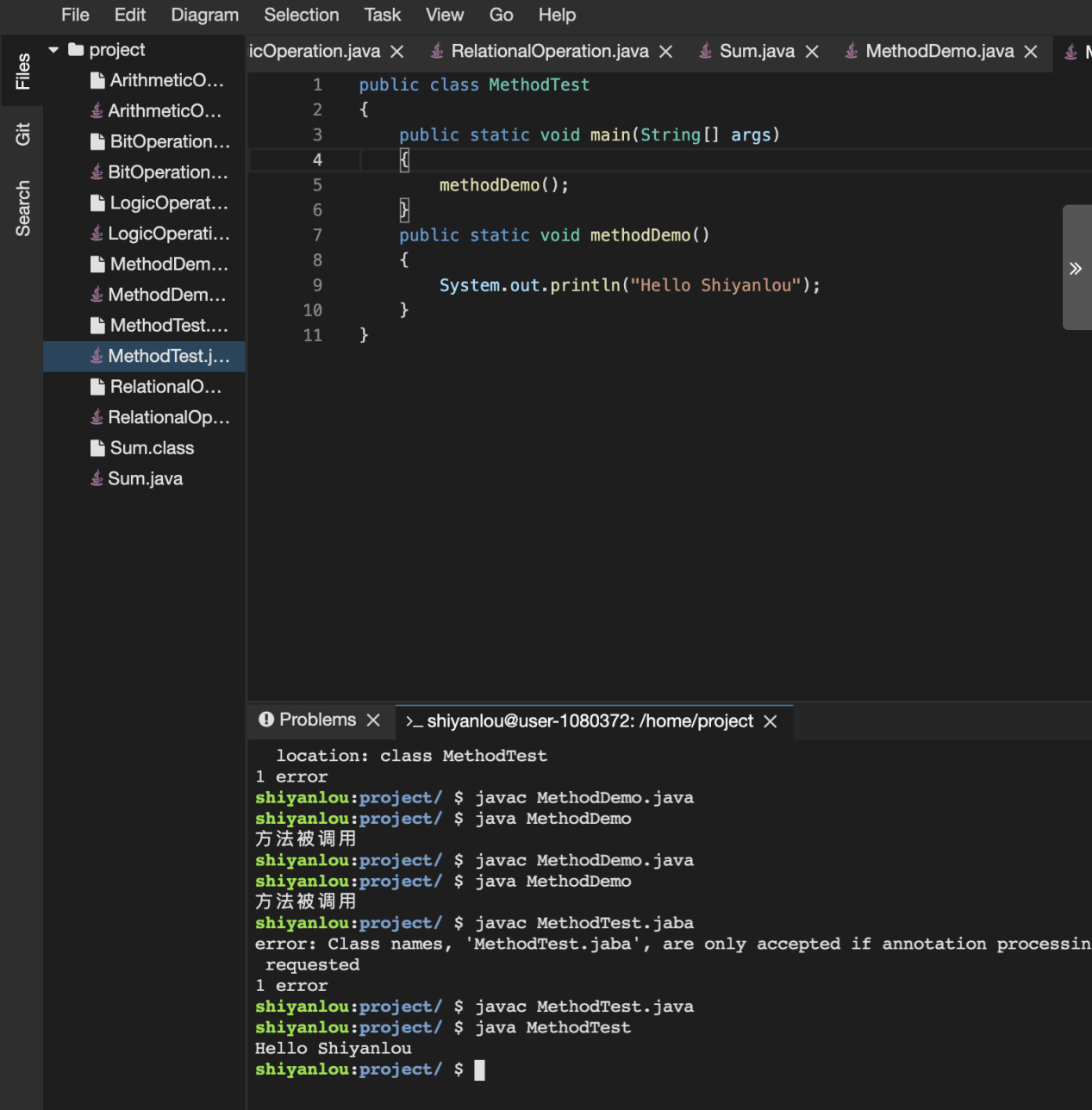
System.out.println("方法被调用");

}

}

1. **练习--方法的使用**

**在/home/project/目录下新建文件MethodTest.java，在其中新建一个方法methodDemo，运行该方法，在控制台输出Hello Shiyanlou。**

****

public class MethodTest

{

public static void main(String[] args)

{

methodDemo();

}

public static void methodDemo()

{

System.out.println("Hello Shiyanlou");

}

}