## 编写一个类TestArray，它只有一个main()方法，在该方法中，创建一个int类型的一维数组sim，元素个数不少于5，对数组赋予初始值，实现数组sim的元素从小到大排序，并输出排序后数组的值。

源代码：  
import java.util.Arrays;

class TestArray {

    public static void main(String[] args) {

        int[] sim = {1,1,4,5,1,4,1,9,1,9,8,1,0};

        int arrayLength = sim.length;

        for (int i = 0; i < arrayLength; i++) {

            int temp;

            boolean flag = false;

            for (int j = 0; j < arrayLength - i - 1; j++) {

                if (sim[j] > sim[j + 1]) {

                    temp = sim[j];

                    sim[j] = sim[j + 1];

                    sim[j + 1] = temp;

                    flag = true;

                }

            }

            if (flag == false) {

                break;

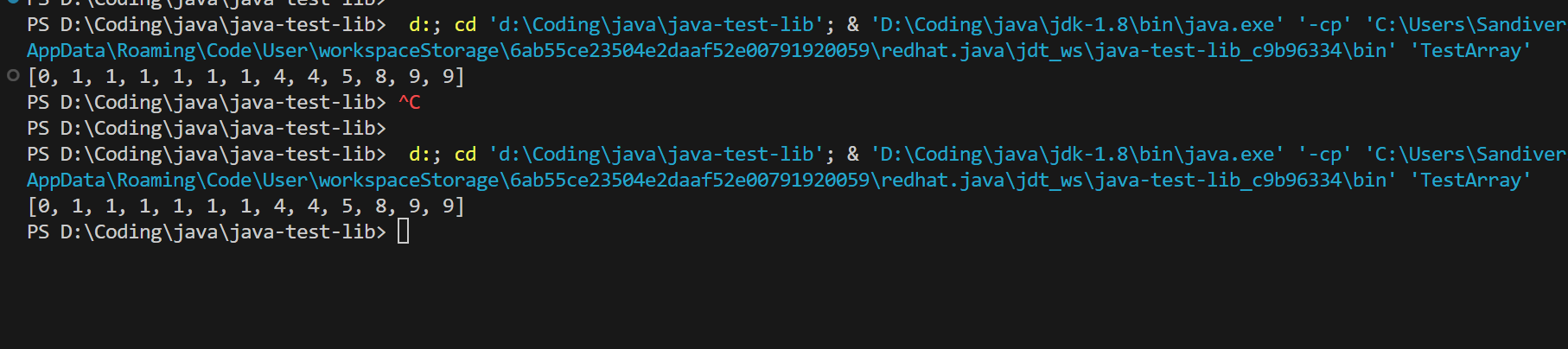
            }

        }

        System.out.println(Arrays.toString(sim));

    }

}

运行截图：  


## 编写一个类TestString,它只有一个main()方法，在该方法中，创建数个String类型的数据，尝试用课件中提到方法操作该字符串，并输出结果。要考虑结果的可读性

源代码：

class TestString {

public static void main(String[] args) {

String test = "Hello world!";

String test2 = "I am a student from University of Electronic Science and Technology of China.";

String test3 = "I am currently studying in School of Information and Software Engineering";

char[] test4 = {'a', 'b', 'c'};

System.out.println(test.charAt(4));

test2.getChars(0, 1, test4, 0);

System.out.println(test3.getBytes());

System.out.println(test.toCharArray());

System.out.println(test.equals("Hello world!"));

System.out.println(test2.equalsIgnoreCase(test3));

System.out.println(test.startsWith("Hell"));

System.out.println(test.endsWith("world!"));

System.out.println(test3.indexOf(66));

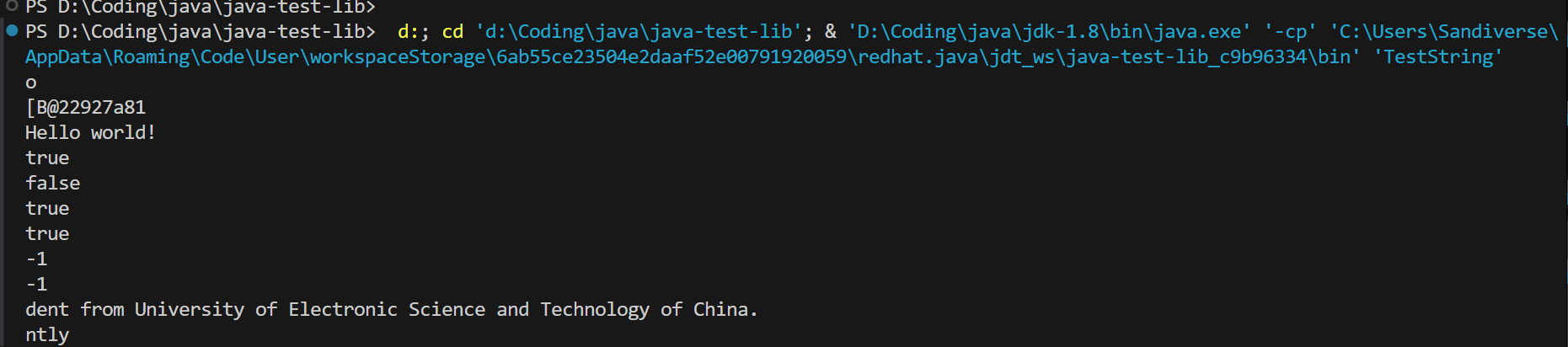
System.out.println(test.lastIndexOf(66));

System.out.println(test2.substring(10));

System.out.println(test3.substring(10, 15));

}

}

运行截图：  


## String的toUpperCase()和toLowerCase()可以把一个字符串中的字符转变为大写或者小写。请编写一个程序，实现两个方法完成相同的功能，但是不能使用上述两个方法。

源代码：

class ConvertString {

    public static void main(String[] args) {

        String str1 = "SDHUIOEHV";

        String str2 = "asdifdsiav";

        System.out.println("To Lower Input: " + str1);

        System.out.println("To Lower Output: " + ToLower(str1));

        System.out.println("To Upper Input: " + str2);

        System.out.println("To Upper Output: " + ToUpper(str2));

    }

    public static String ToLower(String str) {

        for (int i = 0; i < str.length(); i++) {

            char curr = str.charAt(i);

            if (curr >= 'A' && curr <= 'Z') {

                curr += 'a' - 'A';

                str = str.substring(0, i) + curr + str.substring(i + 1);

            }

        }

        return str;

    }

    public static String ToUpper(String str) {

        for (int i = 0; i < str.length(); i++) {

            char curr = str.charAt(i);

            if (curr >= 'a' && curr <= 'z') {

                curr -= 'a' - 'A';

                str = str.substring(0, i) + curr + str.substring(i + 1);

            }

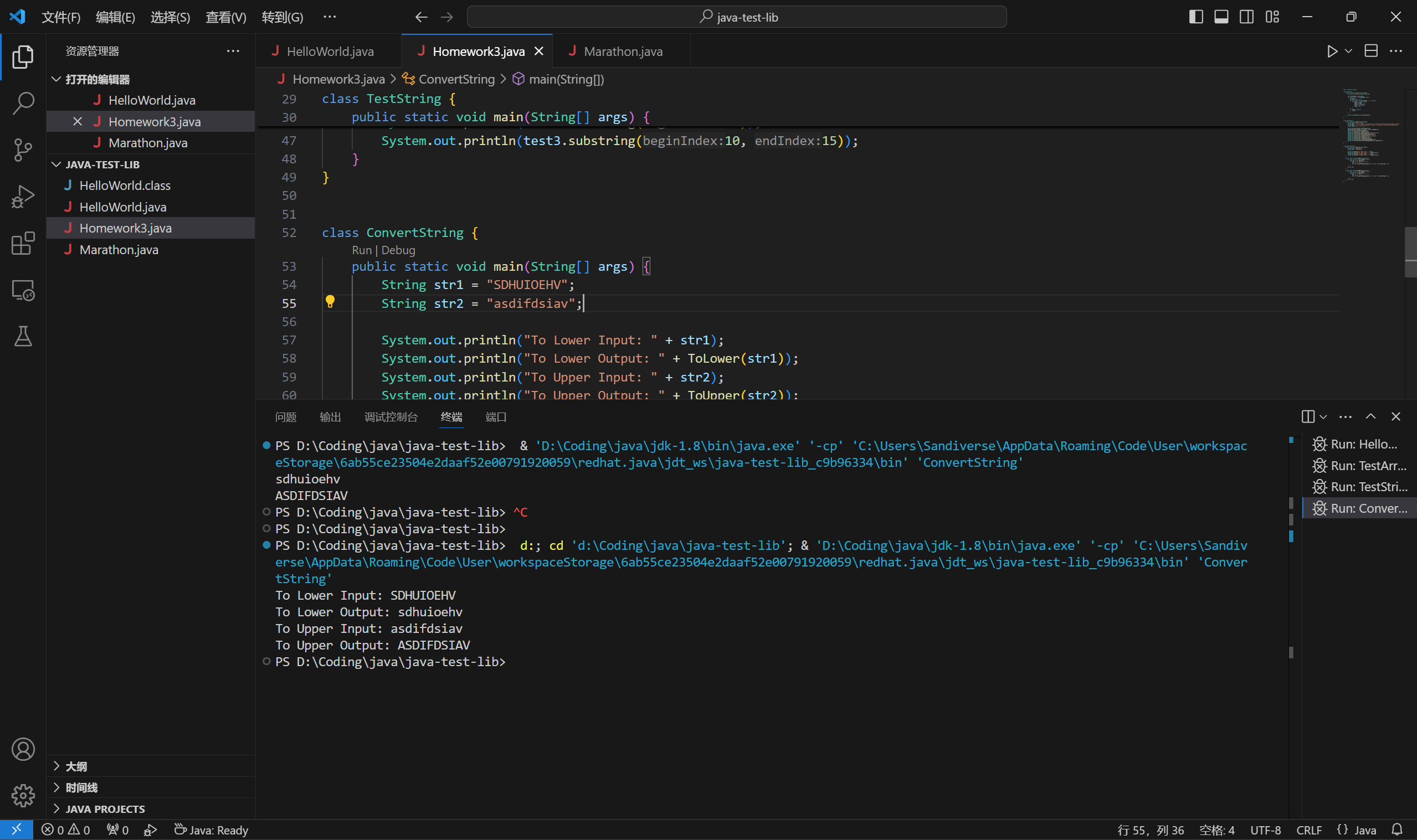
        }

        return str;

    }

}

运行截图：



## 编写一个输出9×9乘法表的程序，注意输出美观。

源代码：

class MultiplicationTable {

    public static void main(String[] args) {

        for (int i = 1; i <= 9; i++) {

            for (int j = 1; j <= i; j++) {

                System.out.printf("%d\*%d=%2d  ", j, i, i \* j);

            }

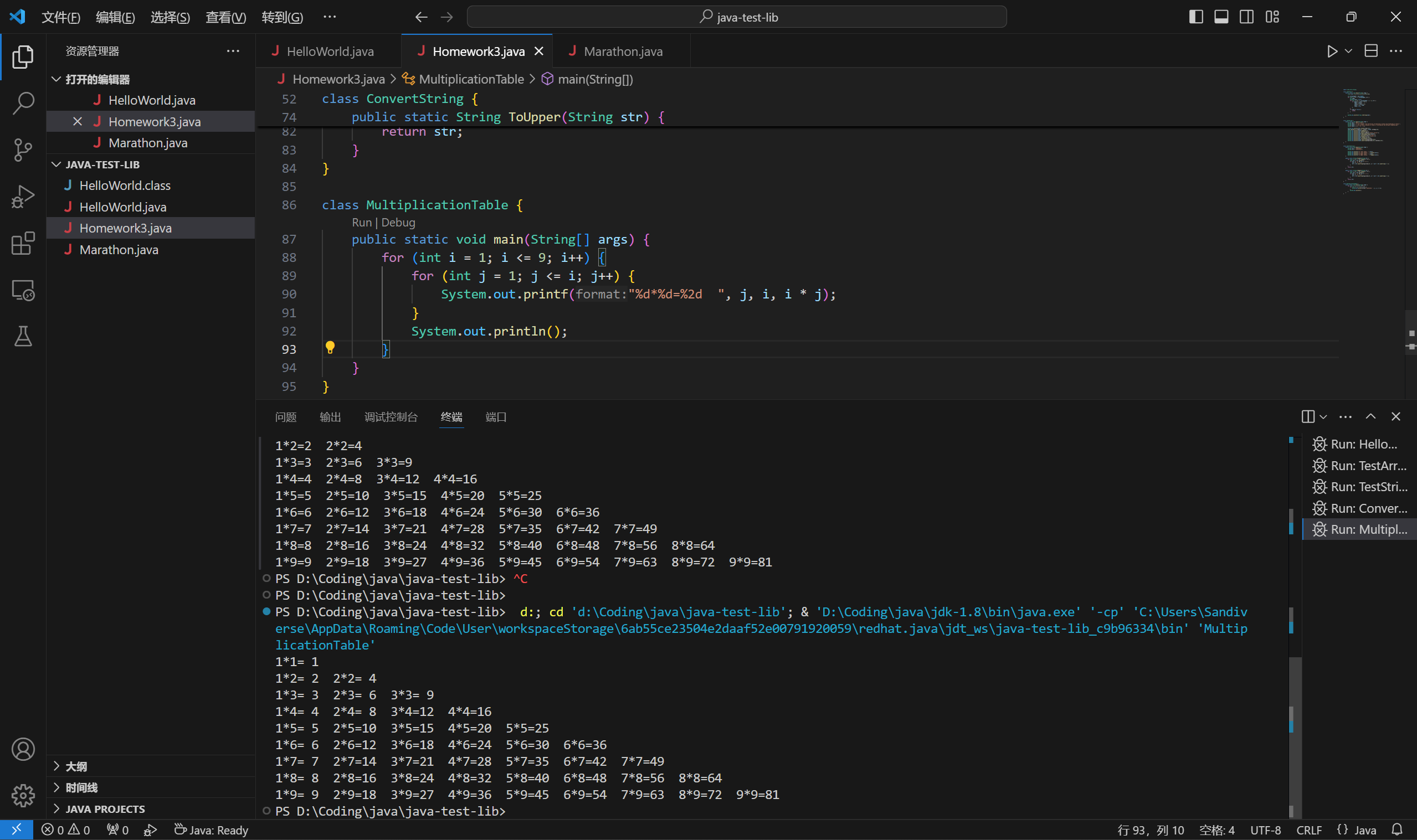
            System.out.println();

        }

    }

}

运行截图：



## 修改第四题的程序，利用带标号的break，使得打印输出改为：打印到6\*6时，不打印6\*6并停止输出后续部分

源代码：

class MultiplicationTable2 {

    public static void main(String[] args) {

        out:

        for (int i = 1; i <= 9; i++) {

            for (int j = 1; j <= i; j++) {

                if (i == 6 && j == 6) {

                    break out;

                }

                System.out.printf("%d\*%d=%2d  ", j, i, i \* j);

            }

            System.out.println();

        }

    }

}

运行截图：

