

东莞理工学院网络空间安全学院

实验报告模板

课程名称：软件测试概论

学期：2022 年秋季

实验名称	数据流测试			实验序号	五
姓 名	皮昊旋	学 号	20214280101 27	班 级	21 杨班
实验地点	8B412	实验日期	20231207	实验类型	设计
同组同学					
教师评语	评阅老师签名(选填)			指导老师	张福勇
				实验成绩	

一、实验目的

- 1、理解数据流测试的原理和方法。
- 2、掌握由数据流测试产生测试用例的方法。
- 3、能够对软件进行数据流测试。

二、实验环境

操作系统：windows

编程语言：自选

三、实验任务

- 1、自编软件的数据流测试：（只测主要函数或过程）
 - 1)、由给定软件规格说明书写出源程序并运行。（见附录 1、2）
 - 2)、用数据流方法产生测试用例（全使用测试）。

- 3)、运行软件，输入测试用例并记录结果。
- 4)、分析结果，调试源程序。（注意：调试前一定要保存上一个版本）
- 5)、重复步骤 3)，直到所有测试用例的输出与预期的一致。

四、实验要求

- 1、做好实验预习，掌握并熟悉本实验中所使用的测试环境及相应的软件。
- 2、完成三中规定的实验任务。
- 3、写出实验报告。

附录 1 规格说明：

编制一元二次方程式 $AX^2+BX+C=0$ 的求根程序。程序有以下功能：

1. 输入 A、B、C 三个系数； $-200 \leq A、B、C \leq 200$
2. 输出根的性质的信息：包括两个相等实根、两个不相等的实根、一个实根、两个虚根、有无数根、无根等。

附录 2 规格说明：

对包含了直角的三角形问题，编写程序输出三角形的类型。要求如下：

1. 输入 a、b、c 三个数作为三角形的三条边； $1 \leq a、b、c \leq 200$ ；
2. 输出三角形类型：包括等边三角形、等腰直角三角形、等腰非直角三角形、不等边直角三角形、不等边非直角三角形、非三角形等。

五、实验过程

①找出两个附录程序的定义节点和使用节点：

对于附录一的程序：

定义变量	定义节点	使用节点
A	18	24、28
B	20	24、30
C	22	24、32
D	24	48、52

对于附录二的程序：

定义变量	定义节点	使用节点
a	9	16、22、28、34、42、48
b	11	16、22、28、34、42、48
c	13	16、22、28、34、42、48

②找出两个程序的定义使用路径

对于附录一的程序，有如下定义使用路径：

定义变量	Du-path	Dc-path
A	18-24	Y
	18-24-28	Y
B	20-24	Y
	20-24-30	Y
C	22-24	Y
	22-24-32	Y
D	24-48	Y
	24-52	Y

对于附录二的程序，有如下定义使用路径：

定义变量	Du-path	Dc-path
a	9-16	Y
	9-22	Y
	9-28	Y
	9-34	Y
	9-42	Y
	9-48	Y
b	11-16	Y
	11-22	Y
	11-28	Y

		11-34	Y	
		11-42	Y	
		11-48	Y	
	c	13-16	Y	
		13-22	Y	
		13-28	Y	
		13-34	Y	
		13-42	Y	
		13-48	Y	

③根据定义使用路径给出测试用例：

对于附录一的程序，给出以下测试用例：

序号	测试用例				预期输出	实际输出	错误类型
	A	B	C	D			
1	0	0	0	0	Innumerable root	Innumerable root	无
2	1	2	1	0	same root	same root	无
3	1	5	1	21	Different root	Different root	无

对于附录二的程序，给出以下测试用例：

序号	测试用例			预期输出	实际输出	错误类型
	A	B	C			
1	0	0	0	Invalid input. Please enter positive values for side lengths.	Invalid input. Please enter positive values for side lengths.	无
2	1	2	1	Not a triangle.	Not a triangle.	无
3	1	1	1	Equilateral triangle.	Equilateral triangle.	无
4	1	1	√2	Isosceles right triangle.	Isosceles right triangle.	无
5	2	2	1	Isosceles triangle.	Isosceles triangle.	无
6	3	4	5	Scalene right triangle.	Scalene right triangle.	无
7	4	5	6	Scalene triangle.	Scalene triangle.	无

六、实验中出错信息及处理方法（可选）

七、实验结果（包括实验处理结果和设计心得）

判断三角形代码:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main(int argc, char const *argv[])
5  {
6      // 输入三角形的边长
7      double a, b, c;
8      cout << "Enter the length of side a: ";
9      cin >> a;
10     cout << "Enter the length of side b: ";
11     cin >> b;
12     cout << "Enter the length of side c: ";
13     cin >> c;
14
15     // 判断三角形的类型
16     if ((a < 1 || a > 200) || (b < 1 || b > 200) || (c < 1 || c > 200))
17     {
18         cout << "Invalid input. Please enter positive values for side lengths." << endl;
19     }
20     else
21     {
22         if (a + b <= c || a + c <= b || b + c <= a)
23         {
24             cout << "Not a triangle." << endl;
25         }
26         else
27         {
28             if (a == b && b == c)
29             {
30                 cout << "Equilateral triangle." << endl;
31             }
32             else
33             {
34                 if ((a == b && a * a + b * b == c * c) ||
35                     (a == c && a * a + c * c == b * b) ||
36                     (b == c && b * b + c * c == a * a))
37                 {
38                     cout << "Isosceles right triangle." << endl;
39                 }
40                 else
41                 {
42                     if (a == b || a == c || b == c)
43                     {
44                         cout << "Isosceles triangle." << endl;
45                     }
46                     else
47                     {
48                         if (a * a + b * b == c * c || a * a + c * c == b * b || b * b + c * c == a * a)
49                         {
50                             cout << "Scalene right triangle." << endl;
51                         }
52                         else
53                         {
54                             cout << "Scalene triangle." << endl;
55                         }
56                     }
57                 }
58             }
59         }
60     }
61
62     return 0;
63 }
64
```

方程求根代码:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int checkScope(float a)
5  {
6      if (a < -200 || a > 200)
7      {
8          return 0;
9      }
10     return 1;
11 }
12
13 int main(int argc, char const *argv[])
14 {
15     float A, B, C;
16
17     cout << "please input A:" << endl;
18     cin >> A;
19     cout << "please input B:" << endl;
20     cin >> B;
21     cout << "please input C:" << endl;
22     cin >> C;
23
24     float D = sqrt(pow(B, 2) - 4 * A * C);
25
26     if (checkScope(A) && checkScope(B) && checkScope(C))
27     {
28         if (A == 0)
29         {
30             if (B == 0)
31             {
32                 if (C == 0)
33                 {
34                     cout << "Innumerable root" << endl;
35                 }
36                 else
37                 {
38                     cout << "No root" << endl;
39                 }
40             }
41             else
42             {
43                 cout << "only one" << endl;
44             }
45         }
46         else
47         {
48             if (D == 0)
49             {
50                 cout << "same root" << endl;
51             }
52             else if (D > 0)
53             {
54                 cout << "Different root" << endl;
55             }
56             else{
57                 cout << "Complex number root" << endl;
58             }
59         }
60     }
61     else
62     {
63         cout << "Invalid" << endl;
64     }
65     return 0;
66 }
67
68
```

实验心得：

在本次数据流测试实验中，通过对两个程序（求解一元二次方程和判断三角形类型）的数据流分析，设计了相应的测试用例，并在实际运行中记录了预期输出和实际输出，进而进行了错误类型的分析。

在附录一的一元二次方程求根程序中，通过对变量 A、B、C 的定义和使用节点进行了路径分析。通过数据流路径的设计，生成了三组测试用例，且实验结果与预期一致，说明程序在测试方面是正确的。

在附录二的三角形类型判断程序中，同样进行了变量 a、b、c 的定义和使用路径分析，生成了七组测试用例。且实验结果与预期相符，没有发现错误。这说明通过数据流测试方法，我们能够有效地设计测试用例，覆盖程序中的各个数据流路径，从而提高了测试的全面性和准确性。

通过本次实验，我更深入地理解了数据流测试的原理和方法，掌握了由数据流测试产生测试用例的技能。同时，在实际应用中，通过对程序的数据流进行分析，我能够更加有针对性地设计测试用例，提高了测试的效率和质量。这对于软件测试的学习和实践都具有积极的意义。