

实 验 报 告

（ 2024 / 2025学年 第1学期）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 软件工程理论与实践 | | | | | |
| 实验名称 | 实验4: 软件测试设计 | | | | | |
| 实验时间 | 2024 | 年 | 11 | 月 | 14 | 日 |
| 指导单位 |  | | | | | |
| 指导教师 |  | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 |  | 班级学号 |  |
| 学院(系) | 物联网学院 | 专 业 | 软件工程 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | | 实验4:软件测试设计 | | | | | 指导教师 | | |  |
| 实验类型 | | 上机 | | 实验学时 | | 4 | 实验时间 | | | 2024.11.14 |
| 1. 实验目的   1. 掌握软件测试的策略  2. 掌握软件测试的方法  3. 学会设计测试用例  4. 学会测试结果的分析 | | | | | | | | | | |
| 二、实验环境(实验设备)  1. 每位学生配备计算机一台  2. 计算机需安装文字处理系统（word）、绘图软件（visio）等 | | | | | | | | | | |
| 三、实验内容  **1.白盒测试练习**  基本路径测试是在程序控制流图的基础上，通过分析控制结构的环路复杂性，导出基本可执行路径集合，从而设计测试用例的方法。伪代码如下：  #include "stdio.h"  void main()  { ①  int n,i;  int x=0;  int y=0;  printf("请输入n\n");  scanf("%d",&n);  printf("请输入i\n");  scanf("%d",&i);  while ( n-- > 0 ) ②  {  if ( i==0 ) ③  x=y+2; ④  else  if ( i==1 ) ⑤  y=y+10; ⑥  else  x=y+20; ⑦  }// while ⑧  printf("x=%d,y=%d\n",x,y);  }// main ⑨  问题：  （1）画出上述算法的程序流程图和程序流图；  （2）用三种方法求程序的环复杂度并设计该程序的基本路径测试用例。  （1）程序流程图和程序流图    图1程序流程图（左）和程序流图（右）  （2）环复杂度  ① V(G)=E-N+2=10-8+2=4（E为流图的边数，N为流图的结点数）  ② V(G)=P+1=3+1=4（P为包含在流图G中的判定结点数）  ③ 域的数量与环复杂性相对应：V(G)=4  基本路径测试用例：   |  |  |  |  | | --- | --- | --- | --- | | **编号** | **路径** | **输入数据** | **预期输出** | | 1 | 1-2-9 | n=-1, i=any | x=0, y=0 | | 2 | 1-2-3-4-8-9 | n=1, i=0 | x=2, y=0 | | 3 | 1-2-3-5-6-8-9 | n=1, i=1 | x=0, y=10 | | 4 | 1-2-3-5-7-8-9 | n=1, i=2 | x=20, y=0 |   **2.黑盒测试练习**  输入三个数a、b、c，分别作为三角形的三条边，通过程序判定该三角形的类型是否为一般三角形、等腰三角形或等边三角形。用等价类划分法为该程序设计测试用例。**分析：**题目中给出和隐含的对输入条件的要求：（1）三个数（2）非零数（3）正数（4）两边之和大于第三边（5）等腰（6）等边。如果 a 、 b 、 c 满足条件（ 1 ） ~ （ 3 ），则输出下列四种情况之一：  1) 如果不满足条件（4），则程序输出为 " 非三角形 " 。  2) 如果三条边相等即满足条件（6），则程序输出为 " 等边三角形 " 。  3) 如果只有两条边相等、即满足条件（5），则程序输出为 " 等腰三角形 " 。  4) 如果三条边都不相等，则程序输出为 " 一般三角形 " 。  列出等价类表并编号，根据编号设计覆盖有效等价类和无效等价类的测试用例。  等价类划分表：   |  |  |  | | --- | --- | --- | | **输入条件** | **有效等价类** | **无效等价类** | | 三个数 | a、b、c 为三个数 | 输入个数不足 3 个（例如：a、b 缺少 c） | | 非零数 | a、b、c 均不为零 | a、b 或 c 中存在零 | | 正数 | a、b、c 均为正数 | a、b 或 c 中存在负数 | | 两边之和大于第三边 | 满足a+b>c, b+c>a，a+c>b | 不满足a+b>c，b+c>a，a+c>b | | 等腰 | 仅有两条边相等 (a=b或b=c或a=c) | 所有三条边不相等或三条边全相等 | | 等边 | a=b=c | 三条边不全相等 |   表1等价类划分表  等价类表：   |  |  | | --- | --- | | **编号** | **等价类描述** | | EC1 | 三个数输入正确 | | EC2 | 输入的三个数均不为零 | | EC3 | 输入的三个数均为正数 | | EC4 | 满足三角形不等式，即a+b>c，b+c>a，a+c>b | | EC5 | 三条边全相等（等边三角形） | | EC6 | 仅有两条边相等（等腰三角形） | | EC7 | 三条边都不相等（一般三角形） | | EC8 | 输入个数不足三位数 | | EC9 | a、b 或 c 中有一个或多个为零 | | EC10 | a、b 或 c 中有一个或多个为负数 | | EC11 | 不满足三角形不等式（a+b≤c，b+c≤a，a+c≤b） |   表2 等价类表  测试用例设计：   |  |  |  |  | | --- | --- | --- | --- | | **测试用例编号** | **输入 (a, b, c)** | **预期输出** | **覆盖等价类** | | TC1 | 3, 4, 5 | 一般三角形 | EC1, EC2, EC3, EC4, EC7 | | TC2 | 5, 5, 5 | 等边三角形 | EC1, EC2, EC3, EC4, EC5 | | TC3 | 5, 5, 3 | 等腰三角形 | EC1, EC2, EC3, EC4, EC6 | | TC4 | 1, 2, 3 | 非三角形 | EC1, EC2, EC3, EC11 | | TC5 | 3, 0, 5 | 非三角形 | EC1, EC9 | | TC6 | -3, 4, 5 | 非三角形 | EC1, EC10 | | TC7 | 3, 4 | 输入错误 | EC8 |   表3测试用例设计表 | | | | | | | | | | |
| 四、实验小结（包括问题和解决方法、心得体会等）  在本次实验中，我通过进行白盒测试和黑盒测试的练习，进一步掌握了软件测试的策略和方法，并学会了设计测试用例和分析测试结果。  在白盒测试的练习中，我使用了基本路径测试的方法，通过程序的控制流图分析了代码的环复杂度。通过这种方式，我对程序的执行路径有了更深的理解，能够更加准确地设计测试用例，覆盖到所有可能的基本路径。  在黑盒测试的练习中，我针对三角形判定程序，使用了等价类划分法设计测试用例。在分析输入条件的有效和无效等价类时，面临的主要挑战是如何全面地考虑所有可能的情况。为了解决这一问题，我首先明确了题目给出的所有输入条件，然后通过细致的分类，列出了每一种输入可能的有效和无效等价类，从而确保测试用例的覆盖范围足够全面。在设计测试用例时，我不仅考虑了正常的输入情况，还包含了一些特殊和异常输入，使得测试用例具有了更高的实用性和健壮性。  通过这次实验，我深刻体会到，软件测试并不仅仅是为了找到程序中的错误，更重要的是通过科学的测试策略和方法来验证程序的正确性和稳定性。通过白盒测试和黑盒测试，我学会了如何从不同角度来验证程序的行为，并掌握了等价类划分和基本路径测试等测试方法的实际应用。  这次实验让我对软件测试有了更加系统和深入的认识，尤其是对测试策略的选择和测试用例的设计有了更清晰的理解。这不仅增强了我的分析问题的能力，也让我认识到测试在软件开发生命周期中的重要性。通过这次实践，我积累了宝贵的经验，也为未来的项目开发打下了良好的基础。 | | | | | | | | | | |
| 成 绩 |  | | 批阅人 | |  | | | 日 期 |  | |