**第一章 白盒测试**

**1.1 实验类型**

实验类型为验证型，4 个学时。

**1.2 实验目的**

（1） 掌握白盒测试技术，并能应用白盒测试技术设计测试用例；

（2） 能用Junit 实施测试。

**1.3 实验设备**

主流PC 机一套，要求安装windows 操作系统、Office、Eclipse 和Junit；

**1.4 实验内容**

**1**．题目一：三角形的类型判断程序

**public class Triangle {**

**protected long lborderA = 0;**

**protected long lborderB = 0;**

**protected long lborderC = 0;**

**// Constructor**

**public Triangle(long lborderA, long lborderB, long lborderC) {**

**this.lborderA = lborderA;**

**this.lborderB = lborderB;**

**this.lborderC = lborderC;**

**}**

**/\*\***

**\* check if it is a triangle**

**\***

**\* @return true for triangle and false not**

**\*/**

**public boolean isTriangle(Triangle triangle) {**

**boolean isTriangle = false;**

**// check boundary**

**if ((triangle.lborderA > 0 && triangle.lborderA <= Long.MAX\_VALUE)**

**&& (triangle.lborderB > 0 && triangle.lborderB <= Long.MAX\_VALUE)&& (triangle.lborderC > 0 && triangle.lborderC <= Long.MAX\_VALUE)) {**

**// check if subtraction of two border larger than the third**

**if (diffOfBorders(triangle.lborderA, triangle.lborderB) < triangle.lborderC**

**&& diffOfBorders(triangle.lborderB, triangle.lborderC) < triangle.lborderA**

**&& diffOfBorders(triangle.lborderC, triangle.lborderA) < triangle.lborderB) {**

**isTriangle = true;**

**}**

**}**

**return isTriangle;**

**}**

**/\*\***

**\* Check the type of triangle**

**\***

**\* Consists of "Illegal", "Regular", "Scalene", "Isosceles"**

**\*/**

**public String getType(Triangle triangle) {**

**String strType = "Illegal";**

**if (isTriangle(triangle)) {**

**// Is Regular**

**if (triangle.lborderA == triangle.lborderB**

**&& triangle.lborderB == triangle.lborderC) {**

**strType = "Regular";**

**}**

**// If scalene**

**else if ((triangle.lborderA != triangle.lborderB)**

**&& (triangle.lborderB != triangle.lborderC)**

**&& (triangle.lborderA != triangle.lborderC)) {**

**strType = "Scalene";**

**}**

**// if isosceles**

**else {**

**strType = "Isosceles";**

**}**

**}**

**return strType;**

**}**

**/\*\***

**\* calculate the diff between borders**

**\***

**\* \*/**

**public long diffOfBorders(long a, long b) {**

**return (a > b) ? (a - b) : (b - a);**

**}**

**/\*\***

**\* get length of borders**

**\*/**

**public long[] getBorders() {**

**long[] borders = new long[3];**

**borders[0] = this.lborderA;**

**borders[1] = this.lborderB;**

**borders[2] = this.lborderC;**

**return borders;**

**}**

**}**

说明：

（1）画出程序的流程图（可以用语句编号表示）。

（2）分别以判定覆盖、MCDC覆盖准则设计测试用例，并分析每个测试用例的可行性。

**2**．题目二：计算最大值

public class Largest{

public int getLargest(int[] array) throws Exception {

if(null == array || 0 == array.length){

throw new Exception("数组不能为空！");

}

int result = array[0];

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

if(result < array[i]){

result = array[i];

}

}

return result;

}

}

（1） 画出程序的控制流图

（2） 试计算此程序段的McCabe 复杂性；

（3） 用基本路径覆盖法设计测试用力；

（4） 分析测试路径执行的可行性。

**1.5** 实验步骤

（1） 根据白盒测试技术设计测试用例，主要考虑逻辑覆盖测试（语句覆盖、判断覆盖、条件覆盖、判断/条件覆盖、条件组合覆盖、路径覆盖）和基本路径测试技术；

（2） 计算测试用例的覆盖率等指标；

(3) 分析测试用例的实际可行性；

(4) 使用Junit编写并运行测试用例；

(5) 使用EclEmma或Clover 进行覆盖分析；

**1.6** 实验要求

（1） 根据题目要求编写测试用例；

（2） 撰写实验报告；