# Лабораторная работа: Тестирование программного продукта методом «Белого ящика»

## 1. Цель работы

Научиться использовать метод «белого ящика» (структурное тестирование, тестирование по маршрутам) для проверки корректности программных алгоритмов.

## 2. Инструктаж на рабочем месте

Инструктаж проводится согласно инструкции по охране труда при работе в лаборатории «Технологии разработки БД, Автоматизированного проектирования технологических процессов и программирования» №319 ИОТ – 029 2016.

## 3. Перечень средств обучения

Персональный компьютер, среда разработки Pascal/C#, текстовый редактор, средство построения блок-схем.

## 4. Теоретическая часть

* Метод «белого ящика» формирует тесты на основе анализа внутренней структуры программы: графа потоков управления, операторов и условий.
* Основные критерии покрытия: покрытие маршрутов (путей), операторов, решений (переходов), условий, решений/условий, комбинаторное покрытие условий, покрытие циклов, покрытие потоков данных.
* Базовое множество независимых путей определяется *цикломатической сложностью* V(G).
* Для циклов применяются специальные наборы тестов: 0, 1, 2, t (t<п), п−1, п, п+1 проходов.

## 5. Индивидуальная задача

**Задача.** Реализовать процедуру t(a,b,x) для преобразования переменной x по условиям:

if (a>1) and (b=0) then x := x / a;  
if (a=2) or (x>1) then x := x + 1;

Необходимо построить алгоритм, реализовать программу, граф передач управления, рассчитать цикломатическую сложность и составить тесты.

## 6. Алгоритм в графической форме (блок-схема)

Start  
 |  
 v  
 Read a, b, x  
 |  
 v  
 [pred1: a>1?]--No-->[pred2: b=0?]--No-->(goto second\_if)  
 |Yes |Yes  
 v v  
 (x := x / a) (x := x / a)  
 | |  
 v v  
second\_if:  
 |  
 v  
 [pred3: a=2?]--Yes--> (x := x + 1)  
 |  
 No  
 v  
 [pred4: x>1?]--Yes--> (x := x + 1)  
 |  
 No  
 v  
 End

## 7. Граф передачи управления и цикломатическая сложность

* Предикатные узлы: 2 if условия.
* V(G) = p + 1 = 2 + 1 = 3
* Независимые пути:
  1. a - x – end if
  2. a - b - x – end if
  3. a - b - y – end if

## 8. Критерии тестирования

* Покрытие операторов
* Покрытие решений
* Покрытие условий
* Покрытие решений/условий
* Комбинаторное покрытие условий
* Покрытие потоков данных
* Покрытие циклов

## 9. Таблица тестов

| № | Назначение теста | Исходные данные (a, b, x) | Ожидаемый результат | Фактический результат |
| --- | --- | --- | --- | --- |
| 1 | Покрытие операторов | 2, 0, 3 | x=2.5 |  |
| 2 | Покрытие операторов | 1,1,1 | x=1 |  |
| 3 | Покрытие решений | 3,0,3 | x=1 |  |
| 4 | Покрытие условий | 1,1,2 | x=3 |  |
| 5 | Комбинаторное покрытие условий | 2,1,1 | x=2 |  |
| 6 | Путь где x не меняется | 1,0,0.5 | x=0.5 |  |
| 7 | Краевой случай | 0,0,5 | Ошибка деления на 0 |  |
| 8 | a>1, b<>0 | 3,1,6 | x=7 |  |

## 10. Программа на Pascal

program TestWhiteBox;  
procedure t(a, b: real; var x: real);  
begin  
 if (a > 1) and (b = 0) then x := x / a;  
 if (a = 2) or (x > 1) then x := x + 1;  
end;  
var a,b,x: real;  
begin  
 readln(a,b,x);  
 t(a,b,x);  
 writeln('x=',x:0:6);  
end.

## 11. Программа на C#

using System;  
  
class WhiteBoxTest  
{  
 static void Transform(ref double x, double a, double b)  
 {  
 if (a > 1 && b == 0)  
 x = x / a;  
 if (a == 2 || x > 1)  
 x = x + 1;  
 }  
  
 static void Main()  
 {  
 Console.WriteLine("Введите a b x через пробел:");  
 string[] input = Console.ReadLine().Split(' ');  
 double a = Convert.ToDouble(input[0]);  
 double b = Convert.ToDouble(input[1]);  
 double x = Convert.ToDouble(input[2]);  
  
 try  
 {  
 Transform(ref x, a, b);  
 Console.WriteLine($"Результат x = {x}");  
 }  
 catch (Exception e)  
 {  
 Console.WriteLine("Ошибка выполнения: " + e.Message);  
 }  
 }  
}

## 12. Контрольные вопросы

1. Почему стратегия структурного тестирования называется также стратегией «белого ящика»?  
   → Тестировщик видит внутреннюю структуру программы.
2. Что показывает цикломатическая сложность алгоритма?  
   → Количество независимых путей и минимальное число тестов для полного покрытия.
3. В чем отличие критериев покрытия условий и покрытия решений/условий?  
   → Покрытие условий проверяет отдельные выражения, решения/условий — все комбинации и ветвления.
4. Какой критерий «белого ящика» самый сильный и почему?  
   → Покрытие решений/условий, обеспечивает проверку всех комбинаций и операторов.
5. Приведите порядок тестирования вложенных циклов.  
   → Сначала внутренний цикл с минимальными параметрами остальных, затем внешний цикл с типовыми значениями вложенных, далее тестируются все циклы поочередно.