ЛАБОРАТОРНАЯ РАБОТА. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

**Цель:** получение навыков проведения функционального тестирования.

Теоретические вопросы

Особенности функционального тестирования программного обеспечения (тестирования

«черного ящика»).

Ошибки, выявляемые при функциональном тестировании. Задачи, решаемые при функциональном тестировании.

***Задание № 1.*** Опишите методы формирования тестовых наборов при использовании стратегии "черного ящика":

|  |  |
| --- | --- |
| Эквивалентное разбиение | Идея тестирования по методу разбиения классов эквивалентности состоит в том, чтобы исключить набор входных данных, которые заставляют систему вести себя одинаково и давать одинаковый результат при тестировании программы. |
| Анализ граничных значений | Это техника проверки поведения продукта на крайних (граничных) значениях входных данных. Граничное тестирование также может включать тесты, проверяющие поведение системы на входных данных, выходящих за допустимый диапазон значений. При этом система должна определённым способом обрабатывать такие ситуации. |
| Анализ причинно-следственных связей | Это структурированный метод, применяемый для определения возможных причин нежелательного события или проблемы. Он систематизирует возможные влияющие факторы в обобщенные категории таким образом, что позволяет рассматривать все возможные гипотезы. |
| Предположение об ошибке | Основная идея метода состоит в том, чтобы составить список, который перечисляет возможные ошибки и ситуации, в которых эти ошибки могли проявиться. |

***Задание № 2.*** Пусть необходимо выполнить тестирование программы, определяющей точку пересечения двух прямых на плоскости. Попутно, она должна определять параллельность прямой одной их осей координат.

В основе программы лежит решение системы линейных уравнений

Ax + By = C и Dx + Ey = F.

1. Используя метод эквивалентных разбиений, получаем для всех коэффициентов один правильный класс эквивалентности (коэффициент – вещественное число) и один неправильный (коэффициент – не вещественное число). Откуда можно предложить 7 тестов:
2. все коэффициенты – вещественные числа;
3. – 7) поочередно каждый из коэффициентов – не вещественное число.
4. По методу граничных условий можно считать, что для исходных данных граничные условия отсутствуют (коэффициенты – "любые" вещественные числа); для результатов – получаем, что возможны варианты: единственное решение, прямые сливаются (множество решений), прямые параллельны (отсутствие решений).

Следовательно, можно предложить тесты, с результатами внутри области и с результатами на границе.

1. По методу анализа причинно-следственных связей определяем множество условий. а) для определения типа прямой;

б) для определения точки пересечения.

Выделяем три группы причинно-следственных связей (определение типа и существования первой линии, определение типа и существования второй линии, определение точки пересечения) и строим таблицы истинности.

К уже имеющимся тестам добавляются:

а) проверки всех случаев расположения обеих прямых – 6 тестов по первой прямой вкладываются в 6 тестов по второй прямой так, чтобы варианты не совпадали, – 6 тестов;

б) выполняется отдельная проверка несовпадения условия x = 0 или y = 0 (в зависимости от того, какой тест был выбран по методу граничных условий) – тест также можно совместить с предыдущими 6 тестами;

1. По методу предположения об ошибке добавим тест, при которомвсе коэффициенты – нули.

Всего получили 20 тестов по всем четырем методикам. Если еще попробовать вложить независимые проверки, то возможно число тестов можно еще сократить.

***Задание № 3.*** Разработать программу определения вида треугольника, заданного длинами его сторон: равносторонний, равнобедренный, прямоугольный, разносторонний.

<HTML>

<link rel="stylesheet" href="treyg.css">

<head>

<title>Треугольники</title>

<script>

function tr (obj ){

let a = Number(obj.num1.value);

let b = Number(obj.num2.value);

let c = Number(obj.num3.value);

let l

let r

let t= "треугольник "

if (a == b && b != c){

t+="равнобедренный"

}

else if(a == b && a == c && c == b){

t+="равносторонний"

}

else if(c\*\*2 == a\*\*2 + b\*\*2){

t+="прямоугольный"

}

else if(a != b && a != c && c != b){

t+="разносторонний"

}

else

t +="построить нельзя"

obj.res.value=t

}

</script>

</head>

<body>

<form name="form1">

<h1> Введите длины сторон:</h1>

Длина стороны 1: <input type="text" name="num1"> <br>

Длина стороны 2: <input type="text" name="num2"> <br>

Длина стороны 3: <input type="text" name="num3"> <br>

<input type="button" value=Вычислить onClick="tr(form1)"><br>

<input type="text" size=28 name="res"><br>

<input type="reset">

</form>

</body>

</HTML>

Предлагаемые тесты свести в таблицу.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Определение вида треугольника | 4 4 4 | Равносторонний треугольник | Равносторонний треугольник | Программа работает без ошибок, все тесты пройдены. |
| 2 | 2 3 4 | Разносторонний треугольник | Разносторонний треугольник |
| 3 | 3 4 4 | Треугольник построить нельзя | Треугольник построить нельзя |
| 4 | 3 4 5 | Прямоугольный треугольник | Прямоугольный треугольник |
| 5 | 4 4 2 | Равнобедренный треугольник | Равнобедренный треугольник |

***Задание № 4.*** Разработать программу решения уравнения *ax*2 + *bx* +*c* = 0, где *a*, *b*, *c* – любые вещестенные числа.

Предлагаемые тесты свести в таблицу.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Решения уравнения *ax*2 + *bx* +*c* = 0 | a=1, b=-3, c=2 | x1 = 2, x2 = 1 | Уравнение имеет два различных решения: x1 = 2, x2 = 1. |  |
| 2 | a=1, b=-2, c=1 | x = 1 | Уравнение имеет одно решение: x = 1 |  |
| 3 | a=1, b=0, c=1 | Уравнение не имеет действительных решений. | Уравнение не имеет действительных решений. |  |
| 4 | a = 0, b ≠ 0 | a=0, b=2, c=4 | x = -2 | Уравнение имеет одно решение: x = -2. |  |
| 5 | a=0, b=0, c=0 | Уравнение имеет бесконечно много решений | Уравнение имеет бесконечно много решений |  |
| 6 | a=0, b=0, c=5 | Уравнение не имеет решений | Уравнение не имеет решений. |  |
| 7 | a=2, b=4, c=-6 | x1 = 1, x2 = -3 | Уравнение имеет два различных решения: x1 = 1, x2 = -3 |  |

using System;

class Program

{

static void Main()

{

Console.WriteLine("Введите коэффициенты a, b и c для уравнения ax^2 + bx + c = 0");

Console.Write("a: ");

double a = Convert.ToDouble(Console.ReadLine());

Console.Write("b: ");

double b = Convert.ToDouble(Console.ReadLine());

Console.Write("c: ");

double c = Convert.ToDouble(Console.ReadLine());

SolveQuadraticEquation(a, b, c);

}

static void SolveQuadraticEquation(double a, double b, double c)

{

if (a == 0)

{

if (b == 0)

{

if (c == 0)

{

Console.WriteLine("Уравнение имеет бесконечно много решений.");

}

else

{

Console.WriteLine("Уравнение не имеет решений.");

}

return;

}

else

{

double x = -c / b;

Console.WriteLine($"Уравнение имеет одно решение: x = {x}");

return;

}

}

double discriminant = b \* b - 4 \* a \* c;

if (discriminant > 0)

{

double x1 = (-b + Math.Sqrt(discriminant)) / (2 \* a);

double x2 = (-b - Math.Sqrt(discriminant)) / (2 \* a);

Console.WriteLine($"Уравнение имеет два различных решения: x1 = {x1}, x2 = {x2}");

}

else if (discriminant == 0)

{

double x = -b / (2 \* a);

Console.WriteLine($"Уравнение имеет одно решение: x = {x}");

}

else

{

Console.WriteLine("Уравнение не имеет действительных решений.");

}

}

}

| Номер теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | |-------------|----------------------------------|--------------------------|------------------------------------------------------|-------------------------------------| | 1 | Два различных действительных корня | a=1, b=-3, c=2 | x1 = 2, x2 = 1 | Программа показывает: Уравнение имеет два различных решения: x1 = 2, x2 = 1. | | 2 | Одно действительное решение | a=1, b=-2, c=1 | x = 1 | Программа показывает: Уравнение имеет одно решение: x = 1. | | 3 | Нет действительных решений | a=1, b=0, c=1 | Уравнение не имеет действительных решений. | Программа сообщает: Уравнение не имеет действительных решений. | | 4 | Проверка случая a = 0, b ≠ 0 | a=0, b=2, c=4 | x = -2 | Программа показывает: Уравнение имеет одно решение: x = -2. | | 5 | Общее решение (все коэффициенты нули) | a=0, b=0, c=0 | Уравнение имеет бесконечно много решений. | Программа сообщает: Уравнение имеет бесконечно много решений. | | 6 | Уравнение не имеет решений | a=0, b=0, c=5 | Уравнение не имеет решений. | Программа сообщает: Уравнение не имеет решений. | | 7 | Два различных действительных корня | a=2, b=4, c=-6 | x1 = 1, x2 = -3 | Программа показывает: Уравнение имеет два различных решения: x1 = 1, x2 = -3. |

Эта программа может быть дополнительно улучшена или адаптирована в зависимости от потребностей. Тесты охватывают различные сценарии, такие как случай с двумя различными корнями, одним корнем, отсутствием корней и специальные случаи с нулевыми коэффициентами.

***Задание № 5.*** Оформить отчет.

Отчет: