Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования

Новосибирский государственный технический университет

Кафедра Автоматизированных систем управления

**ОТЧЁТ**

**по лабораторной работе № 5**

**Тестирование по методу «Черного ящика»**

**по дисциплине: «Технология программирования»**

Выполнили: Проверил:

Студенты гр. «АВТ-813», «АВТФ» Лукоянычева Ольга Викторовна

Букова Анна

Ланг Татьяна

Пустовских Дмитрий

Новосибирск 2020

**Цель работы:** Познакомиться с многосвязными структурами данных, получить представление об использовании таких структур для описания реальных данных, приобрести навыки программирования процедур доступа к данным.

**Задание:**

Разработать программу исследования уравнений второго порядка с двумя неизвестными A+2Bxy+C+2Dx+2Ey+F=0. Программа должна определять вид графика: эллипс, парабола, гипербола, две пересекающиеся прямые, две параллельные прямые, две мнимые прямые.

Вид прямой второго порядка определяется по двум дискриминантам

большому: и малому .

Малый дискриминант для эллипса положителен, для гиперболы отрицателен, для параболы равен нулю. Если большой дискриминант равен нулю, то линия второго порядка распадается на две прямых:

для эллиптического вида - пересекающиеся мнимые прямые (точка), для гиперболического вида - пара пересекающихся действительных прямых, для параболического вида - две параллельные прямые.

**Разработка тестов по методике «черного ящика»**

**Эквивалентное разбиение**

*Исходные данные*

|  |  |  |
| --- | --- | --- |
| Входное условие | Допустимые классы эквивалентности | Недопустимые классы эквивалентности |
| Тип вводимых данных | 3 вещественных числа в диапазоне  -2147483647 X 2147483647 (1) | Нецифровой символ (4),  X<-2147483647 (5),  X > 2147483647 (6) |
| 3 целых числа в диапазоне  -2147483647 X 2147483647 (2) |
| Комбинация вещественных и целых чисел в диапазоне  -2147483647 X 2147483647 (3) |

*Тесты*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Проверка для вещественных чисел | A= 1.5  B= 3.05  C= 2.2  D= 0.7  E= 0.75  F= 3.6 | Гипербола | Гипербола | Тест пройден успешно |
| 2 | Проверка для целых чисел | A=1  B=2  C=1  D=2  E=2  F=1 | Две параллельные прямые | Две параллельные прямые | Тест пройден успешно |
| 3 | Проверка для комбинации целых и вещественных чисел | A=1  B=2  C=1.5  D=2  E=2.6  F=1 | Эллипс | Эллипс | Тест пройден успешно |
| 4 | Проверка для нецифровых символов | A=1  B=21  C=abc  D=2  E=2.6  F=1 | Возникла проблема: Введённые значения не являются числами. | Возникла проблема: Введённые значения не являются числами. | Тест пройден успешно |
| 5 | Проверка на выход за минимум | A=0  B=8  C=1  D=7.1  E=-2147483648  F=3 | Возникла проблема: Введённые значения выходят из допустимого диапазона снизу. | Возникла проблема: Введённые значения выходят из допустимого диапазона снизу. | Тест пройден успешно |
| 6 | Проверка на выход за максимум | A=8  B=2147483648  C=4.3  D=3  E=6  F=3 | Возникла проблема: Введённые значения выходят из допустимого диапазона сверху. | Возникла проблема: Введённые значения выходят из допустимого диапазона сверху. | Тест пройден успешно |

**Анализ граничных значений**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 7 | Проверка на выход за минимум | A=4  B=-2147483648  C=6  D=7.1  E=9  F=8 | Возникла проблема: Введённые значения выходят из допустимого диапазона снизу. | Возникла проблема: Введённые значения выходят из допустимого диапазона снизу. | Тест пройден успешно |
| 8 | Проверка на погрешность | A=2147483646.99  B=1073741823.495  C=2147483646.99  D=1073741823.495  E=1073741823.495  F=2147483646.99 | Эллипс | Эллипс | Тест пройден успешно |
| 9 | Проверка на выход за максимум | A=1  B=3  C=2147483648  D=5  E=60  F=7 | Возникла проблема: Введённые значения выходят из допустимого диапазона сверху. | Возникла проблема: Введённые значения выходят из допустимого диапазона сверху. | Тест пройден успешно |

**Анализ причинно-следственных связей**

*Таблица истинности*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Результат |
| 1 | 1 | 0 | 0 | Две параллельные прямые |
| 0 | 1 | 0 | 0 | Парабола |
| 1 | 0 | 1 | 0 | Две мнимые прямые |
| 0 | 0 | 1 | 0 | Эллипс |
| 1 | 0 | 0 | 1 | Две пересекающиеся прямые |
| 0 | 0 | 0 | 1 | Гипербола |

*Тесты*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 10 | Проверка на эллипс | A=3  B=0.5  C=3  D=0.5  E=0.5  F=2 | Эллипс | Эллипс | Тест пройден успешно |
| 11 | Проверка на параболу | A=1  B=1  C=1  D=2  E=1  F=2 | Парабола | Парабола | Тест пройден успешно |
| 12 | Проверка на гиперболу | A=1  B=2  C=1  D=1  E=1  F=2 | Гипербола | Гипербола | Тест пройден успешно |
| 13 | Проверка на две пересекающиеся прямые | A=1  B=2  C=1  D=1  E=2  F=1 | Две пересекающиеся прямые | Две пересекающиеся прямые | Тест пройден успешно |
| 14 | Проверка на две параллельные прямые | A=1  B=1  C=1  D=1  E=1  F=1 | Две параллельные прямые | Две параллельные прямые | Тест пройден успешно |
| 15 | Проверка на две мнимые прямые | A=2  B=1  C=2  D=2  E=1  F=2 | Две мнимые прямые | Две мнимые прямые | Тест пройден успешно |

**Вывод:**

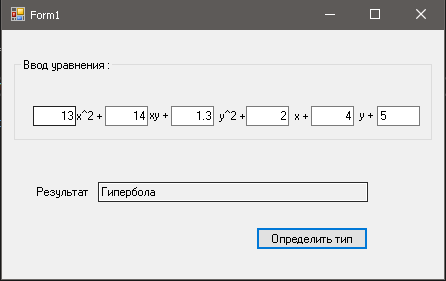
Тестирование по стратегии «чёрного ящика» происходит без рассмотрения внутреннего устройства тестируемого объекта.

* Метод эквивалентного разбиения обнаруживает ошибки при несоответствии

входных значений входным условиям. Недостатком является неполная проверка входных значений на всевозможные комбинации.

* Метод анализа граничных значений позволяет избежать появления ошибок при учете граничных значений входных данных. Недостатком является то, что метод не проверяет комбинации входных значений.
* Метод анализа причинно-следственных связей дает информацию о результате при различных комбинациях входных данных. Недостатком является громоздкость при достаточно большом количестве входных данных и плохое исследование граничных условий.

**Демонстрация работы программы:**



**Код программы:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using lab3\_VM;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

this.textBox1.TextAlign = HorizontalAlignment.Right;

this.textBox2.TextAlign = HorizontalAlignment.Right;

this.textBox3.TextAlign = HorizontalAlignment.Right;

this.textBox4.TextAlign = HorizontalAlignment.Right;

this.textBox5.TextAlign = HorizontalAlignment.Right;

}

private void showMessage(string msg)

{

MessageBox.Show("Возникла проблема: " + msg, "Ошибка!!!");

}

private bool checkNum(string num)

{

bool isNum;

double Num;

isNum = double.TryParse(num, out Num);

return isNum;

}

private bool checkNumRang(string num)

{

int MaxLength = 10;

string MaxNumm = "2147483647";

string[] NumArr = num.Split(new char[] {','});

string wholeNum = NumArr[0];

wholeNum.Trim(new char[] { '-' });

if (wholeNum.Length < MaxLength)

{

return true;

}

if (wholeNum.Length > MaxLength)

{

return false;

}

if (wholeNum.Length == MaxLength)

{

int temp = String.Compare(wholeNum, MaxNumm);

if(temp == 0)

{

return NumArr.Length == 1;

}

return (temp < 0);

}

return false;

}

private bool checkNUM(string Num)

{

if (Num.Length == 0)

{

showMessage("Не были заполнены поля.");

return false;

}

if (!checkNum(Num))

{

showMessage("Введённые значения не являются числами.");

return false;

}

if (!checkNumRang(Num))

{

if (Num[0] == '-')

showMessage("Введённые значения выходят из допустимого диапазона снизу.");

else

showMessage("Введённые значения выходят из допустимого диапазона сверху.");

return false;

}

return true;

}

private void button1\_Click(object sender, EventArgs e)

{

string A = textBox1.Text.Replace(".", ",");

string B = textBox2.Text.Replace(".", ",");

string C = textBox3.Text.Replace(".", ",");

string D = textBox4.Text.Replace(".", ",");

string E = textBox5.Text.Replace(".", ",");

string F = textBox6.Text.Replace(".", ",");

string res = "";

Matrix BigDet = new Matrix(3, 3);

Matrix SmolDet = new Matrix(2, 2);

if (!checkNUM(A) || !checkNUM(B)|| !checkNUM(C)|| !checkNUM(D)|| !checkNUM(E)|| !checkNUM(F))

return;

BigDet[0, 0] = Convert.ToDouble(A);

BigDet[0, 1] = Convert.ToDouble(B)/2;

BigDet[0, 2] = Convert.ToDouble(D)/2;

BigDet[1, 0] = Convert.ToDouble(B)/2;

BigDet[1, 1] = Convert.ToDouble(C);

BigDet[1, 2] = Convert.ToDouble(E)/2;

BigDet[2, 0] = Convert.ToDouble(D)/2;

BigDet[2, 1] = Convert.ToDouble(E)/2;

BigDet[2, 2] = Convert.ToDouble(F);

SmolDet[0, 0] = Convert.ToDouble(A);

SmolDet[0, 1] = Convert.ToDouble(B)/2;

SmolDet[1, 0] = Convert.ToDouble(B)/2;

SmolDet[1, 1] = Convert.ToDouble(C);

double Detsm = SmolDet.getDeterminant();

double DetBg = BigDet.getDeterminant();

if (DetBg == 0)

{

if (Detsm == 0)

{

res = "Две параллельные прямые";

}

if (Detsm > 0)

{

res = "Две пересекающиеся мнимые прямые";

}

if (Detsm < 0)

{

res = "Две пересекающиеся действительные прямые";

}

}

else

{

if (Detsm == 0)

{

res = "Парабола";

}

if (Detsm > 0)

{

res = "Эллипс";

}

if (Detsm < 0)

{

res = "Гипербола";

}

}

this.textBox10.Text = res;

}

}

}

**Класс Matrix:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab3\_VM

{

public class Matrix

{

private double[,] matrix = null;

private int n;

private int m;

public Matrix(int x = 1, int y = 1)

{

matrix = new double[x, y];

n = x;

m = y;

}

public int N { get => n; }

public int M { get => m; }

public double this[int y, int x]

{

get { return matrix[y, x]; }

set { matrix[y, x] = value; }

}

public double getDeterminant()

{

double Deter = 0;

if (this.M == 1)

{

return matrix[0, 0];

}

for (int i = 0; i < N; ++i)

{

Matrix minor = new Matrix(N - 1, M - 1);

int x = 0, y = 0;

for (int j = 1; j < N; ++j)

{

for (int k = 0; k < M; ++k)

{

if (k != i)

{

minor[y, x] = this.matrix[j,k];

x++;

}

}

x = 0;

y++;

}

Deter += Math.Pow(-1, i) \* this.matrix[0,i]\*minor.getDeterminant();

}

return Deter;

}

}

}