Министерство образования и науки Российской Федерации Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌ высшего‌ ‌образования‌

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №1**

Дисциплина: «Объектно-ориентированное программирование»

Тема: Выполнение простой программы. Организация ввода и вывода данных

Вариант 8

Выполнил работу

студент группы РИС-23-1б

Жуланов Н. А.

Проверила

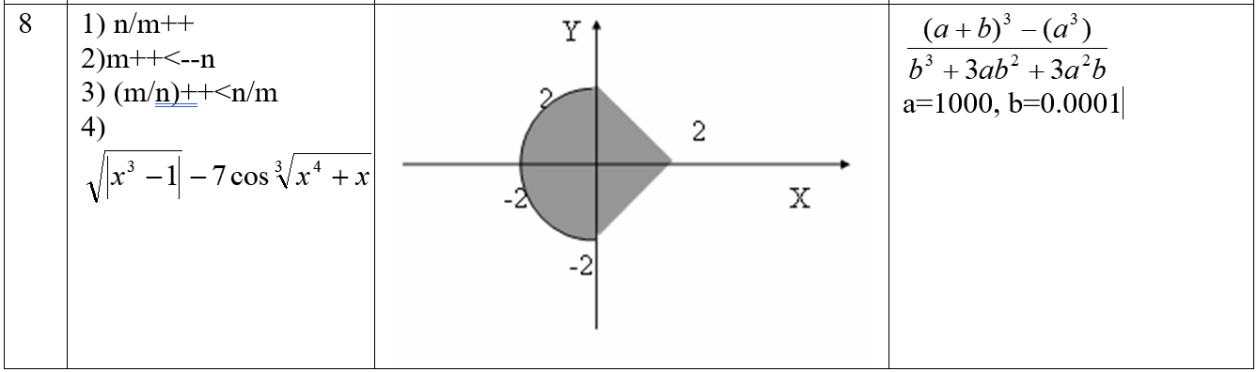
Доцент кафедры ИТАС

Викентьева О. Л.

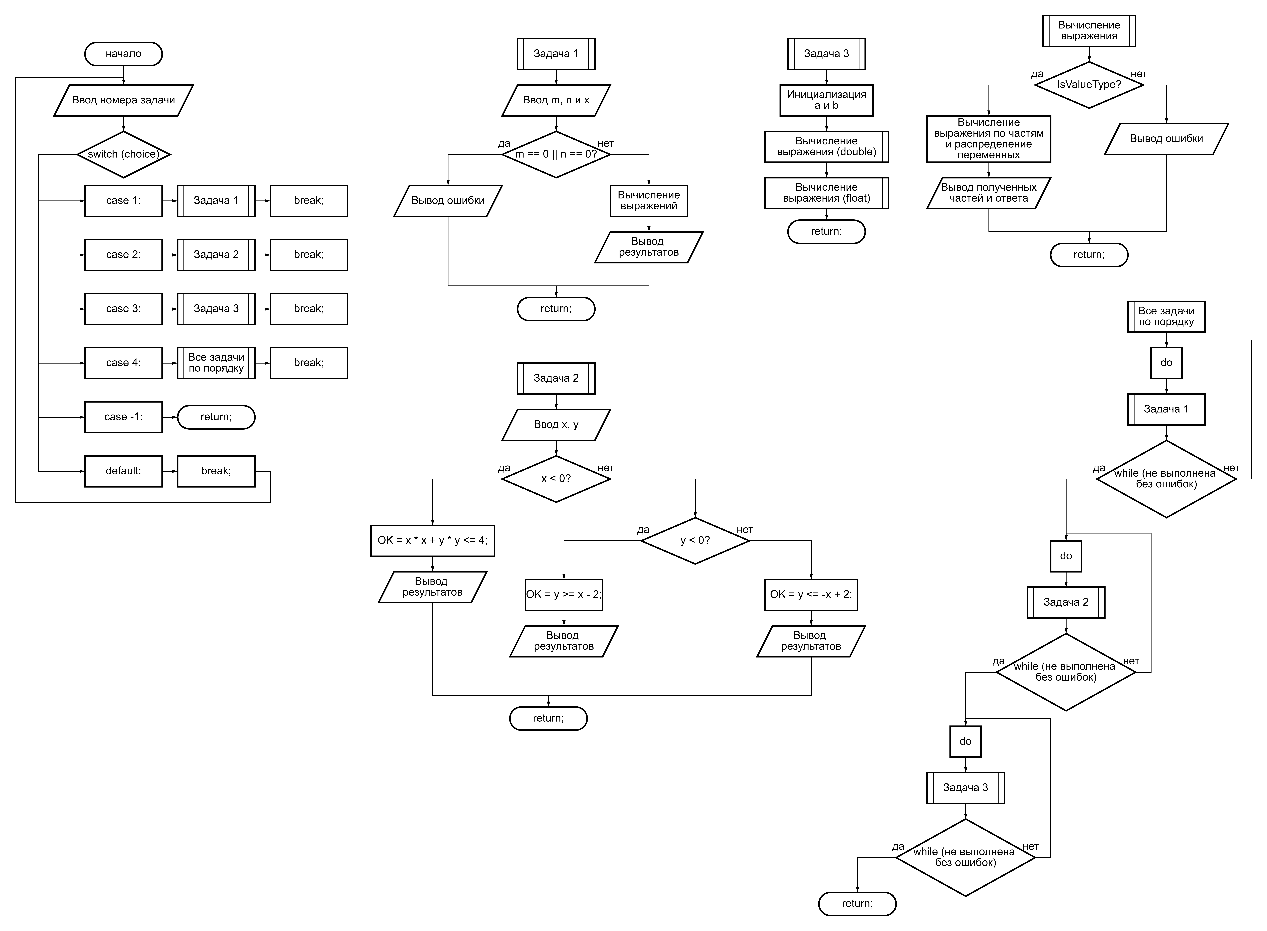
Пермь, 2024

**Постановка задачи**

1. Для задачи 1 найти значения выражений. Если необходимо, то определить при каких исходных данных выражение не может быть вычислено и выдать сообщение об ошибке.
2. Составить систему тестов и вычислить полученное выражение для нескольких значений Х.
3. Для задачи 2 записать выражение, зависящее от координат точки X1 и Y1 и принимающее значение TRUE, если точка принадлежит заштрихованной области, и FALSE, если не принадлежит.
4. Составить систему тестов и вычислить полученное выражение.
5. Для задачи 3 вычислить значение выражения, используя различные вещественные типы данных (float и double).
6. Результаты всех вычислений вывести на печать.
7. Объяснить полученные результаты.



**Блок-схема алгоритма**



**Работа программы**

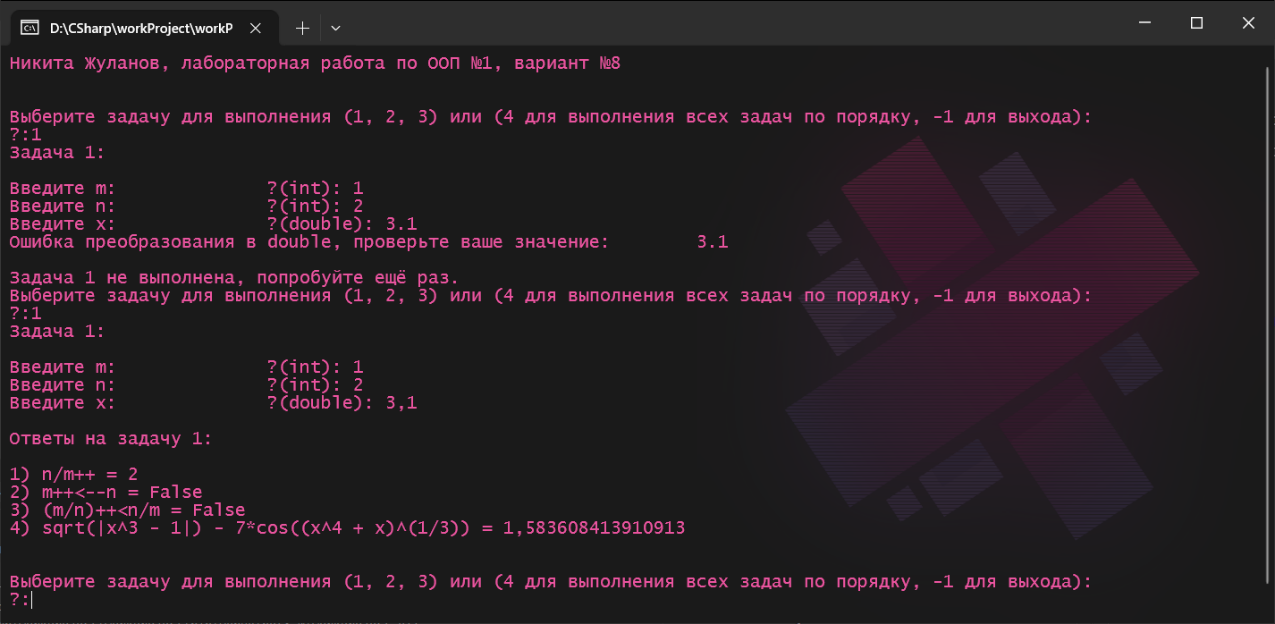
****

Рисунок 1 – Неудачное и успешное решение первого задания

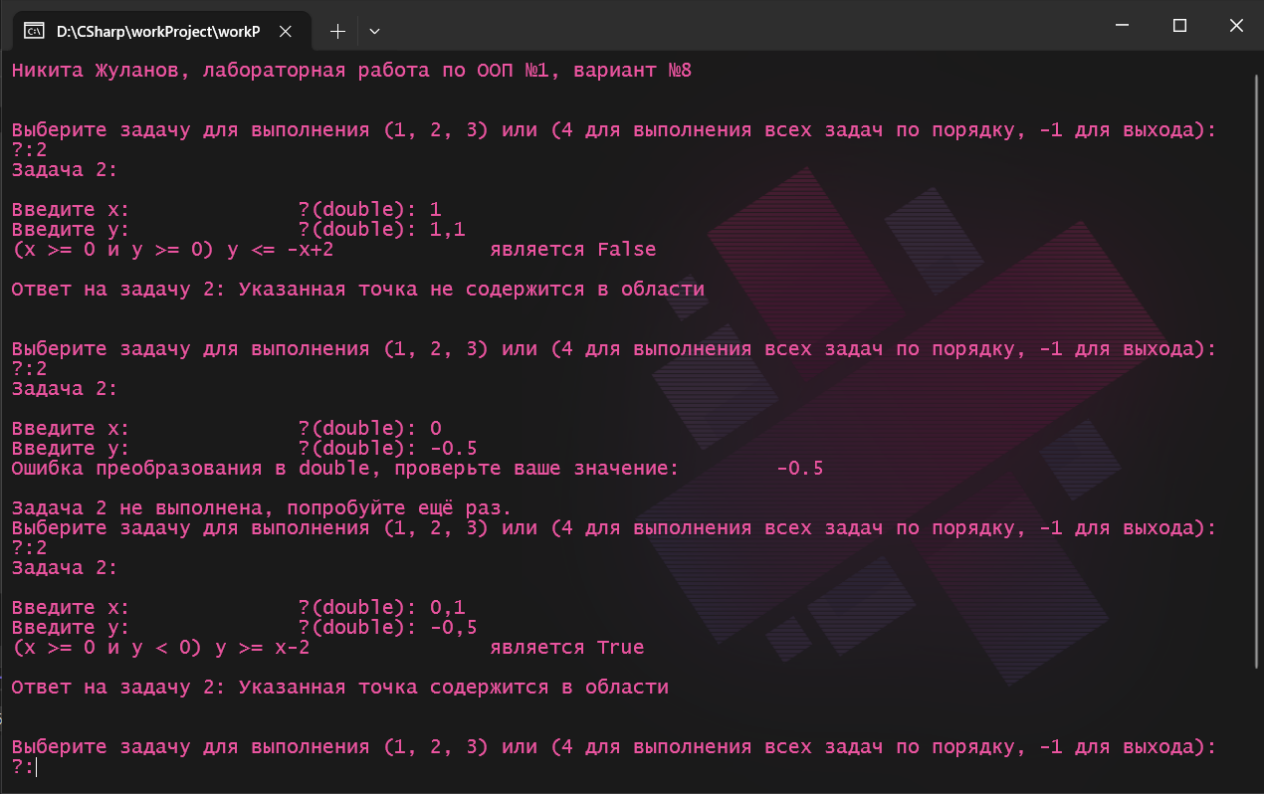


Рисунок 2 – Неудачное и успешные решения второго задания

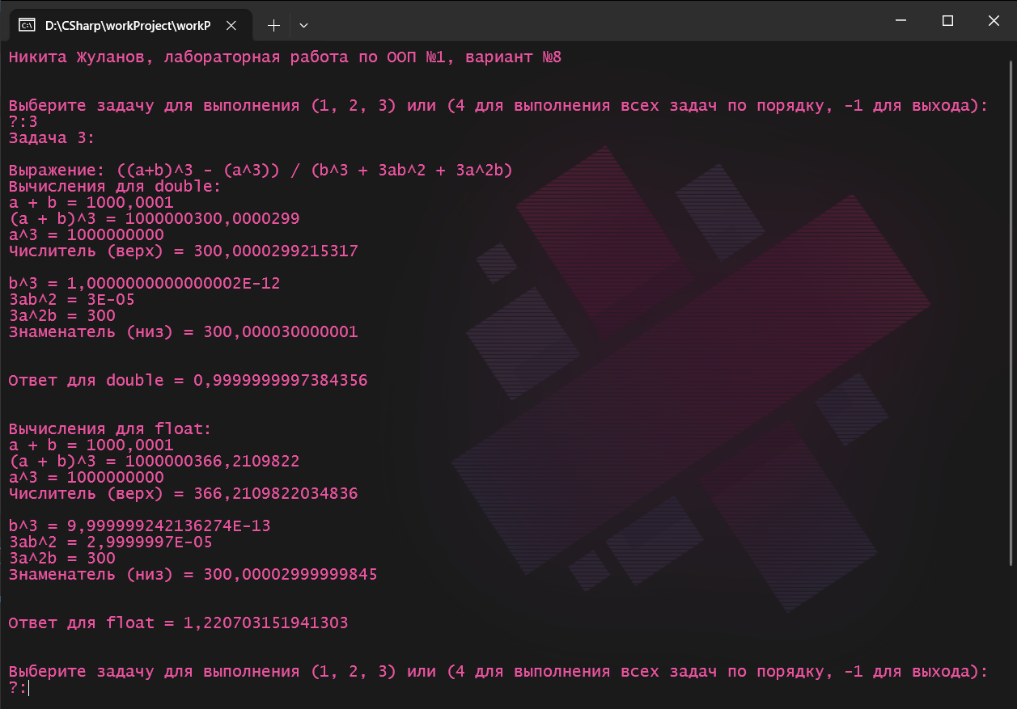


Рисунок 3 – Решение третьего задания

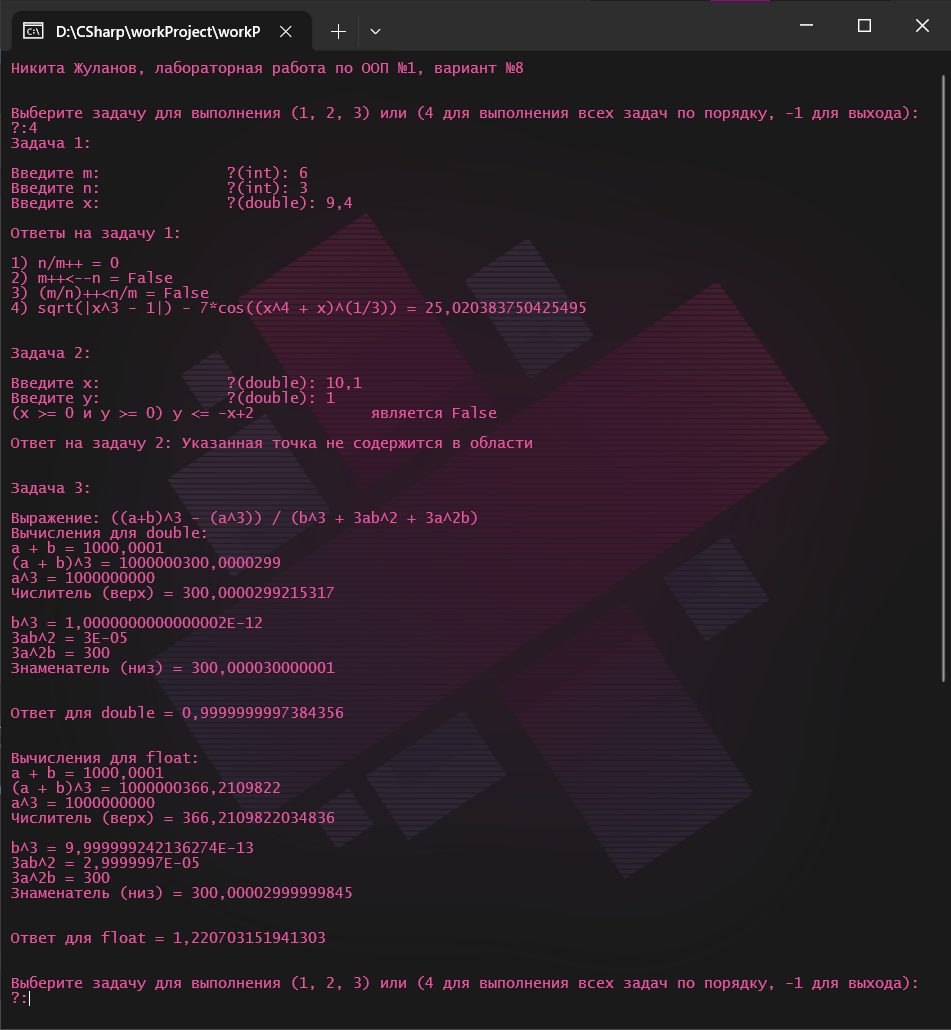


Рисунок 4 – Решение всех задач по порядку

**Тестирование**

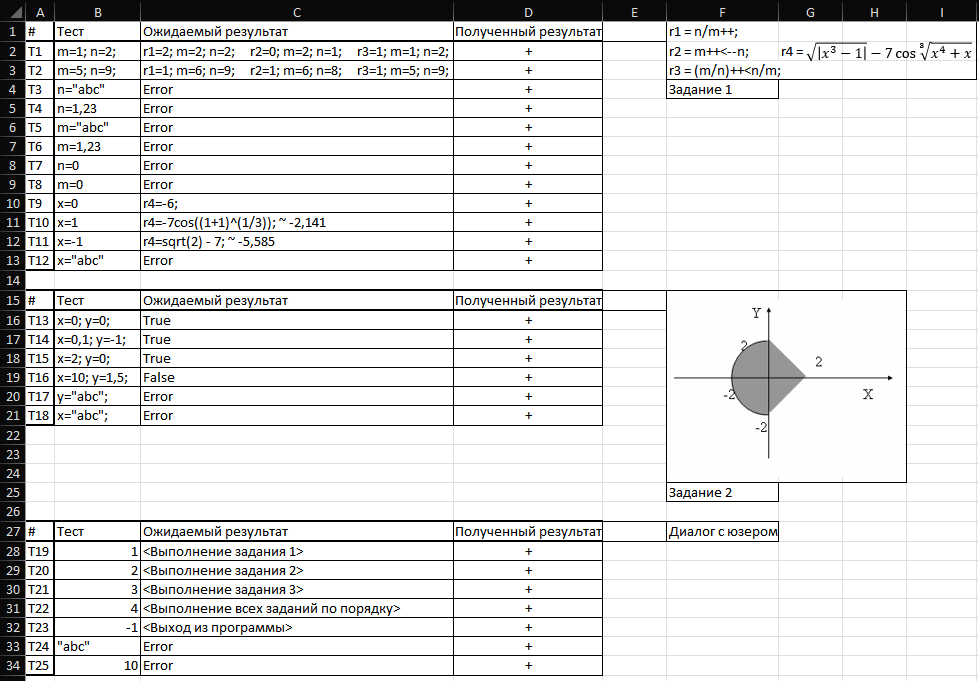
****

Рисунок 5 – Тесты для программы

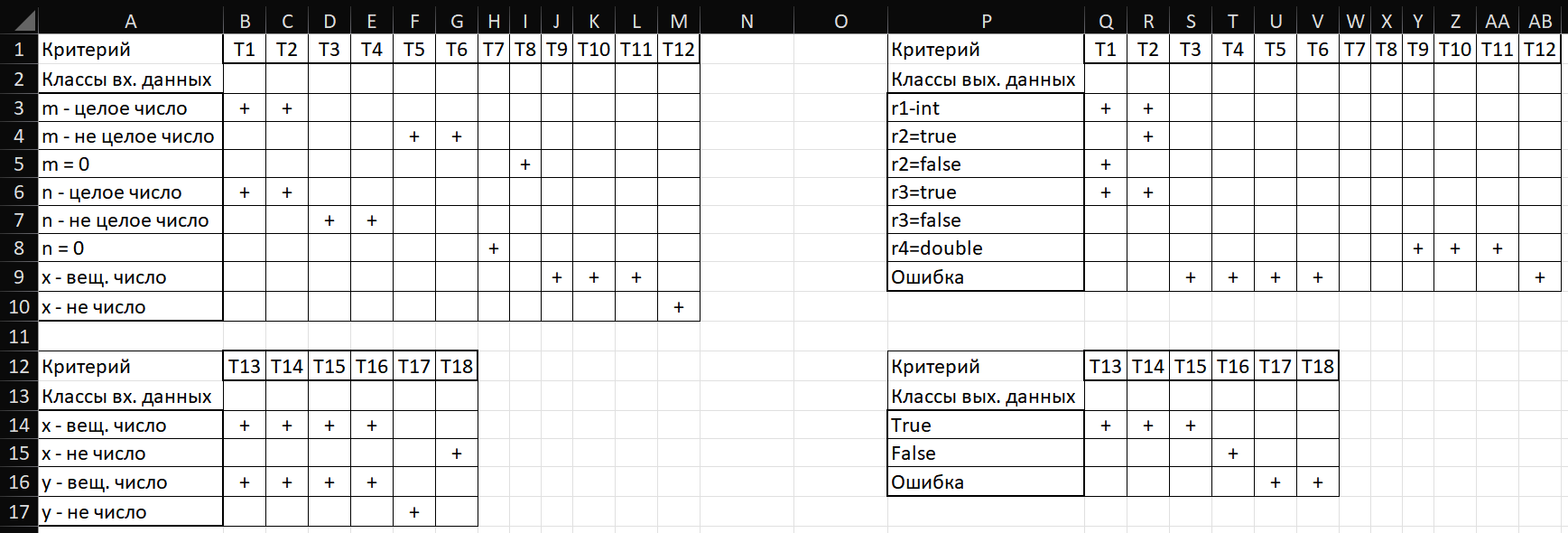
****

Рисунок 6 – Оценка тестов по критериям чёрного ящика

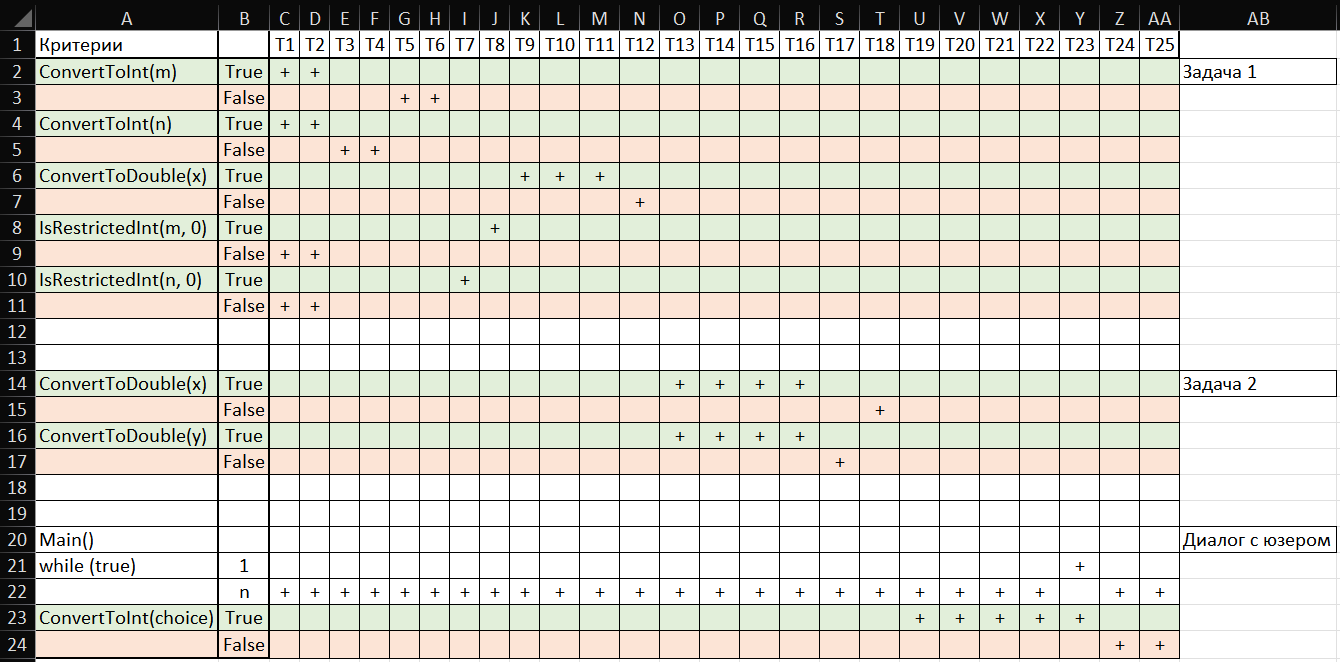


Рисунок 7 – Оценка тестов по критериям белого ящика

**Код программы**

using System;

using Lab1.Helpers;

using Lab1.Tasks;

/\*

Никита Жуланов, лабораторная работа по ООП №1, вариант №8

\*/

namespace Lab1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Никита Жуланов, лабораторная работа по ООП №1, вариант №8\n\n");

while (true)

{

Console.Write("Выберите задачу для выполнения (1, 2, 3) или (4 для выполнения всех задач по порядку, -1 для выхода):\n?:");

string input = Console.ReadLine();

if (Utils.ConvertToInt(input, out int choice))

{

switch (choice)

{

case 1:

if (!Task1.PerformTask1())

Console.WriteLine("Задача 1 не выполнена, попробуйте ещё раз.");

break;

case 2:

if (!Task2.PerformTask2())

Console.WriteLine("Задача 2 не выполнена, попробуйте ещё раз.");

break;

case 3:

if (!Task3.PerformTask3())

Console.WriteLine("Задача 3 не выполнена, попробуйте ещё раз.");

break;

case 4:

RunAllTasks();

break;

case -1:

Console.WriteLine("Выход из программы.");

return;

default:

Console.WriteLine("Неверный выбор, пожалуйста, введите допустимое число\n");

break;

}

}

}

}

static void RunAllTasks()

{

bool isTask1Successful = false,

isTask2Successful = false,

isTask3Successful = false;

do

{

isTask1Successful = Task1.PerformTask1();

if (!isTask1Successful)

Console.WriteLine("Задача 1 не выполнена, попробуйте ещё раз\n");

} while (!isTask1Successful);

do

{

isTask2Successful = Task2.PerformTask2();

if (!isTask2Successful)

Console.WriteLine("Задача 2 не выполнена, попробуйте ещё раз\n");

} while (!isTask2Successful);

do

{

isTask3Successful = Task3.PerformTask3();

if (!isTask3Successful)

Console.WriteLine("Задача 3 не выполнена, попробуйте ещё раз\n");

} while (!isTask3Successful);

}

}

}

using System;

namespace Lab1.Helpers

{

public static class Utils

{

public static bool ConvertToInt(string value, out int intValue)

{

bool isConverted = int.TryParse(value, out intValue);

if (!isConverted)

{

Console.WriteLine($"Ошибка преобразования в целое число, проверьте ваше значение:\t\t{value}\n");

}

return isConverted;

}

public static bool ConvertToDouble(string value, out double doubleValue)

{

bool isConverted = double.TryParse(value, out doubleValue);

if (!isConverted)

{

Console.WriteLine($"Ошибка преобразования в вещественное число с двойной точностью, проверьте ваше значение:\t{value}\n");

}

return isConverted;

}

public static bool IsRestrictedInt(int toCheck, int value)

{

if (toCheck == value) Console.WriteLine($"Запрещённое целочисленное значение:\t\t{toCheck}\n");

return toCheck == value;

}

public static string GetValueString(string name, string valueType = "int")

{

Console.Write($"Введите {name}:\t\t?({valueType}): ");

return Console.ReadLine();

}

}

}

using System;

using static System.Math;

using Lab1.Helpers;

namespace Lab1.Tasks

{

public static class Task1

{

public static bool PerformTask1()

{

Console.WriteLine("Задача 1:\n");

int m, n;

double x;

if (!Utils.ConvertToInt(Utils.GetValueString("m"), out m) ||

!Utils.ConvertToInt(Utils.GetValueString("n"), out n) ||

!Utils.ConvertToDouble(Utils.GetValueString("x", "double"), out x))

{

return false;

}

if (Utils.IsRestrictedInt(m, 0) || Utils.IsRestrictedInt(n, 0))

{

return false;

}

Console.WriteLine("\nОтветы на задачу 1:\n");

Console.WriteLine($"1) n/m++ = {n / m++} (m = {m}; n = {n})");

m--; // Коррекция значений, чтобы оставить изначальные m и n

Console.WriteLine($"2) m++<--n = {m++ < --n} (m = {m}; n = {n})");

m--; n++;

int intermediateValue = m / n;

Console.WriteLine($"3) (m/n)++<n/m = {intermediateValue++ < n / m} (m = {m}; n = {n})");

Console.WriteLine($"4) sqrt(|x^3 - 1|) - 7\*cos((x^4 + x)^(1/3)) = " +

$"{Sqrt(Abs(Pow(x, 3) - 1)) - 7 \* Cos(Pow(Pow(x, 4) + x, 1.0 / 3.0))}");

Console.WriteLine("\n");

return true;

}

}

}

using System;

using Lab1.Helpers;

namespace Lab1.Tasks

{

public static class Task2

{

public static bool PerformTask2()

{

Console.WriteLine("Задача 2:\n");

double x, y;

if (!Utils.ConvertToDouble(Utils.GetValueString("x", "double"), out x) ||

!Utils.ConvertToDouble(Utils.GetValueString("y", "double"), out y))

{

return false;

}

bool OK;

// OK = (x\*x + y\*y <= 4 && x < 0) || (y >= x-2 && x >= 0 && y <= 0) || (y <= -x+2 && x >= 0 && y > 0);

if (x < 0)

{

OK = x \* x + y \* y <= 4;

Console.WriteLine($"(x < 0) x^2 + y^2 <= 4\t\tявляется {OK}");

}

else

{

if (y < 0)

{

OK = y >= x - 2;

Console.WriteLine($"(x >= 0 и y < 0) y >= x-2\t\tявляется {OK}");

}

else

{

OK = y <= -x + 2;

Console.WriteLine($"(x >= 0 и y >= 0) y <= -x+2\t\tявляется {OK}");

}

}

Console.WriteLine($"\nОтвет на задачу 2: Указанная точка {(OK ? "" : "не ")}содержится в области");

Console.WriteLine("\n");

return true;

}

}

}

using System;

using static System.Math;

namespace Lab1.Tasks

{

public static class Task3

{

public static bool PerformTask3()

{

Console.WriteLine("Задача 3:\n");

Console.WriteLine($"Выражение: ((a+b)^3 - (a^3)) / (b^3 + 3ab^2 + 3a^2b)");

double aD = 1000.0, bD = 0.0001;

float aF = 1000f, bF = 0.0001f;

CalculateExpression(aD, bD, "double");

Console.WriteLine("\n");

CalculateExpression(aF, bF, "float");

Console.WriteLine("\n");

return true;

}

private static bool CalculateExpression<NumType>(NumType a, NumType b, string valueType)

{

if (!typeof(NumType).IsValueType)

{

Console.WriteLine("Принимаются только типы значений!");

return false;

}

Console.WriteLine($"Вычисления для {valueType}:");

dynamic aDyn = a;

dynamic bDyn = b;

// ((a+b)^3 - (a^3)) /

// (b^3 + 3ab^2 + 3a^2b)

dynamic top1 = Pow(aDyn + bDyn, 3);

dynamic top2 = Pow(aDyn, 3);

dynamic top = top1 - top2;

dynamic bottom1 = Pow(bDyn, 3);

dynamic bottom2 = 3 \* aDyn \* bDyn \* bDyn;

dynamic bottom3 = 3 \* aDyn \* aDyn \* bDyn;

dynamic bottom = bottom1 + bottom2 + bottom3;

dynamic result = top / bottom;

Console.WriteLine($"a + b = {aDyn + bDyn}\n(a + b)^3 = {top1}\na^3 = {top2}\nЧислитель (верх) = {top}\n");

Console.WriteLine($"b^3 = {bottom1}\n3ab^2 = {bottom2}\n3a^2b = {bottom3}\nЗнаменатель (низ) = {bottom}\n");

Console.WriteLine($"\nОтвет для {valueType} = {result}");

return true;

}

}

}