Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа №1

По дисциплине

“Объектно-ориентированное программирование”

# Тема “Выполнение программы простой структуры. Вычисление выражений с использованием стандартных функций”

Вариант 14

Выполнил работу

студент группы РИС-19-1б

Миннахметов Э.Ю.

Проверила

доцент кафедры ИТАС

Викентьева О.Л.

Работу выполнил:

Пермь 2020

**1 ПОСТАНОВКА ЗАДАЧИ**

1. Для задачи 1 определить тип заданных выражений и найти их значения;
2. Составить систему тестов и вычислить полученное выражение для нескольких значений Х, определить при каких Х выражение не может быть вычислено;
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 для нескольких точек, принадлежащих и не принадлежащих заштрихованной области;
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.

|  |  |  |
| --- | --- | --- |
| Задача 1 | Задача 2 | Задача 3 |
| 1) n++\*--m  2) n--<m++  3) --n>--m  4) |  | а=10, b=0.01 |

**Требуемые формулы**

* **–** определяет множество точек, находящихся внутри круга или на его границе.

**2 ИСХОДНЫЕ КОДЫ ПРОГРАММ**

**2.1 Задача 1**

using System;

namespace Task1

{

delegate void Example();

class Program

{

private const char c\_cN = 'n';

private const char c\_cM = 'm';

private const char c\_cX = 'x';

private const string c\_sGetNumber = "Введите число {0}: ";

private const string c\_sExample = "ПРИМЕР {0}";

private const string c\_sExpression = "Выражение: {0}";

private const string c\_sExpressionType = "Тип выражения: {0}";

private const string c\_sExpressionValue = "Значение выражения: {0}";

private const string c\_sVariable = "Переменная {0}: {1}";

private const int EXAMPLES\_COUNT = 4;

private static readonly Example[] examples =

{ Example1, Example2, Example3, Example4 };

public static void Main(string[] args)

{

for (int i = 0; i < EXAMPLES\_COUNT; ++i)

{

Console.WriteLine(c\_sExample, i + 1);

examples[i]();

Console.WriteLine();

}

Console.ReadKey();

}

private static void Example1()

{

GetNum(out int n, out int m);

Console.WriteLine(c\_sExpression, "n++ \* --m");

Expression(n++ \* --m);

WriteNM(n, m);

}

private static void Example2()

{

GetNum(out int n, out int m);

Console.WriteLine(c\_sExpression, "n-- < m++");

Expression(n-- < m++);

WriteNM(n, m);

}

private static void Example3()

{

GetNum(out int n, out int m);

Console.WriteLine(c\_sExpression, "--n < --m");

Expression(--n < --m);

WriteNM(n, m);

}

private static void Example4()

{

int x = 0;

for (bool flag = false; !flag;)

{

GetNum(out x, c\_cX);

flag = x != 0;

}

Console.WriteLine(c\_sExpression, "|x + 1|^(1/4) + 1 / x^2");

Expression(Math.Pow(Math.Abs(x + 1), 1 / 4.0) + 1 / Math.Pow(x, 2));

WriteNum(x, c\_cX);

}

private static void GetNum(out int number, char simbol)

{

number = 0;

for(bool flag = false; !flag;)

{

Console.Write(c\_sGetNumber, simbol);

string sNum = Console.ReadLine();

flag = int.TryParse(sNum, out number);

}

}

private static void GetNum(out int n, out int m)

{

GetNum(out n, c\_cN);

GetNum(out m, c\_cM);

}

private static void Expression<T>(T result)

{

Console.WriteLine(c\_sExpressionType, typeof(T).Name);

Console.WriteLine(c\_sExpressionValue, result);

}

private static void WriteNum(int number, char simbol)

{

Console.WriteLine(c\_sVariable, simbol, number);

}

private static void WriteNM(int n, int m)

{

WriteNum(n, c\_cN);

WriteNum(m, c\_cM);

}

}

}

**2.2 Задача 2**

using System;

namespace Task2

{

class Program

{

private const char c\_cX = 'x';

private const char c\_cY = 'y';

private const char c\_cD = 'д';

private const string c\_sPointInfo = "Точка M({0}, {1})";

private const string c\_sNot = " не";

private const string c\_sIsEntered = " входит в область фигуры";

private const string c\_sContinue = "Продолжить? (д - да, иначе - нет)";

private const string c\_sGetNumber = "Введите число {0}: ";

static void Main(string[] args)

{

do {

RunCalculate(GetNum(c\_cX), GetNum(c\_cY));

Console.Write(c\_sContinue);

} while (Console.ReadLine()[0] == c\_cD);

Console.WriteLine();

TestSystem();

Console.ReadKey();

}

private static void TestSystem()

{

RunCalculate(0.0, 0.1);

RunCalculate(0.2, -0.3);

RunCalculate(0.4, 0.5);

RunCalculate(0.6, -0.7);

RunCalculate(0.8, 0.9);

RunCalculate(1.0, -1.1);

RunCalculate(1.2, 1.3);

RunCalculate(1.4, -1.5);

RunCalculate(0.0, -0.1);

RunCalculate(-0.2, 0.3);

RunCalculate(0.4, -0.5);

RunCalculate(-0.6, 0.7);

RunCalculate(0.8, -0.9);

RunCalculate(-1.0, 1.1);

RunCalculate(1.2, -1.3);

RunCalculate(-1.4, 1.5);

}

private static void RunCalculate(double x, double y)

{

Console.Write(c\_sPointInfo, x, y);

if (Calculate(x, y))

Console.Write(c\_sNot);

Console.WriteLine(c\_sIsEntered);

}

private static bool Calculate(double x, double y)

{

return !(RightTop(x, y)

|| LeftTop(x, y)

|| LeftBottom(x, y)

|| RightBottom(x, y));

}

private static bool RightTop(double x, double y)

{

return x >= 0

&& y >= 0

&& x <= 1

&& y <= 1

&& y - x >= 0;

}

private static bool LeftTop(double x, double y)

{

return x <= 0

&& y >= 0

&& Math.Pow(Math.Pow(x,2)+ Math.Pow(y,2),1/2.0) <= 1;

}

private static bool LeftBottom(double x, double y)

{

return x <= 0

&& y <= 0

&& x >= -1

&& y >= -1

&& x - y >= 0;

}

private static bool RightBottom(double x, double y)

{

return x >= 0

&& y <= 0

&& Math.Pow(Math.Pow(x, 2) + Math.Pow(y, 2), 1 / 2.0) <= 1;

}

private static double GetNum(char simbol)

{

double number = 0;

for (bool flag = false; !flag;)

{

Console.Write(c\_sGetNumber, simbol);

string sNum = Console.ReadLine();

flag = double.TryParse(sNum, out number);

}

return number;

}

}

}

**2.3 Задача 3**

using System;

namespace Task3

{

class Program

{

private const char c\_cA = 'a';

private const char c\_cB = 'b';

private const double c\_dA = 10d;

private const double c\_dB = 0.1d;

private const double c\_fA = 10f;

private const double c\_fB = 0.1f;

private const string c\_sDouble = "double";

private const string c\_sFloat = "float";

private const string c\_sGetNumber = "Введите число {0} ({1}): ";

private const string c\_sProgramText =

"Операция с float:\n{0}\n\nОперация с double:\n{1}\n";

static void Main(string[] args)

{

Console.WriteLine(c\_sProgramText,

Run(c\_dA, c\_dB), Run(c\_fA, c\_fB));

Console.WriteLine("\n" + c\_sProgramText,

Run(GetDouble(c\_cA), GetDouble(c\_cB)),

Run(GetFloat(c\_cA), GetFloat(c\_cB)));

Console.ReadKey();

}

private static double Run(double a, double b)

{

return (Pow(a + b, 4) - (Pow(a, 4) + 6 \* Pow(a, 2) \* Pow(b, 2) + Pow(b, 4)))

/ (4 \* a \* Pow(b, 3) + 4 \* b \* Pow(a, 3));

}

private static float Run(float a, float b)

{

return (Pow(a + b, 4) - (Pow(a, 4) + 6 \* Pow(a, 2) \* Pow(b, 2) + Pow(b, 4)))

/ (4 \* a \* Pow(b, 3) + 4 \* b \* Pow(a, 3));

}

private static double Pow(double a, int b)

{

while (--b > 0)

a \*= a;

return a;

}

private static float Pow(float a, int b)

{

while (--b > 0)

a \*= a;

return a;

}

private static double GetFloat(char simbol)

{

float number = 0;

for (bool flag = false; !flag;)

{

Console.Write(c\_sGetNumber, simbol, c\_sFloat);

string sNum = Console.ReadLine();

flag = float.TryParse(sNum, out number);

}

return number;

}

private static double GetDouble(char simbol)

{

double number = 0;

for (bool flag = false; !flag;)

{

Console.Write(c\_sGetNumber, simbol, c\_sDouble);

string sNum = Console.ReadLine();

flag = double.TryParse(sNum, out number);

}

return number;

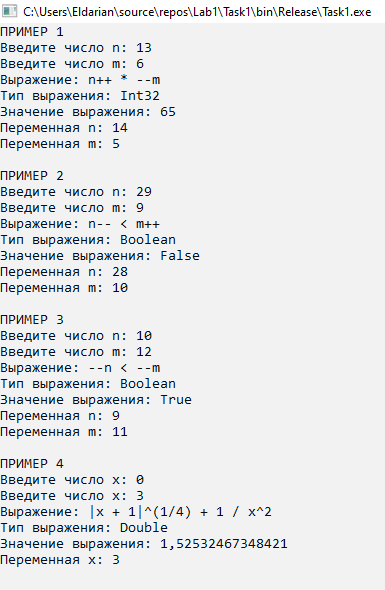
}

}

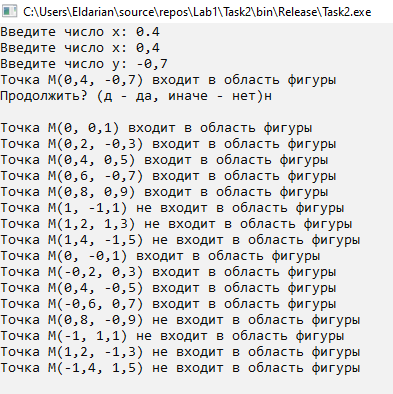
}

**3 СКРИНШОТЫ ВЫПОЛНЕНИЯ ПРОГРАММ**

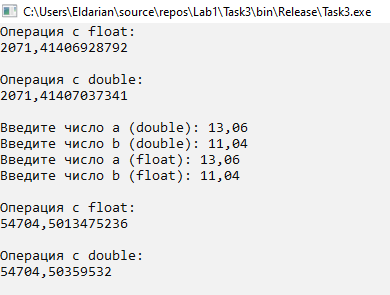
**3.1 Задача 1**



**3.2 Задача 2**



**3.3 Задача 3**



**4 АНАЛИЗ РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММ**