Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа 1**

**«Выполнение простой программы. Типы данных. Организация ввода и вывода данных»**

Выполнил:   
студент группы РИС-23-1б   
Мокрушин Никита Дмитриевич

Проверила:   
доцент кафедры ИТАС   
О.Л.Викентьева

Пермь 2024 г.

**Постановка задачи:**

**Цель:**

* Реализовать вычисление математических выражений и проверку условий, используя целые и вещественные типы данных.
* Разработать алгоритмы, которые позволяют определить, при каких исходных данных выражение не может быть вычислено, и вывести соответствующее сообщение об ошибке.
* Написать программу, которая проверяет принадлежность точки заданной области.
* Вычислить выражения с использованием различных вещественных типов данных (float и double) и проанализировать результаты.

**Требования:**

* Программа должна принимать ввод данных с клавиатуры.
* Для каждой задачи разработать систему тестов и проанализировать результаты вычислений.
* При проверке принадлежности точки области использовать переменную логического типа.
* При работе с выражениями использовать условные операторы для проверки возможности выполнения вычислений.

**Результаты:**

* Вывести результаты всех вычислений на экран.
* Объяснить результаты работы программы и провести анализ тестов.

**Методы:**

* Использовать вспомогательные переменные для хранения промежуточных значений.
* Выполнить приведение типов данных при работе с вещественными числами.
* Проанализировать достаточность тестов по критериям черного и белого ящика.

**Постановка задачи для варианта №14:**

**Задача 1:**

* Найти значения следующих выражений:
  1. n++ \* --m
  2. n-- < m++
  3. --n > --m
* Определить, при каких исходных данных выражение не может быть вычислено. Если такие данные присутствуют, выдать сообщение об ошибке.

**Задача 2:**

* Записать выражение, зависящее от координат точки X1​ и Y1​, которое возвращает TRUE, если точка принадлежит заштрихованной области, и FALSE, если не принадлежит.
* Составить и выполнить систему тестов для проверки данного выражения с разными значениями координат.

**Задача 3:**

* Для значений a = 1000 и b = 0.001:
  + Вычислить значение выражения с использованием различных вещественных типов данных (float и double).
* Использовать вспомогательные переменные для хранения промежуточных значений и выполнить приведение типов данных, если это необходимо.

**Результаты и вывод:**

* Все результаты вычислений вывести на экран.
* Объяснить результаты, включая возможные ограничения и погрешности, связанные с использованием различных типов данных.

**Анализ входных и выходных данных:**

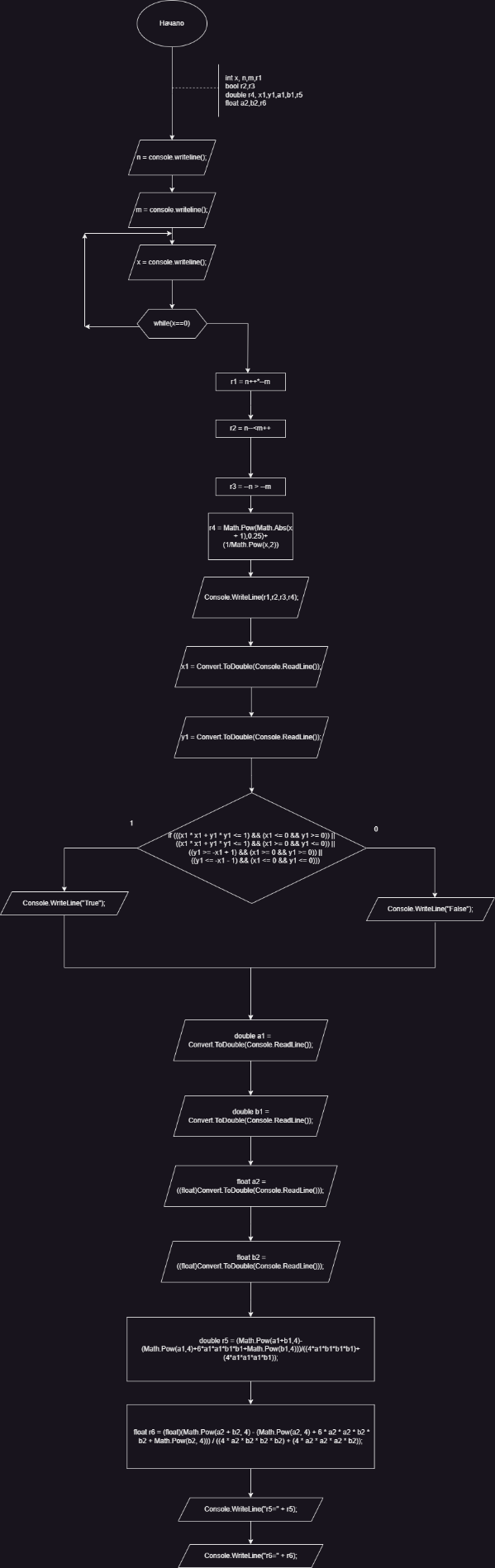
**Входные данные:**

1. n, m, x:
   * n и m: Целые числа.
   * x: Целое число (не может быть 0).
2. **x1, y1:**
   * Два вещественных числа (Double), представляющие координаты точки.
3. **a и b:**
   * a1, b1: Два вещественных числа (Double).
   * a2, b2: Два вещественных числа (Float).

**Выходные данные:**

1. **Результаты первой задачи:**
   * r1: Результат выражения (n++) \* (--m). Значение m и n после выполнения операции, а также результат выражения.
   * r2: Логический результат выражения (n--) < (m++). Значения m и n после выполнения операции, а также результат выражения.
   * r3: Логический результат выражения (--n) > (--m). Значения m и n после выполнения операции, а также результат выражения.
   * r4: Значение выражения Math.Pow(Math.Abs(x + 1), 0.25) + (1 / Math.Pow(x, 2)).
2. **Результат второй задачи:**
   * Логическое значение True или False, которое указывает, принадлежит ли точка (x1, y1) заданной области.
3. **Результаты третьей задачи:**
   * r5: Результат вычисления выражения (Math.Pow(a1 + b1, 4) - (Math.Pow(a1, 4) + 6 \* a1 \* a1 \* b1 \* b1 + Math.Pow(b1, 4))) / ((4 \* a1 \* b1 \* b1 \* b1) + (4 \* a1 \* a1 \* a1 \* b1)) для вещественных чисел типа Double.
   * r6: Результат вычисления аналогичного выражения, но для вещественных чисел типа Float.

**Блок-Схема:**



**Код программы:**

using System;

internal class Program

{

static void Main(string[] args)

{

int n = 0, m = 0;

double x = 0;

bool isValid;

isValid = false;

while (!isValid)

{

Console.WriteLine("Введите n:");

if (int.TryParse(Console.ReadLine(), out n))

{

isValid = true;

}

else

{

Console.WriteLine("Ошибка: введите корректное целое число.");

}

}

isValid = false;

while (!isValid)

{

Console.WriteLine("Введите m:");

if (int.TryParse(Console.ReadLine(), out m))

{

isValid = true;

}

else

{

Console.WriteLine("Ошибка: введите корректное целое число.");

}

}

do

{

Console.WriteLine("Введите x:");

if (!double.TryParse(Console.ReadLine(), out x))

{

Console.WriteLine("Ошибка: введите корректное целое число.");

continue;

}

if (x == 0)

Console.WriteLine("Недопустимое значение x, попробуйте снова");

} while (x == 0);

Console.WriteLine("Результат первой задачи:");

Console.WriteLine("Переменные до изменений:");

Console.WriteLine("m=" + m);

Console.WriteLine("n=" + n);

int r1 = (n++) \* (--m);

Console.WriteLine("1) n++ \* --m=" + r1);

Console.WriteLine("Переменные после изменений:");

Console.WriteLine("m=" + m);

Console.WriteLine("n=" + n);

bool r2, r3;

Console.WriteLine("Переменные до изменений:");

Console.WriteLine("m=" + m);

Console.WriteLine("n=" + n);

r2 = (n--) < (m++);

Console.WriteLine("2) n-- < m++=" + r2);

Console.WriteLine("Переменные после изменений:");

Console.WriteLine("m=" + m);

Console.WriteLine("n=" + n);

Console.WriteLine("Переменные до изменений:");

Console.WriteLine("m=" + m);

Console.WriteLine("n=" + n);

r3 = (--n) > (--m);

Console.WriteLine("3) --n > --m=" + r3);

Console.WriteLine("Переменные после изменений:");

Console.WriteLine("m=" + m);

Console.WriteLine("n=" + n);

double r4 = Math.Pow(Math.Abs(x + 1), 0.25) + (1 / Math.Pow(x, 2));

Console.WriteLine("Math.Pow(Math.Abs(x + 1), 0.25) + (1 / Math.Pow(x, 2)) = " + r4);

Console.WriteLine("x=" + x);

double x1 = 0, y1 = 0;

isValid = false;

while (!isValid)

{

Console.WriteLine("Введите x1 и y1:");

if (double.TryParse(Console.ReadLine(), out x1) && double.TryParse(Console.ReadLine(), out y1))

{

isValid = true;

}

else

{

Console.WriteLine("Ошибка: введите корректные числа типа double.");

}

}

Console.WriteLine("Результат второй задачи:");

if (((x1 \* x1 + y1 \* y1 <= 1) && (x1 <= 0 && y1 >= 0)) ||

((x1 \* x1 + y1 \* y1 <= 1) && (x1 >= 0 && y1 <= 0)) ||

((y1 == -x1 + 1) && (x1 >= 0 && y1 >= 0)) ||

((y1 == -x1 - 1) && (x1 <= 0 && y1 <= 0)))

{

Console.WriteLine("True");

}

else

{

Console.WriteLine("False");

}

double a1 = 0, b1 = 0;

isValid = false;

while (!isValid)

{

Console.WriteLine("Введите a и b типа double:");

if (double.TryParse(Console.ReadLine(), out a1) && double.TryParse(Console.ReadLine(), out b1))

{

isValid = true;

}

else

{

Console.WriteLine("Ошибка: введите корректные числа типа double.");

}

}

float a2 = 0, b2 = 0;

isValid = false;

while (!isValid)

{

Console.WriteLine("Введите a и b типа float:");

if (float.TryParse(Console.ReadLine(), out a2) && float.TryParse(Console.ReadLine(), out b2))

{

isValid = true;

}

else

{

Console.WriteLine("Ошибка: введите корректные числа типа float.");

}

}

double c1 = Math.Pow(a1 + b1, 4);

double d1 = Math.Pow(a1, 4) + 6 \* a1 \* a1 \* b1 \* b1 + Math.Pow(b1, 4);

double e1 = (4 \* a1 \* b1 \* b1 \* b1) + (4 \* a1 \* a1 \* a1 \* b1);

float c2 = (float)Math.Pow(a2 + b2, 4);

float d2 = (float)(Math.Pow(a2, 4) + 6 \* a2 \* a2 \* b2 \* b2 + Math.Pow(b2, 4));

float e2 = (float)((4 \* a2 \* b2 \* b2 \* b2) + (4 \* a2 \* a2 \* a2 \* b2));

double r5 = (c1 - d1) / e1;

float r6 = (c2 - d2) / e2;

Console.WriteLine("Результат третьей задачи:");

Console.WriteLine("c1=" + c1);

Console.WriteLine("d1=" + d1);

Console.WriteLine("e1=" + e1);

Console.WriteLine("c2=" + c2);

Console.WriteLine("d2=" + d2);

Console.WriteLine("e2=" + e2);

Console.WriteLine("r5=" + r5);

Console.WriteLine("r6=" + r6);

}

}

**Тесты:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Тест | Ожид. результат | Полученный результат |  | № | Тест | Ожид. результат | Полученный результат |
| 1 | m = 1; n = 2 | r1 = 3, n = 2, m = 2, r2 = false, m = 1, n = 1, r3 = true, m = 1, n = 1 | + |  | 1 | x1 = 0, y1 = 0 | True | + |
| 2 | m = 3; n = 5 | r1 = 8, m = 5, n = 4, r2 = true, m = 4, n = 3, r3 = false, m = 4, n = 3 | + |  | 2 | x1 = -1, y1 = 0 | True | + |
| 3 | m = 'abc' | Ошибка ввода | + |  | 3 | x1 = 1, y1 = 0 | True | + |
| 4 | m = 1.23 | Ошибка ввода | + |  | 4 | x1 = 0, y1 = 1 | True | + |
| 5 | n = 'abc' | Ошибка ввода | + |  | 5 | x1 = 0, y1 = -1 | True | + |
| 6 | n = 1.23 | Ошибка ввода | + |  | 6 | x1 = 0.5, y1 = 0.5 | True | + |
| 7 | x = 0 | Сообщение об ошибке "Недопустимое значение x" | + |  | 7 | x1 = -1, y1 = 1 | False | + |
| 8 | x = 1 | r4 = 1.8414 | + |  | 8 | x1 = "abc", y1 = 1 | Ошибка ввода | + |
| 9 | x = 'abc' | Ошибка ввода | + |  | 9 | x1 = 1, y1 = "abc" | Ошибка ввода | + |
|  |  |  |  |  | 10 | x1 = 0, y1 = -0.5 | True | + |
|  |  |  |  |  | 11 | x1 = 1, y1 = 1 | False | + |

**Анализ достаточности тестов:  
Черный ящик:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Критерий** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| Классы входных данных |  |  |  |  |  |  |  |  |  |
| m целое число | + | + |  |  |  |  | + | + |  |
| m не целое число |  |  | + | + |  |  |  |  |  |
| n целое число | + | + |  |  |  |  | + | + |  |
| n не целое число |  |  |  |  | + | + |  |  |  |
| x != 0 | + | + | + | + | + | + | + |  |  |
| x = 0 |  |  |  |  |  |  |  | + |  |
| Классы выходных данных |  |  |  |  |  |  |  |  |  |
| r1 целое число | + | + |  |  |  |  | + |  |  |
| r2 true | + |  |  |  |  |  | + |  |  |
| r2 false |  | + |  |  |  |  |  | + |  |
| r3 false | + | + |  |  |  |  | + | + |  |
| r3 true |  |  |  |  |  |  |  |  | + |
| r4 число вещественное | + | + |  |  |  |  | + | + |  |
| Ошибка |  |  | + | + | + | + |  |  | + |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Критерий** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |  | **11** |
| Классы входных данных |  |  |  |  |  |  |  |  |  |  |  |  |
| x1 целое число | + |  | + |  |  |  |  |  |  |  |  |  |
| x1 не целое число |  | + |  |  |  |  |  |  |  |  |  |  |
| y1 целое число | + |  |  | + |  |  |  |  |  |  |  |  |
| y1 не целое число |  |  |  |  | + |  |  |  |  |  |  |  |
| Классы выходных данных |  |  |  |  |  |  |  |  |  |  |  |  |
| Верный результат True | + |  |  |  | + |  |  |  |  |  |  |  |
| Верный результат False |  | + |  |  |  | + | + |  |  |  |  |  |
| Ошибка ввода |  |  | + | + |  |  |  |  | + | + |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий | Входные данные 1 (2, 3, 1) | Входные данные 2 (-1, 4, -1) | Входные данные 3 (0, 0, 2) | Входные данные 4 (5, -3, 3) | Входные данные 5 (2, 3, 0) |
| Ввод данных n, m, x | 2, 3, 1 | -1, 4, -1 | 0, 0, 2 | 5, -3, 3 | 2, 3, 0 |
| 1. Инкремент n++ и вычисление | + | + | + | + | + |
| Ожидаемый результат r1 | 5 | -2 | 0 | -18 | Повтор ввода x |
| Значение n после выполнения | 3 | 0 | 1 | 6 | - |
| Значение m после выполнения | 2 | 3 | -1 | -4 | - |
| 2. Цикл while (x == 0) | - | - | - | - | Повтор ввода |
| Ожидаемый результат | - | - | - | - | Повтор ввода x |
| 3. Сравнение (m--) > n | + | + | + | + | + |
| Ожидаемый результат r2 | false | true | false | false | - |
| Значение n после выполнения | 3 | 0 | 1 | 6 | - |
| Значение m после выполнения | 3 | 4 | 0 | -3 | - |
| 4. Сравнение (n--) > m | + | + | + | + | - |
| Ожидаемый результат r3 | true | false | true | true | - |
| Значение n после выполнения | 2 | -1 | 0 | 5 | - |
| Значение m после выполнения | 2 | 3 | 0 | -4 | - |
| 5. Вычисление Math.Pow(Math.Abs(x + 1),0.25) + (1/Math.Pow(x,2)) | + | + | + | + | - |
| Ожидаемый результат r4 | 1,84 | 1,84 | 1,48 | 1,11 | - |
| Значение x после выполнения | 1 | -1 | 2 | 3 | - |

**Белый Ящик:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий | Тестовые данные 1 (0, 0) | Тестовые данные 2 (1, 1) | Тестовые данные 3 (0.5, -0.5) | Тестовые данные 4 (0, -1) | Тестовые данные 5 (1, "abc") |
| Ввод данных x1, y1 | 0, 0 | 1, 1 | 0.5, -0.5 | 0, -1 | 1, "abc" |
| 6. Проверка условия if | + | + | + | + | Ошибка ввода |
| Ожидаемый результат | True | False | True | False | Ошибка |
| 7. Проверка условия else | False | True | False | True | - |
| Ожидаемый результат | - | - | - | - | - |

**Объяснение результатов:**

Программа позволяет увидеть влияние порядка выполнения операций, проверку условий, а также различия в точности вычислений между типами данных. Она также демонстрирует, как выполнять контроль ошибок и анализировать результаты в математических и логических задачах.