Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 5

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Условный оператор»

Выполнил: ст. гр. ТУУ-111

Попов В.А.

Вариант №3

\_\_\_\_\_\_\_\_\_\_\_

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_\_\_\_\_\_\_\_\_\_

(дата приёмки)

Москва – 2025 г.

1. **Цель**

Целью данной работы является освоение применения условных операторов для обработки ввода данных, а также использование метода TryParse для проверки корректности ввода.

1. **Формулировка задачи**

Необходимо реализовать контроль ввода исходных данных. В случае некорректного ввода хотя бы одного из элементов, входящих в множество исходных данных, программа должна уведомить пользователя об ошибке и прекратить выполнение вычислений. Для проверки корректности ввода следует использовать различные состояния метода TryParse, доступного для значимых (valuable) типов данных, таких как int, byte, float, double и другие.

* 1. **Задание 1**

Найти значение Z, учитывая все возможные ограничения, накладываемые не только на переменные, но и на участвующие в расчёте функциональные зависимости.

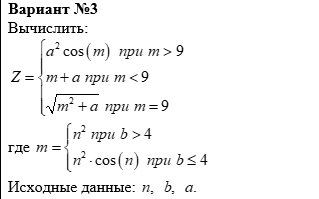


Рисунок 2.1.1 – Вариант задания

* 1. **Задание 2**

Ввести два символа в отдельные переменные. Если первый символ в таблице ASCII имеет больший порядковый номер, чем второй, выполнить обмен значений между переменными. В противном случае оставить их без изменений.

1. **Блок-схема**
   1. **Задание 1**

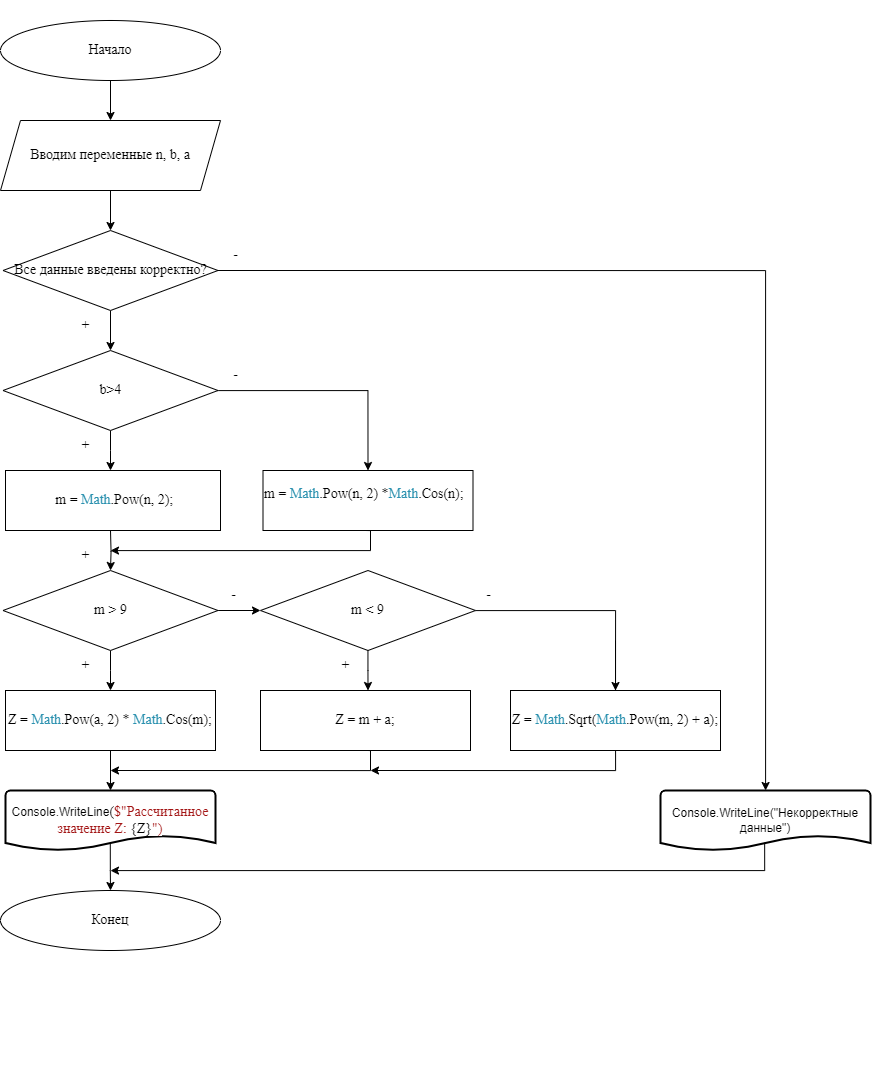


Рисунок 3.1.1 – Блок-схема алгоритма

* 1. **Задание 2**

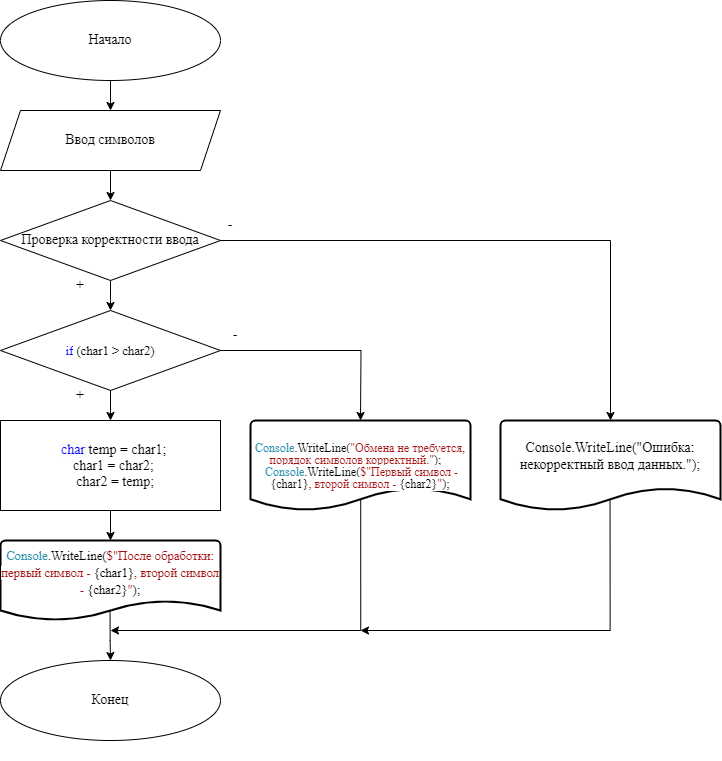


Рисунок 3.2.1 – Блок-схема алгоритма

1. **Подбор тестовых примеров**
   1. **Задание 1**
2. Входные данные:

n = 2, b = 5, a = 3.

Расчёт:

b > 4 → m = 2² = 4.

m < 9 → Z = m + a = 4 + 3 = 7.

Ожидаемый результат:

"Рассчитанное значение Z: 7"

1. Входные данные:

n = 3, b = 3, a = 4.

Расчёт:

b ≤ 4 → m = 3² \* cos(3) ≈ 9 \* (-0.99) ≈ -8.91 (примерное значение, cos(3)

≈ -0.99)

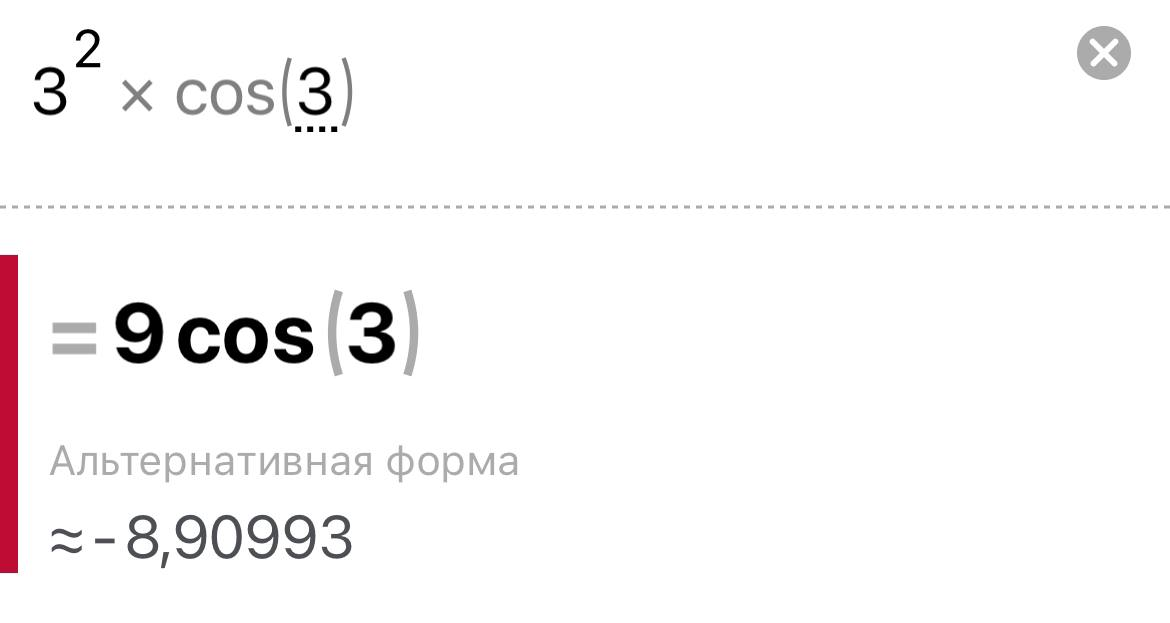


Рисунок 4.1.1 – Тестовый пример №2

Так как m < 9, то Z = m + a ≈ -8.91 + 4 ≈ -4.91.

Ожидаемый результат:

"Рассчитанное значение Z: -4.91" (с учетом округления)

1. Входные данные:

n = 3, b = 6, a = 2.

Расчёт:

b > 4 → m = 3² = 9.

Так как m = 9, Z = sqrt(m² + a) = sqrt(9² + 2) = sqrt(81 + 2) = sqrt(83) ≈ 9.11.

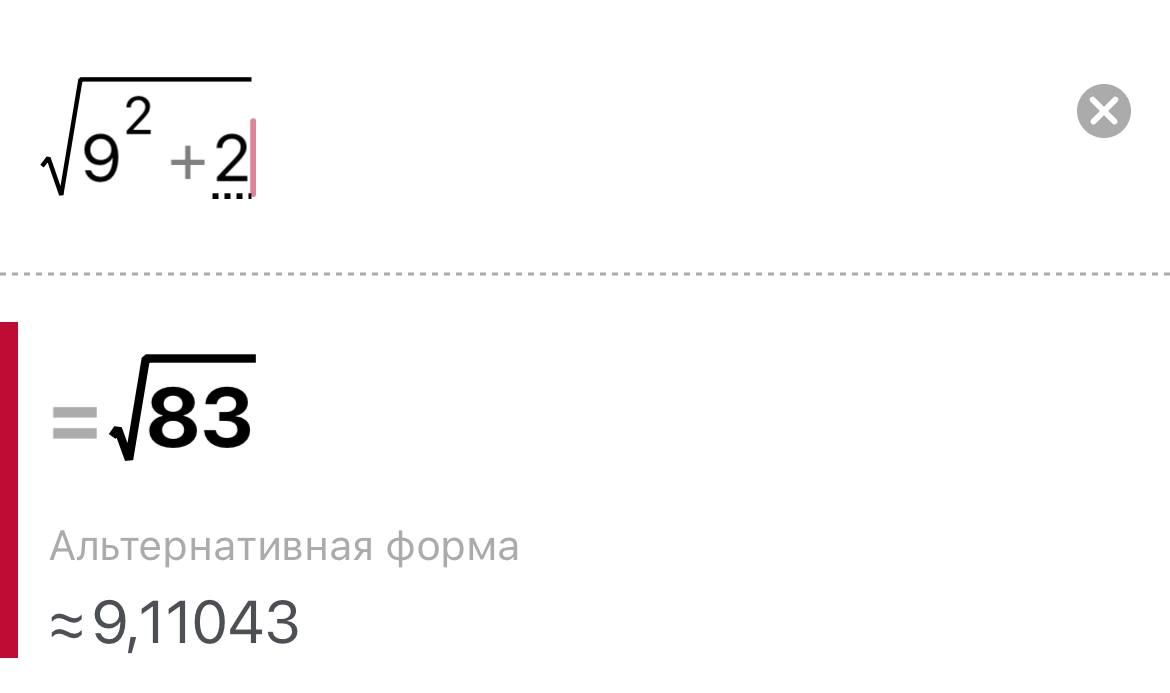


Рисунок 4.1.2 – Тестовый пример №3

Ожидаемый результат:

"Рассчитанное значение Z: 9.11"

1. Входные данные:

n = "abc", b = 4, a = 3.

Ожидаемый результат:

"Ошибка: некорректный ввод данных."

* 1. **Задание 2**

1. Входные данные:

Первый символ: 'd', Второй символ: 'a'.

Расчёт:

Поскольку 'd' (ASCII 100) > 'a' (ASCII 97), происходит обмен.

Ожидаемый результат:

"После обработки: первый символ - a, второй символ - d"

1. Входные данные:

Первый символ: 'A', Второй символ: 'B'.

Расчёт:

Поскольку ASCII-код 'A' (65) < 'B' (66), обмен не выполняется.

Ожидаемый результат:

"После обработки: первый символ - A, второй символ - B"

1. Входные данные:

Первый символ: пустая строка или несколько символов, Второй символ: 'c'.

Ожидаемый результат:

"Ошибка: некорректный ввод символов."

1. **Листинг (код программы)**
   1. **Задание 1**

using System;

class Program

{

static void Main()

{

// Объявление переменных для исходных данных и результатов

double n, b, a, m, Z;

bool isValid = true; // Флаг проверки корректности ввода

// Ввод значения n с проверкой

Console.Write("Введите число n: ");

isValid &= double.TryParse(Console.ReadLine(), out n);

Console.WriteLine();

// Ввод значения b с проверкой

Console.Write("Введите число b: ");

isValid &= double.TryParse(Console.ReadLine(), out b);

Console.WriteLine();

// Ввод значения a с проверкой

Console.Write("Введите число a: ");

isValid &= double.TryParse(Console.ReadLine(), out a);

Console.WriteLine();

// Проверка корректности ввода всех данных

if (isValid)

{

// Вычисление m в зависимости от значения b

if (b > 4)

{

m = Math.Pow(n, 2); // m = n²

}

else

{

m = Math.Pow(n, 2) \* Math.Cos(n); // m = n² \* cos(n)

}

// Вычисление Z в зависимости от значения m

if (m > 9)

{

Z = Math.Pow(a, 2) \* Math.Cos(m); // Если m > 9, Z = a² \* cos(m)

}

else if (m < 9)

{

Z = m + a; // Если m < 9, Z = m + a

}

else // m == 9

{

Z = Math.Sqrt(Math.Pow(m, 2) + a); // Если m = 9, Z = sqrt(m² + a)

}

// Вывод результата вычислений

Console.WriteLine($"Рассчитанное значение Z: {Z}");

}

else

{

Console.WriteLine("Ошибка: некорректный ввод данных.");

}

// Ожидание нажатия клавиши для завершения программы

Console.ReadKey(true);

}

}

* 1. **Задание 2**

using System;

class Program

{

static void Main()

{

// Объявление переменных для символов

char char1, char2;

// Ввод первого символа с проверкой

Console.Write("Введите первый символ: ");

if (!char.TryParse(Console.ReadLine(), out char1))

{

Console.WriteLine("Ошибка: некорректный ввод первого символа.");

return;

}

Console.WriteLine();

// Ввод второго символа с проверкой

Console.Write("Введите второй символ: ");

if (!char.TryParse(Console.ReadLine(), out char2))

{

Console.WriteLine("Ошибка: некорректный ввод второго символа.");

return;

}

Console.WriteLine();

// Если ASCII-код первого символа больше ASCII-кода второго, выполняется обмен

if (char1 > char2)

{

char temp = char1;

char1 = char2;

char2 = temp;

Console.WriteLine($"После обработки: первый символ - {char1}, второй символ - {char2}");

}

else

{

Console.WriteLine("Обмена не требуется, порядок символов корректный.");

Console.WriteLine($"Первый символ - {char1}, второй символ - {char2}");

}

// Ожидание нажатия клавиши для завершения программы

Console.ReadKey(true);

}

}

1. **Тестирование (расчёт тестовых примеров ПК)**
   1. **Задание 1**

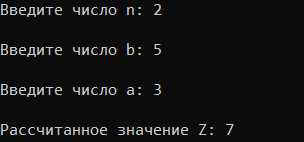
****

Рисунок 6.1.1 – Тестирование №1

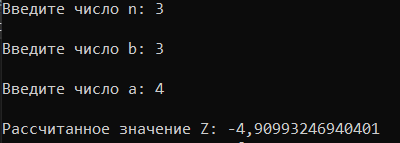


Рисунок 6.1.2 – Тестирование №2

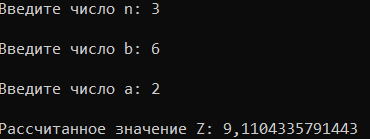


Рисунок 6.1.3 – Тестирование №3

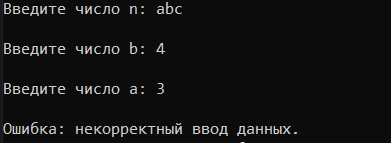


Рисунок 6.1.4 – Тестирование №4

* 1. **Задание 2**

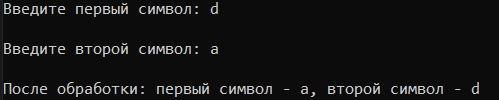
****

Рисунок 6.2.1 – Тестирование №1

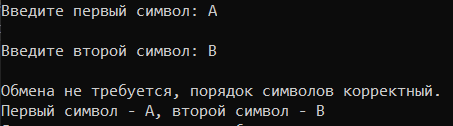


Рисунок 6.2.2 – Тестирование №2

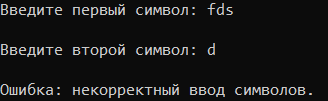


Рисунок 6.3.3 – Тестирование №3

1. **Вывод**

В ходе выполнения лабораторной работы были получены следующие результаты:

Программа для вычисления значения Z демонстрирует применение условных операторов с разветвлением логики, в зависимости от входных данных. Использование метода TryParse позволяет обеспечить контроль корректности ввода и предотвращает выполнение дальнейших вычислений при ошибочном вводе.

Программа для обмена символов корректно сравнивает ASCII-коды введённых символов и, при необходимости, выполняет обмен их значений.

Тестовые примеры подтверждают корректность работы обеих программ как при корректном вводе данных, так и при возникновении ошибок ввода.

Выполненная работа способствовала лучшему пониманию принципов работы условного оператора и методов контроля ввода, что является важным элементом надежного программирования.