МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет радиотехники и электроники

(*факультет/институт*)

Кафедра твердотельной электроники

Направление подготовки 11.03.04 Электроника и наноэлектроника

(*код,**наименование*)

Профиль Микроэлектроника и твердотельная электроника

КУРСОВАЯ РАБОТА

по дисциплине «Информационные технологии в электронике»

Тема «Разработка консольных приложений на языке высокого уровня С#»

Вариант № 40

|  |  |  |
| --- | --- | --- |
| Разработал | М.А.Колесников | |
|  | *(подпись, дата)* | *(инициалы, фамилия)* |
| Руководитель | Е.Ю. Плотникова | |
|  | *(подпись, дата)* | *(инициалы, фамилия)* |
| Защищена | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  | *(дата)* |  |

Воронеж 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет радиотехники и электроники

(*факультет/институт*)

Кафедра твердотельной электроники

ЗАДАНИЕ

на курсовую работу

по дисциплине Информационные технологии в электронике

Тема проекта Разработка консольных приложений на языке высокого уровня С#

Студент группы бЭН-242 Колесников Максим Андреевич

Фамилия, имя, отчество

Номер задания 40

Технические условия Оборудование кафедры полупроводниковой электроники и наноэлектроники, программное обеспечение MS Visual Studio 2022 Community

Объем проекта (графические работы, расчёты и прочее)

Курсовая работа содержит 00 страниц, 00 рисунков, 0 формулы, 0 источника литературы.

Сроки выполнения этапов 10.02.2025-05.06.2025

Срок защиты курсового проекта 06.06.2025

Руководитель Е. Ю. Плотникова

Подпись, дата Инициалы, фамилия

Задание принял студент М.А.Колесников

Подпись, дата Инициалы, фамилия

# ЗАМЕЧАНИЯ РУКОВОДИТЕЛЯ

# СОДЕРЖАНИЕ

[Замечания руководителя 3](#_Toc199776420)

[Содержание 4](#_Toc199776421)

[Реферат 5](#_Toc199776422)

[Введение 6](#_Toc199776423)

[1 Работа с текстом 7](#_Toc199776424)

[2 Расчет простой формулы с проверкой расчёта в MathCad и в Excel](#_Toc199776425)  10

[3 Использование условных конструкций «if-else» и «switch-case» 13](#_Toc199776426)

[4 Расчет сложной формулы 18](#_Toc199776427)

[5 Найти корень уравнения методом Бисекции 25](#_Toc199776430)

[6 Найти сумму и произведение при помощи инструмента «метод» 28](#_Toc199776431)

[7 Работа с массивами 32](#_Toc199776433)

[Заключение 34](#_Toc199776435)

[Список использованных источников 35](#_Toc199776436)

РЕФЕРАТ

C# — Язык программирования C# (произносится как "си-шарп") является современным объектно-ориентированным языком, разработанным компанией Microsoft в рамках платформы .NET. Он предназначен для создания разнообразных приложений, включая десктопные, веб- и мобильные решения. C# отличается высокой производительностью, простотой синтаксиса и богатым набором возможностей, что делает его популярным среди разработчиков. Этот фреймворк позволяет быстро разрабатывать масштабируемые и производительные веб-сайты и сервисы. C# поддерживает асинхронное программирование через ключевые слова async и await, что упрощает разработку приложений, работающих с длительными операциями ввода-вывода без блокировки потоков. За разработку С# и адаптацию его под Windows взялась группа программистов Microsoft, которую возглавляли Андерс Хейлсенберг и Скотт Вильтаумот, уже создавшие до этого языки Delphi, Pascal Turbo и TypeScriht. C# — это мощный и гибкий язык программирования, который нашел применение в самых различных областях IT. Его особенности, такие как объектно-ориентированность, сильная типизация и автоматическое управление памятью, делают его привлекательным выбором для разработчиков.

Цель работы: изучить язык программирования «C#» и создания консольных приложений на его основе.

Объект исследования: программирование на языке «C#».

ВВЕДЕНИЕ

Современные технологии программирования играют ключевую роль в разработке программного обеспечения, и язык C# является одним из наиболее востребованных инструментов в этой области. Благодаря своей универсальности, высокой производительности и интеграции с платформой .NET, C# широко применяется для создания desktop-приложений, веб-сервисов, мобильных и игровых приложений. Актуальность данной курсовой работы обусловлена возрастающей потребностью в эффективных и надежных программных решениях, а также необходимостью изучения современных подходов к разработке на языке C# [1].

С момента своего появления C# быстро завоевал популярность благодаря своей простоте и мощным возможностям. В 2005 году вышла версия C# 2.0, которая добавила такие функции, как обобщения и анонимные методы. В 2010 году была представлена версия C# 4.0, включающая динамическую типизацию и поддержку параллелизма [3].

За годы своего существования C# приобрел широкую популярность среди разработчиков по всему миру. Он активно используется в создании корпоративных приложений, веб-сервисов, мобильных приложений, игр (особенно с использованием игрового движка Unity) [2], а также в области облачных технологий и интернета вещей. Благодаря своей универсальности и поддержке современных стандартов, C# остается одним из ведущих языков программирования в индустрии.

# Базовые математические операции языка C#

# Вывод текста в рамке

В первом задании необходимо вывести текст в рамке.

Текст, который должен быть в рамке: «Хоффман развил свою предыдущую работу по моделированию TFT [27] исследованиями на DC (постоянном токе) до модели тонкопленочного транзистора с длинным каналом, что очень важно при расчете зависимости подвижности от напряжения на затворе [46].»

Программа для вывода:

namespace Task\_1

{

class Program

{

static void Main(string[] args)

{

string text = "Хоффман развил свою предыдущую работу по моделированию TFT [27] исследованиями на DC (постоянном токе) до модели тонкопленочного транзистора с длинным каналом, что очень важно при расчете зависимости подвижности от напряжения на затворе [46].";

//

PrintTextInFrame(text);

}

static void PrintTextInFrame(string text)

{

//

string[] lines = WrapText(text, Console.WindowWidth - 6);

//

int maxLength = 0;

foreach (string line in lines)

{

//

if (line.Length > maxLength) maxLength = line.Length;

}

// Для текста:

// ╔ ╗ ║ ╚ ╝ ═

// https://www.megacooltext.com/generator/big-letters/

Console.WriteLine("╔" + new string('═', maxLength + 2) + "╗");

foreach (string line in lines)

{

Console.WriteLine("║ " + line.PadRight(maxLength) + " ║");

}

Console.WriteLine("╚" + new string('═', maxLength + 2) + "╝");

}

static string[] WrapText(string text, int maxWidth)

{

string[] words = text.Split(' ');

System.Collections.Generic.List<string> lines = new System.Collections.Generic.List<string>();

string currentLine = "";

foreach (string word in words)

{

if ((currentLine.Length + word.Length + 1) <= maxWidth)

{

currentLine += (currentLine.Length == 0) ? word : " " + word;

}

else

{

lines.Add(currentLine);

currentLine = word;

}

}

if (!string.IsNullOrEmpty(currentLine))

lines.Add(currentLine);

return lines.ToArray();

}

}

}

Результат задания 1 на рисунке 1.

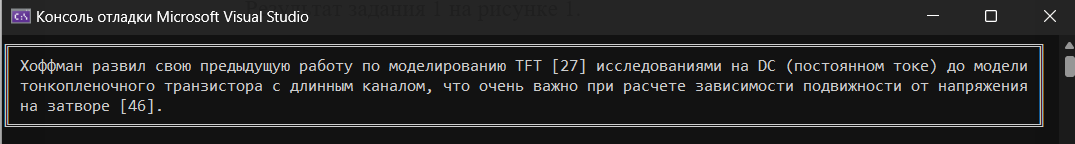


Рисунок 1 – Вывод текста в рамке

2 Расчет простой формулы с проверкой расчёта в MathCad и в Excel

Ввести с клавиатуры данные для расчета в формате с плавающей запятой. Вывести на консоль решение с 2 знаками после запятой, и сделать проверку в MathCad и в Excel (1)

*σ=F/s=Eε*  (1)

где – относительное удлинение;

;

;

;

namespace Task\_2

{

class Program

{

static void Main(string[] args)

{

//

Console.WriteLine("a=F/s=Ee механич. напряжение, Па");

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

Console.Write("Решение будет по той формуле, значения для которой ~= 0.\nЕсли подходят обе, то берётся случайная\n");

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

//

// Значение для проверки валидности, если F и S валидны

int choice = 0;

//

// Проверка чтобы на эксепшены, ( IsValid() )

//

// local numF = GetNumber() or 0

//

Console.WriteLine("Введите F: ");

double.TryParse(Console.ReadLine(), out double numF);

//

Console.WriteLine("Введите s:");

double.TryParse(Console.ReadLine(), out double numS);

//

Console.WriteLine("Введите E:");

double.TryParse(Console.ReadLine(), out double numE);

//

Console.WriteLine("Введите e:");

double.TryParse(Console.ReadLine(), out double numExt);

//

if (numF != 0 && numS != 0)

{

choice = 1;

}

if (numE != 0 && numExt != 0)

{

choice = 2;

}

if (numF != 0 && numS != 0 && numE != 0 && numExt != 0)

{

Random random = new Random();

choice = random.Next(1, 2);

}

//

switch(choice)

{

case 0:

Console.WriteLine("Невозможно выполнить операцию.");

break;

case 1:

Console.WriteLine("Механическое напряжение = " + Math.Round(numF/numS, 2) + " Па");

break;

case 2:

Console.WriteLine("Механическое напряжение = " + Math.Round(numE\*numExt, 2) + " Па");

break;

}

}

}

}

Результаты работы программы на рисунке 2.

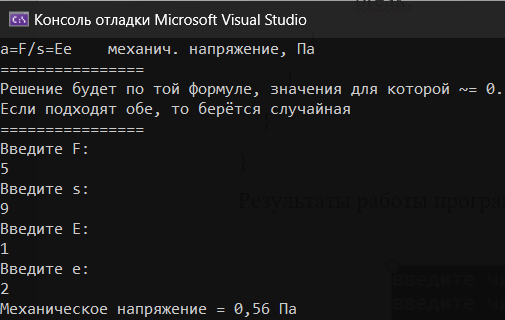


Рисунок 2 – Расчет относительного удлинения

Результаты проверки расчета в Excel на рисунке 3.

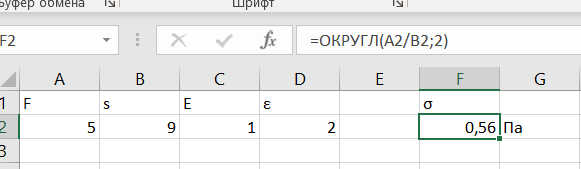


Рисунок 3 - Расчет относительного удлинения

Результаты расчета в MathCad на рисунке 4.

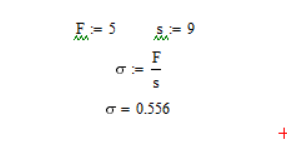


Рисунок 4 - Расчет относительного удлинения

# 3 Использование условных конструкций «if-else» и «switch-case»

В первой случае необходимо решить задание при помощи конструкции «if-else», а во втором случае при помощи «switch-case».

Решить в целых числах уравнение ( ax + b ) : ( cx + d ) = 0. Вводятся 4 числа: a, b, c и d; c и d не равны нулю одновременно. Необходимо вывести все решения, если их число конечно, “NO” (без кавычек), если решений нет, и “INF” (без кавычек), если решений бесконечно много.

namespace Task\_3

{

class Program

{

static void Main()

{

Console.WriteLine("Введите четыре числа a, b, c, d (c и d не равны нулю одновременно):");

// Parse круче Convert, Change my mind

int numA = CheckForInt(Console.ReadLine());

int numB = CheckForInt(Console.ReadLine());

int numC = CheckForInt(Console.ReadLine());

int numD = CheckForInt(Console.ReadLine());

if (numC == 0 && numD == 0)

{

Console.WriteLine("c и d не могут быть нулями одновременно, ахтунг и эксепшн, завершение работы.");

return;

}

// local function checkForInt( var )

static int CheckForInt( string inputStr )

{

if (int.TryParse(inputStr, out int ResultInt))

{

return ResultInt;

}

else;

{

Environment.Exit(1);

}

return 0;

}

Console.WriteLine("Вывод:");

bool hasSolution = false;

int? solution = null;

// (ax + b) : (cx + d) = 0

if (numA != 0)

{

if ((-numB) % numA == 0)

{

int numX = -numB / numA;

//

if (numC\*numX+numD != 0)

{

hasSolution = true;

solution = numX;

}

}

}

else

{

if (numB == 0)

{

if (numC == 0)

{

Console.WriteLine("INF");

return;

}

else

{

Console.WriteLine("INF");

return;

}

}

else

{

}

}

if (hasSolution)

{

Console.WriteLine(solution);

}

else

{

Console.WriteLine("NO");

}

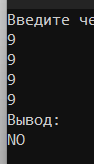
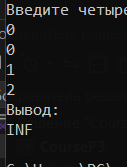
return;

}

}

}

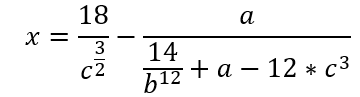
Результат работы программы на рисунке 5

# Рисунок 5 – Решение с помощью «if-else» и «switch-case»

4 Расчет сложной формулы

Рассчитать формулу из задания с подстановкой любых целых значений *a*, *b* и *с* значений в переменные, выполнить проверку расчёта в MathCad и в Excel (2)



using System;

class Program

{

static void Main()

{

double a = 3;

double b = 3;

double c = 4;

double x\_alt = (18 / Math.Pow(c, 1.5)) - (a / ((14 / (b \* b)) + a - 12 \* Math.Pow(c, 3)));

Console.WriteLine(x\_alt);

}

}

Результат работы программы на рисунке 6.

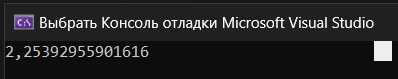


Рисунок 6 – Расчет сложной формулы

Проверка расчета программы в MathCad и в Excel на рисунках 7 – 8.

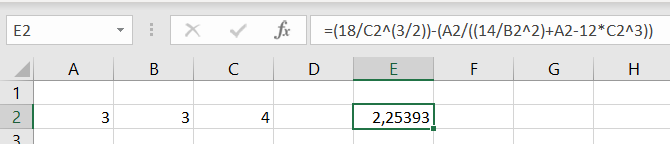


Рисунок 7 – Расчет сложной формулы в Excel

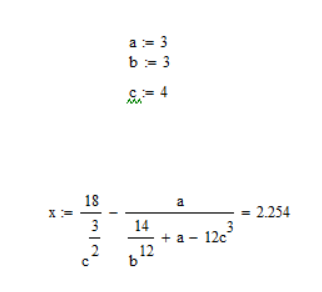


Рисунок 8 – Расчет сложной формулы в MathCad

Изменяя указанную в варианте переменную, пересчитать исходную формулу с помощью цикла «For», выполнить проверку расчёта в MathCad и в Excel. Вывести на консоль последнее число *x*.

using System;

class Program

{

static void Main()

{

double b = 3;

double c = 4;

// Цикл for для a от -5 до 5 с шагом 0.5

for (double a = -5; a <= 5; a += 0.5)

{

double x\_alt = (18 / Math.Pow(c, 1.5)) - (a / ((14 / (b \* b)) + a - 12 \* Math.Pow(c, 3)));

Console.WriteLine($"a = {a,4:F1} -> x = {x\_alt:F6}");

}

}

}

Результаты работы программы на рисунке 9.

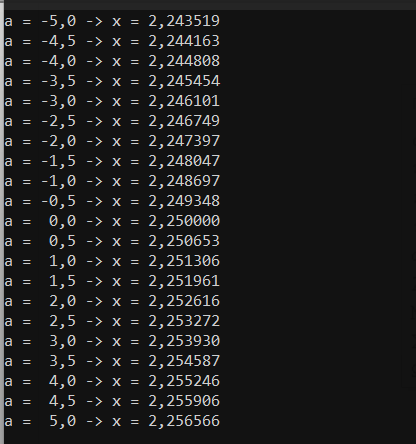
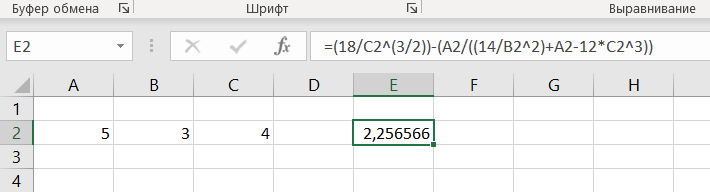


Рисунок 9 – Расчет последнего значения x

Результаты работы в Excel на рисунках 10.



Результаты проверки программы в MathCad на русунке11

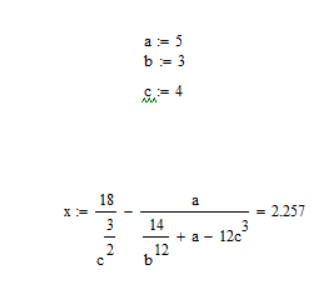


Рисунок 11 – Расчет последнего значения x в MathCad

Изменяя указанную в варианте переменную, пересчитать исходную формулу с помощью циклов «While» и «Do-While», выполнить проверку расчёта в MathCad и в Excel. Вывести на консоль последнее число *x*.

using System;

class Program

{

static void Main()

{

double b = 3;

double c = 4;

// Цикл WHILE

double a\_while = -5;

double lastXWhile = 0;

while (a\_while <= 5)

{

lastXWhile = (18 / Math.Pow(c, 1.5)) - (a\_while / ((14 / (b \* b)) + a\_while - 12 \* Math.Pow(c, 3)));

a\_while += 0.5;

}

// Цикл DO-WHILE

double a\_do\_while = -5;

double lastXDoWhile = 0;

do

{

lastXDoWhile = (18 / Math.Pow(c, 1.5)) - (a\_do\_while / ((14 / (b \* b)) + a\_do\_while - 12 \* Math.Pow(c, 3)));

a\_do\_while += 0.5;

}

while (a\_do\_while <= 5);

Console.WriteLine($"WHILE: a = {a\_while - 0.5:F1}, x = {lastXWhile:F6}");

Console.WriteLine($"DO-WHILE: a = {a\_do\_while - 0.5:F1}, x = {lastXDoWhile:F6}");

}

}

Результат работы программы на рисунке 12.

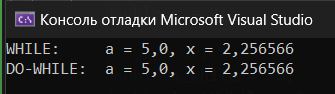


Рисунок 12 – Расчет последнего значения x с помощью циклов «While» и «Do-While»

Результат расчета в Excel на рисунках 13.

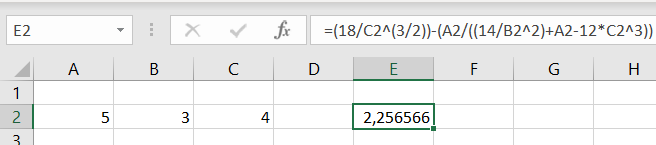


Рисунок 13 – Расчет последнего значения x в Excel

Результаты расчета в MathCad на рисунке 14.

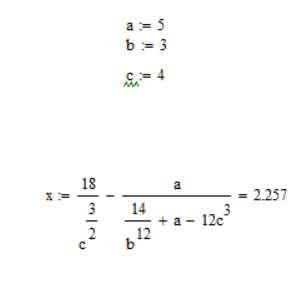


Рисунок 14 – Расчет последнего значения x в MathCad

# 5 Найти корень уравнения методом Бисекций

Найти корень нелинейного уравнения методом бисекций, построить график функции в Excel, построить график функции и найти корень уравнения в MathCad (3)

Cos(x-6)=-x4+1 (3)

using System;

class Program

{

static void Main()

{

double x0 = 0.1;

double epsilon = 1e-6;

int maxIterations = 100;

double x = x0;

Console.WriteLine($"{"Итерация",-10} {"x",-15} {"f(x)",-15} {"Погрешность",-15}");

Console.WriteLine(new string('-', 55));

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

{

double xNew = IterationFunction(x);

double error = Math.Abs(xNew - x);

Console.WriteLine($"{i + 1,-10} {x,-15:F8} {Function(x),-15:F8} {error,-15:E2}");

if (error < epsilon)

{

return;

}

x = xNew;

}

}

static double IterationFunction(double x)

{

double arg = -Math.Pow(x, 4) + 1;

if (arg < -1) arg = -1;

if (arg > 1) arg = 1;

return 6 - Math.Acos(arg);

}

static double Function(double x)

{

return Math.Cos(x - 6) + Math.Pow(x, 4) - 1;

}

}Результат номера 5 на рисунке 15.

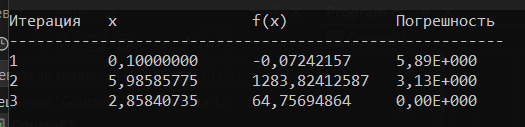


Рисунок 15 – Поиск корней уравнения методом простых итераций

Результаты 5 в Excel на рисунке 16.

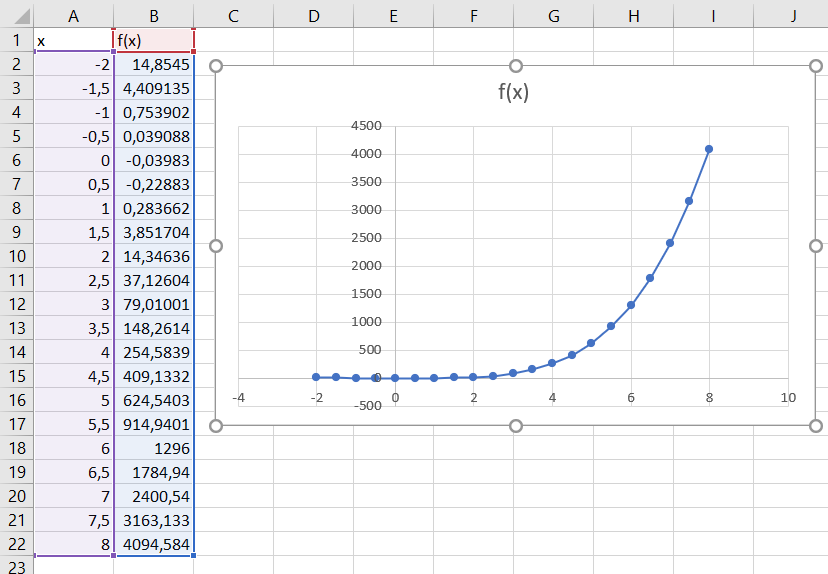


Рисунок 16 – График функции в Excel

Результаты 5 в MathCad на рисунке 17.

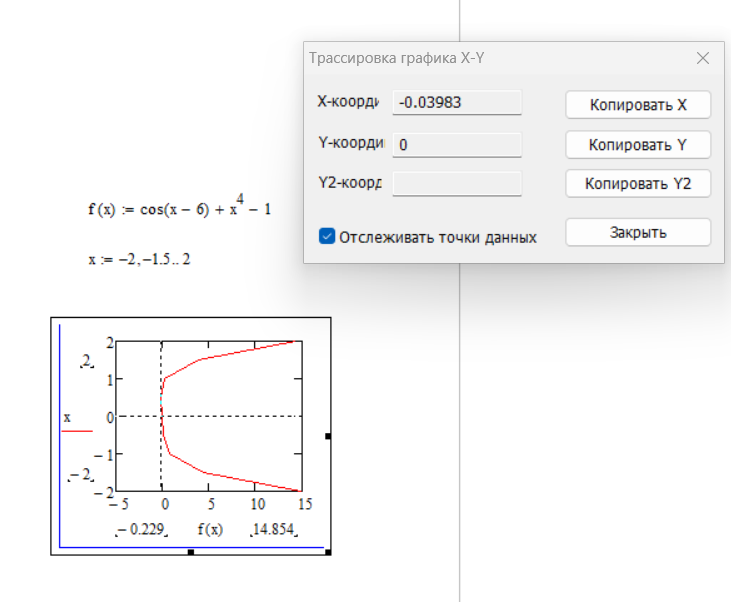


Рисунок 17 – График функции в MathCad

# 6 Найти сумму и произведение при помощи инструмента «метод»

Найти сумму и произведение заданного выражения, используя для решения инструмент «метод». Количество элементов суммирования и произведения в C#, на MathCadе и в Excel должно совпадать (4)

(4)

using System;

namespace SumAndProduct

{

class Program

{

static long Factorial(int number)

{

long result = 1;

for (int i = 1; i <= number; i++)

{

result \*= i;

}

return result;

}

static void Main(string[] args)

{

int nStart = 1;

int nEnd = 5;

double sum = 0;

double product = 1;

Console.WriteLine("Вычисление для n от {0} до {1}:", nStart, nEnd);

Console.WriteLine("n\tЗначение выражения");

for (int n = nStart; n <= nEnd; n++)

{

double expressionValue = ((double)n / Factorial(n)) \* Math.Sin(2 / Math.Pow(3, n));

Console.WriteLine($"{n}\t{expressionValue:F10}");

sum += expressionValue;

product \*= expressionValue;

}

Console.WriteLine("\nСумма выражений: " + sum);

Console.WriteLine("Произведение выражений: " + product);

Console.WriteLine("Произведение (науч. нотация): " + product.ToString("E5"));

}

}

}

Результат 6 на рисунке 18.

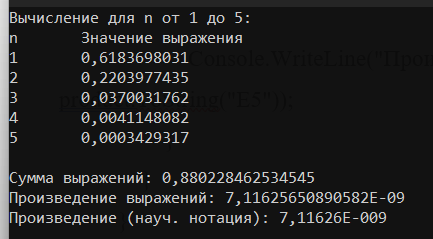


Рисунок 18 – Поиск суммы и произведения

Расчет суммы и произведения в Excel на рисунке 19.

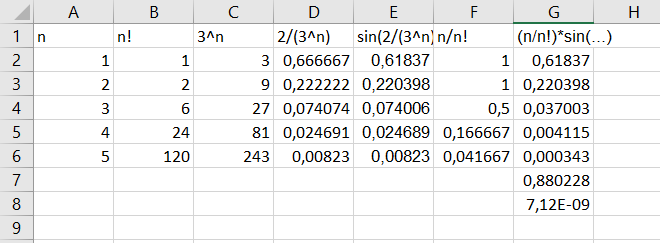


Рисунок 19 – Расчет суммы и произведения в Excel

Расчет суммы и произведения в mathcad на рисунке 20.

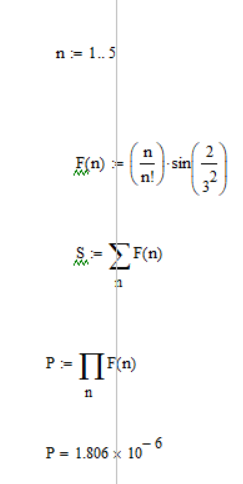


Рисунок 20 – Расчет суммы и произведения в mathcad

7 Работа с массивами

Работа с массивами: ввод количества элементов массива, ввод элементов массива заданным образом; расчет массива по варианту; вывод результатов расчёта.

Ввести количество элементов массива в исходный код программы (не менее 10).

Ввести элементы массива случайным образом.

Упорядочить элементы по убыванию.

Вывести на экран результат работы программы.

class Program

{

static void Main()

{

int arrayLength = 15;

Console.WriteLine($"Количество элементов массива: {arrayLength}");

int[] array = new int[arrayLength];

Random random = new Random();

Console.WriteLine("\nИсходный массив:");

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

{

array[i] = random.Next(-100, 101);

Console.Write(array[i] + " ");

}

Array.Sort(array);

Array.Reverse(array);

Console.WriteLine("\n\nМассив, упорядоченный по убыванию:");

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

{

Console.Write(array[i] + " ");

}

Console.WriteLine("\n\nНажмите любую клавишу для выхода...");

Console.ReadKey();

}

}

Результат 7 на рисунке 21

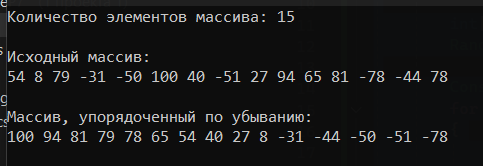


Рисунок 21 – Работа с массивами

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была рассмотрена и реализована разработка консольного приложения с использованием языка программирования высокого уровня C#. В процессе работы были изучены основные принципы создания консольных программ, особенности синтаксиса языка C#, а также методы взаимодействия с пользователем через консольный интерфейс.

Разработка приложения позволила на практике закрепить знания о структуре C#-программы, работе с переменными, операторами управления, методами, классами и коллекциями. Также было уделено внимание обработке исключений и обеспечению стабильной работы программы при различных вводных данных.

Полученные в ходе выполнения курсовой работы навыки могут быть успешно применены при создании более сложных программных решений, а опыт проектирования и отладки консольных приложений является прочной основой для дальнейшего изучения технологий программирования на платформе .NET.

Таким образом, поставленные цели курсовой работы достигнуты: теоретические знания были успешно применены на практике, а полученные значения в Visual Studio совпали с проверкой в Excel и в MathCad.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Троелсен Э. Язык программирования C# 10 и платформа .NET 6: учеб. пособие / Э. Троелсен, Ф. Джепикс. – 8-е изд., перераб. и доп. – М.: Вильямс, 2022. – 1056 с. – Текст: непосредственный.

2. Шарп Дж. Microsoft Visual C#. Подробное руководство. / Дж. Шарп. – 8-е изд. – СПб.: Питер, - 2017. – 848 с.

3. Шилдт Г. C# 9 и .NET 5. Полное руководство: учеб. пособие / Г. Шилдт. – 1-е изд. – М.: ДМК Пресс, 2021. – 928 с. – Текст: непосредственный.

4. Гофман А. А. Программирование на C# для начинающих: учеб. пособие / А. А. Гофман. – 2-е изд., перераб. и доп. – СПб.: Питер, 2020. – 384 с. – Текст: непосредственный.