**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Синергия |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Программист |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе №2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Организация циклов | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Разработка программных модулей |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Осовицкий Борис Николаевич |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-205прог |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сиберев И.В. |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**

**Лабораторная работа №2. «Организация циклов»**

Задание 1.

Таблица значений функции Вычислить и вывести на экран в виде таблицы значения функции, заданной графически (см. задание 1 лабораторной работы 2), на интервале от х до х с шагом dr. Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.

Листинг кода:

**using System;**

**class Program**

**{**

**static void Main()**

**{**

**// Заголовок таблицы**

**Console.WriteLine("Таблица значений функции f(x)");**

**Console.WriteLine("-------------------------------");**

**Console.WriteLine("|    x    |       f(x)        |");**

**Console.WriteLine("-------------------------------");**

**// Значения x от -5.0 до 5.0 с шагом 0.5**

**for (double x = -5.0; x <= 5.0; x += 0.5)**

**{**

**// Вычисление f(x)**

**double fx;**

**if (x <= 0)**

**{**

**fx = x \* x; // Для x ≤ 0: f(x) = x²**

**}**

**else**

**{**

**// Для x > 0: f(x) = sin(x)/x**

**fx = Math.Sin(x) / x;**

**}**

**// Форматированный вывод строки таблицы**

**Console.WriteLine($"| {x,6:F2} | {fx,17:F6} |");**

**}**

**// Нижняя граница таблицы**

**Console.WriteLine("-------------------------------");**

**}**

**}**

**Код на компиляторе:**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

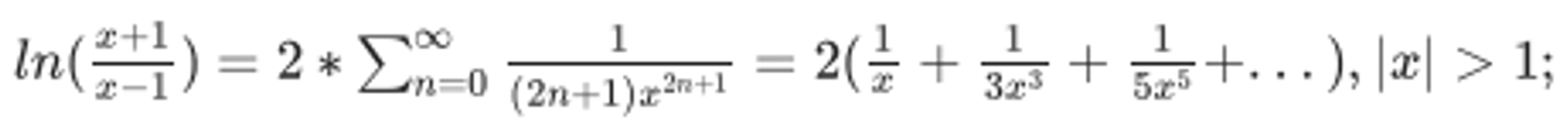
Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Контент, сгенерированный ИИ, может содержать ошибки.

Задание 2.

Ряды Тейлора Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от х до х с шагом dх с точностью в. Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда.



**Листинг кода:**

**using System;**

**class Program**

**{**

**static void Main()**

**{**

**double xStart = 1.1;    // Начальное значение x (>1)**

**double xEnd = 3.0;      // Конечное значение x**

**double dx = 0.2;        // Шаг**

**double epsilon = 1e-6;  // Точность**

**Console.WriteLine("Таблица значений функции ln((x + 1)/(x - 1)), вычисленной по ряду Тейлора");**

**Console.WriteLine("--------------------------------------------------");**

**Console.WriteLine("|    x    |  ln((x+1)/(x-1))  |   Члены ряда (N)  |");**

**Console.WriteLine("--------------------------------------------------");**

**for (double x = xStart; x <= xEnd; x += dx)**

**{**

**int termsUsed;**

**double seriesResult = TaylorLn((x + 1) / (x - 1), epsilon, out termsUsed);**

**Console.WriteLine($"| {x,6:F2} | {seriesResult,18:F10} | {termsUsed,17} |");**

**}**

**Console.WriteLine("--------------------------------------------------");**

**}**

**// Функция для вычисления ряда Тейлора**

**static double TaylorLn(double y, double eps, out int termsUsed)**

**{**

**double sum = 0.0;**

**double term;**

**int n = 0;**

**termsUsed = 0;**

**do**

**{**

**term = 1.0 / ((2 \* n + 1) \* Math.Pow(y, 2 \* n + 1));**

**sum += term;**

**n++;**

**termsUsed++;**

**} while (Math.Abs(term) >= eps);**

**return 2 \* sum;**

**}**

**}**

**Код на компиляторе:**

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.