

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

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

.

**Отчет по лабораторной работе № 3**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **на тему** | |  | Функции и рекурсия | | | | |
|  | | | | | |  | (наименование темы) |
|  |  | |  | | | | |
| **по дисциплине** | | | |  | Разработка программных модулей | | |
|  | | | | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Чилингиров Нури Эскендерович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-112 Прог |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва, 2025**

**7** Вариант

7. Вычислить значение x = a , используя рекуррентную формулу xn = 1 1 1 2 n n a x x − − ⎛ ⎞ ⎜ ⎟ + ⎝ ⎠ , в качестве начального значения использовать x0 = 0,5(1 + a).

Листинг кода:

using System;

class Program

{

static void Main()

{

Console.WriteLine("Вычисление квадратного корня методом Ньютона");

Console.WriteLine("Рекуррентная формула: xₙ = 0.5 \* (xₙ₋₁ + a/xₙ₋₁)");

Console.WriteLine("Начальное приближение: x₀ = 0.5 \* (1 + a)");

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

// Вариант 1: Вычисление для конкретного числа

double a = 2.0;

double precision = 1e-10;

int maxIterations = 100;

Console.WriteLine($"\nВычисление √{a} с точностью {precision}:");

double result = CalculateSquareRoot(a, precision, maxIterations);

Console.WriteLine($"Результат: {result:F10}");

Console.WriteLine($"Проверка: Math.Sqrt({a}) = {Math.Sqrt(a):F10}");

// Вариант 2: Таблица вычислений для разных значений

Console.WriteLine("\nТаблица вычислений для a от 1 до 10:");

Console.WriteLine(" a | Наш метод | Math.Sqrt | Итераций");

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

for (int i = 1; i <= 10; i++)

{

double value = i;

var (sqrt, iterations) = CalculateSquareRootWithInfo(value, precision, maxIterations);

Console.WriteLine($"{value,2} | {sqrt,9:F6} | {Math.Sqrt(value),9:F6} | {iterations,8}");

}

}

// Метод вычисления квадратного корня с возвратом количества итераций

static (double result, int iterations) CalculateSquareRootWithInfo(double a, double precision, int maxIterations)

{

if (a < 0)

throw new ArgumentException("Число должно быть неотрицательным");

double xPrev = 0.5 \* (1 + a); // Начальное приближение

double xCurrent;

int iterations = 0;

for (iterations = 0; iterations < maxIterations; iterations++)

{

xCurrent = 0.5 \* (xPrev + a / xPrev);

if (Math.Abs(xCurrent - xPrev) < precision)

break;

xPrev = xCurrent;

}

return (xPrev, iterations + 1);

}

// Основной метод вычисления квадратного корня

static double CalculateSquareRoot(double a, double precision, int maxIterations)

{

var (result, \_) = CalculateSquareRootWithInfo(a, precision, maxIterations);

return result;

}

}

Скрин кода: 