Реферат

Пояснительная записка курсовой работы состоит из 18 страниц, 6 рисунков, 1 таблиц, 3 источника, 1 приложение.

КЛЮЧЕВЫЕ СЛОВА КУРСОВОЙ РАБОТЫ

МЕТОД СИМПСОНА, АЛГОРИТМ, КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ, ИНТЕРВАЛ, ШАГ.

Объектом исследования является – консольное приложение в C#.

Цель работы состоит в изучении правил и методов конструирования программ с использованием языка программирования C# и создания консольного приложения для решения поставленной математической задачи.

К полученным результатам относятся – консольное приложение на языке высокого уровня с#, для решения определенных интегралов методом Симпсона.

Содержание

|  |  |
| --- | --- |
| Введение | 5 |
| Нормативные ссылки | 6 |
| Описание методов вычислительной математики, используемых при решении поставленной задачи | 7 |
| Описание среды разработки | 11 |
| Листинг | 12 |
| Описание программы | 13 |
| Результат машинного тестирования программы | 14 |
| Заключение | 16 |
| Список использованных источников | 17 |
| Приложение | 18 |

Введение

Целью данной курсовой работы является написание консольного приложения на языке высокого уровня С#. Данное приложение решает определенные интегралы методом Симпсона.

Для достижения поставленной цели необходимо решить следующие взаимосвязанные задачи:

- Выбрать оптимальное средство разработки;

- Разработать приложение;

- Протестировать приложение и сравнить результаты с онлайн калькулятором.

1 Нормативные ссылки

ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления"

ГОСТ Р 7.0.5–2008 "Стандартизация в Российской Федерации. Основные положения"

2 Теоретическая постановка.

2.1 Описание методов вычислительной математике, используемых при решении поставленной задачи

Если заменить график функции    на каждом отрезке    разбиения не отрезками прямых как в методах прямоугольников и трапеций, а дугами парабол, то получим более точную формулу приближенного вычисления определенного интеграла.



Предварительно найдем площадь *S* криволинейной трапеции, ограниченной сверху графиком параболы   , сбоку – прямыми  ,   и снизу – отрезком    (см. рисунок 1).

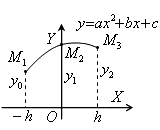


Рисунок 1 - Криволинейная фигура, ограниченная по сторонам

Пусть парабола проходит через три точки   ,   ,  , где  
 – ордината параболы в точке  ;  
  – ордината параболы в точке  ;  
  – ордината параболы в точке  .



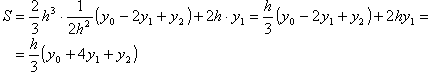
Площадь криволинейной фигуры, показанной на рисунке 2 равна



Выразим эту площадь через    . Из равенств для ординат    находим, что    ,  .



Подставляя эти значения **a** и **b** в равенство    , получаем



Получим теперь формулу парабол для вычисления интеграла   . Для этого отрезок [*a;b*] разобьем на *2n* равных частей (отрезков) длиной   c точками   , где   .



В точках деления    вычисляем значения подынтегральной функции   , где   .

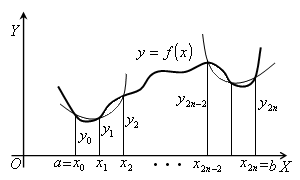
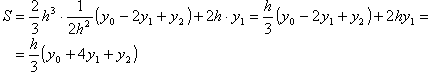


Рисунок 2 - Криволинейная фигура, полученная по методу Симпсона (парабол)

Заменяем каждую пару соседних элементарных криволинейных трапеций с основаниями, равным *h*, одной элементарной параболической трапецией с основанием, равным *2h*.

На отрезке    парабола проходит через три точки   ,   ,   . Используя формулу



находим  .



Аналогично находим , .



Сложив полученные равенства, имеем



или

.



Формула



называется формулой Симпсона (парабол).

Абсолютная погрешность вычисления по формуле оценивается соотношением   ,



где .



Отметим, что формула



дает точное значение интеграла    во всех случаях, когда *f(x)* – многочлен, степень которого меньше или равна трем (тогда   ).



2.2 Описание среды разработки

Используется среда разработки Microsoft Visual Studio 2019. Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

3 Реализация изученных методов

3.1 Листинг

Данная задача была реализована в листинге 1.

Листинг 1

using System;

namespace kursovoi\_project\_sharangiya

{

class Program

{

static void Main(string[] args)

{

// Нижний и верхний пределы интегрирования, количество шагов.

double a, b, steps;

Console.Write("Нижний предел интегрирования = ");

a = Convert.ToDouble(Console.ReadLine());

Console.Write("Верхний предел интегрирования = ");

b = Convert.ToDouble(Console.ReadLine());

Console.Write("Количество шагов = ");

steps = Convert.ToDouble(Console.ReadLine());

// Шаг интегрирования, сумма четных и нечетных элементов

double h, sumEven = 0, sumOdd = 0;

h = (b - a) / (2 \* steps);

for (int i = 1; i <= 2 \* steps - 1; i += 2)

{

// Значения с нечётными индексами, которые нужно умножить на 4.

sumOdd += function\_integral(a + h \* i);

// Значения с чётными индексами, которые нужно умножить на 2.

sumEven += function\_integral(a + h \* (i + 1));

}

Console.Write("Ответ:");

Console.Write((b - a) / (6 \* steps) \* (function\_integral(a) + 4 \* sumOdd + 2 \* sumEven - function\_integral(b)));

}

// Функция интеграла

static double function\_integral(double x)

{

return x \* x \* Math.Sqrt(4 - x \* x);

}

}

}

3.2 Описание программы

Блок схема программы (см. рисунок 3)

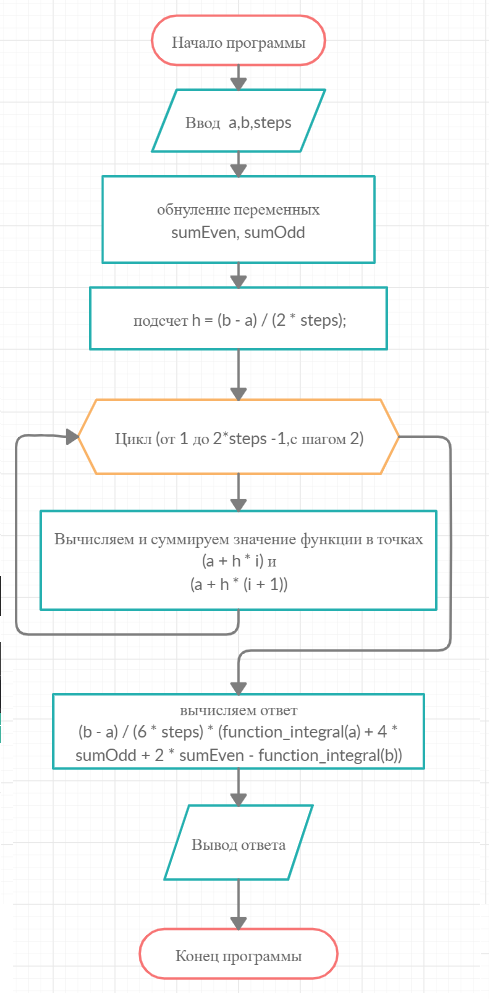


Рисунок 3 – Блок схема программы

3.3 Результат машинного тестирования программы

Вывод консольного приложения (см. рисунок 4)

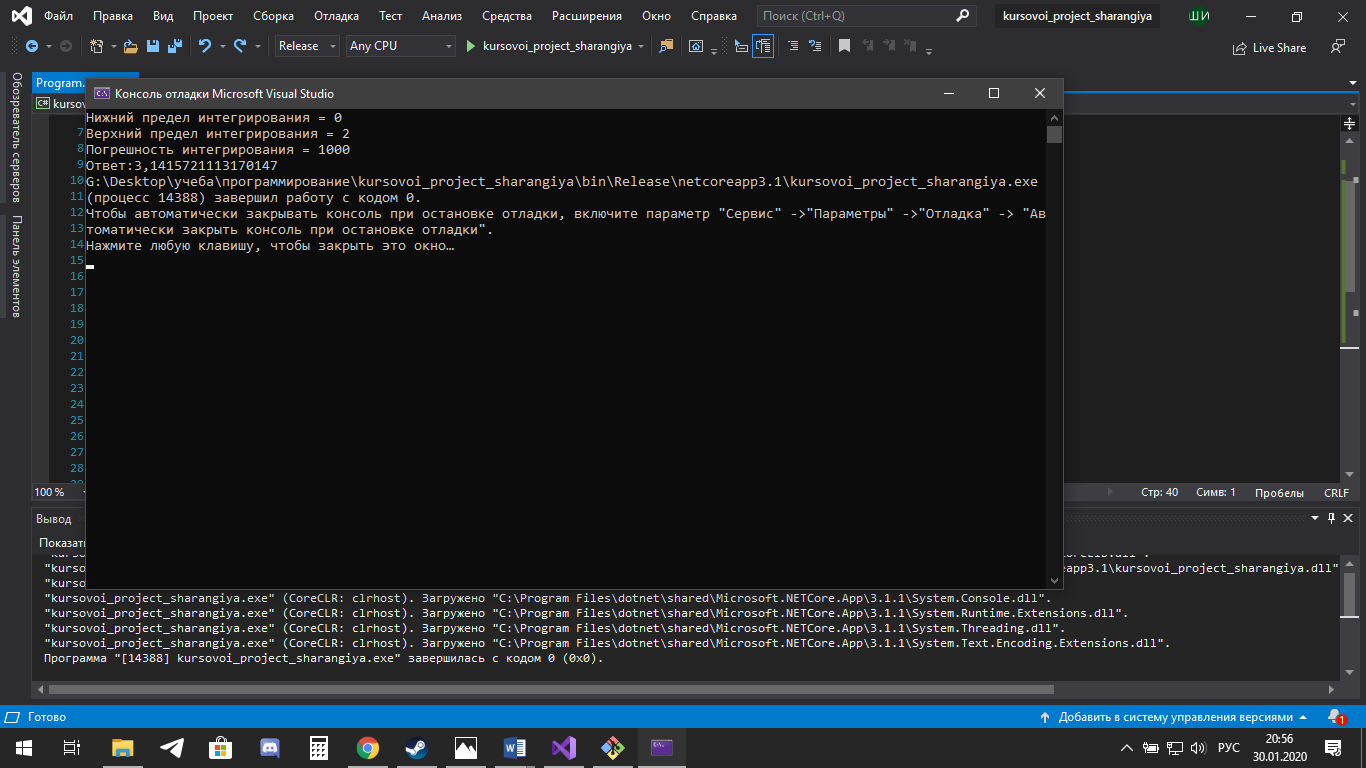


Рисунок 4 – Вывод выполнения программы.

Сравнение с онлайн калькулятором [электронный ресурс] [https://www.kontrolnaya-rabota.ru/s/integral/opredelennyij/?top=2&function=x%5E2\*sqrt%284-x%5E2%29&X=x&bottom=0](https://www.kontrolnaya-rabota.ru/s/integral/opredelennyij/?top=2&function=x%5E2*sqrt%284-x%5E2%29&X=x&bottom=0) (дата обращения:  30.01.2020) (см. рисунок 5-6)

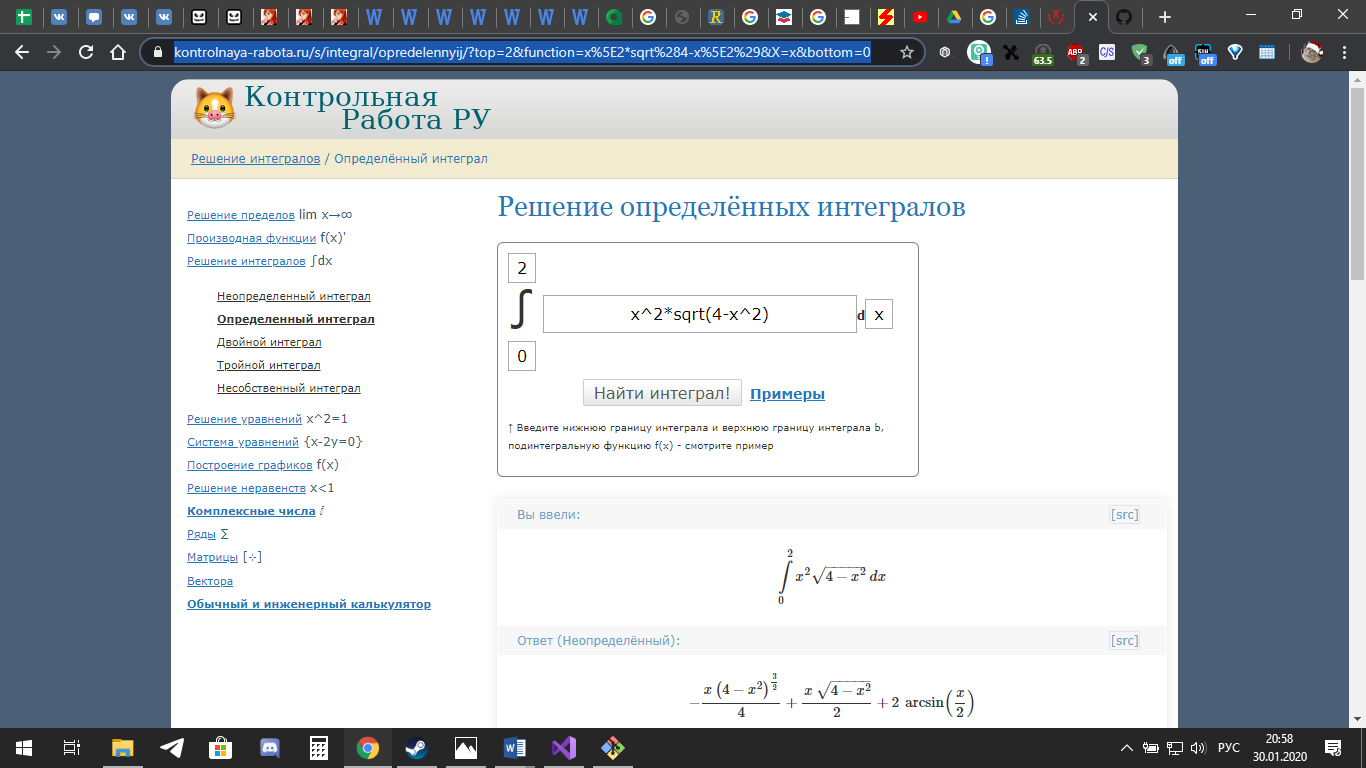


Рисунок 5 – Онлайн калькулятор

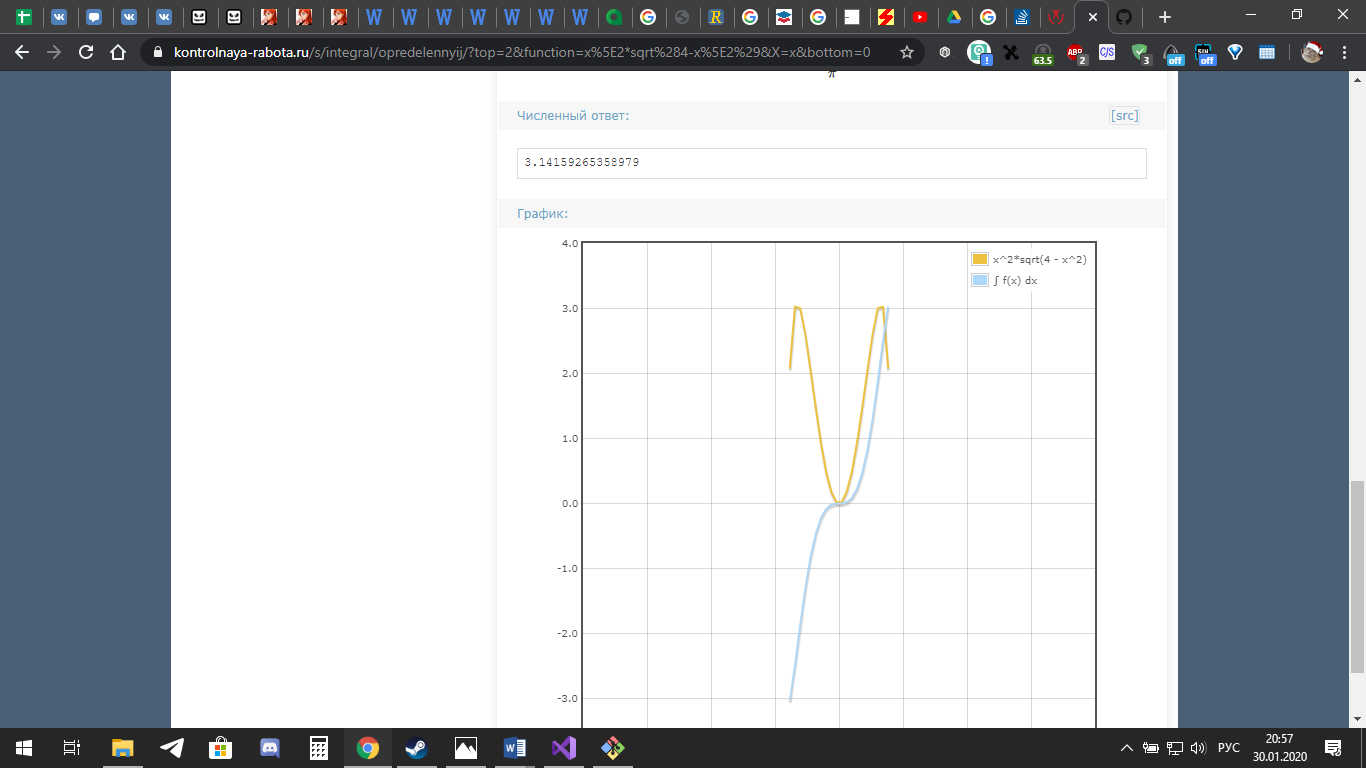


Рисунок 6 – Ответ в онлайн калькуляторе

4 Заключение

Были достигнуты цели курсовой работы, написание консольного приложения на языке высокого уровня С#.Были изучены методы создания программ, изучена среда разработки вижуал студия, изучены методы решения определенныхз интегралов.

Список использованных источников

1. Документация по C# [Электронный ресурс] https://docs.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения:  30.01.2020)
2. Джозеф Албахари, Бен Албахари С# 6.0. Карманный справочник Изд-во Вильямс, 2016 г
3. C# для начинающих Пахомов Б.И. Изд-во «БХВ-Петербург» 2014 г

Приложение

