Изображение выглядит как цепь, медальон

Автоматически созданное описание

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.03.01 | Информатика и вычислительная техника | | |
| Направленность (профиль) | |  | Системы автоматизированного проектирования | | |
| Форма обучения | |  | очная | | |
|  | |  |  | | |
| Факультет | |  | Информационных технологий и управления | | |
| Кафедра | |  | Систем автоматизированного проектирования и управления | | |
| Учебная дисциплина | |  | Программирование | | |
| Курс | II | | | Группа | 414 |

Отчёт по контрольной работе № 1

Вариант № 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Исполнитель: |  |  |  |  |
| обучающийся группы 414 |  |  |  | Кондиляброва  Вероника Данииловна |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |
| Проверили: |  |  |  | Корниенко Иван Григорьевич |
|  |  | (дата, подпись) |  | Федин Алексей Константинович |

Содержание

[1 Постановка задачи 3](#_Toc114327844)

[2 Исходные данные 3](#_Toc114327845)

[3 Особые ситуации 3](#_Toc114327846)

[4 Математические методы и алгоритмы решения задач 3](#_Toc114327847)

[5 Форматы представления данных 4](#_Toc114327848)

[6 Структура программы 4](#_Toc114327849)

[7 Описание хода выполнения 5](#_Toc114327850)

[8 Блок-схема алгоритма решения задачи 6](#_Toc114327851)

[9 Результаты работы программы 11](#_Toc114327852)

[10 Исходный код полученного программного решения 16](#_Toc114327853)

## 1 Постановка задачи

Вычислить интеграл заданной функции методом трапеций и методом парабол.

## 2 Исходные данные

В качестве исходных данных программа использует коэффициенты параболы a, b и c, а также интервал нахождения интеграла. Ввод с клавиатуры, из файла.

## 3 Особые ситуации

1 Коэффициент уравнения при x2 не должен быть равен нулю, если пользователь введёт ноль, программа выведет сообщение с просьбой ввести значение заново.

2 Если человек ввёл верхнюю границу меньше нижней границы, программа выведет сообщение с просьбой ввести верхнюю границу еще раз.

## 4 Математические методы и алгоритмы решения задач

Согласно постановке задачи для составления программы будут использованы метод трапеций и параболы, первый заключается в разбиении отрезка на несколько промежуточных отрезков и приближения графика ломаной линией. Таким образом, площадь приближается суммой площадей трапеций. Метод парабол заключается в приближении не ломаной линией, а маленькими параболами. Сколько промежутков – столько и маленьких парабол.

Метод трапеций:

Есть определенный интеграл , где – функция на [a, b]. Нужно провести разбиение интервала [a, b] на n равных отрезков: [x0, x1], …, [xn-1, xn]. При этом x0 это нижняя граница интервала [a, b], а xn верхняя.

Тогда определенный интеграл можно вычислить по формуле:

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |

Метод парабол:

Есть определенный интеграл , где – функция на [a, b]. Нужно провести разбиение интервала [a, b] на 2n равных отрезков: [x0, x1], …, [x2n-1, x2n]. При этом x2n-1 это нижняя граница интервала [a, b], а x2n верхняя.

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |

Тогда определенный интеграл можно вычислить по формуле:

То есть сумму четных элементов умножаем на 2, а сумму нечетных элементов умножаем на 4, не включая граничные точки.

## 5 Форматы представления данных

Таблица 1 – Переменные, используемы в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| get\_result | bool | Проверка правильного ввода с файла |
| e | double | Точность вычисления интеграла |
| n | int | Количество отрезков |
| a | double | Коэффициент параболы при x2 |
| b | double | Коэффициент параболы при x |
| c | double | Коэффициент параболы |
| x1 | double | Нижняя граница интервала |
| x2 | double | Верхняя граница интервала |
| end | int | Выход из программы |
| user\_choice | int | Выбор пункта меню |

Таблица 2 – Константы, используемые в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| e | double | Точность вычисления интеграла |
| n | int | Количество отрезков |

## 6 Структура программы

Таблица 3 – Функции, составляющие программу

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| Input\_choice | Проверка ввода выбора сохранения в файл |
| Input\_menu | Проверка ввода пункта меню |
| Input\_a | Проверка ввода коэффициента при x2 |
| Input\_coef | Проверка ввода коэффициентов параболы |
| Input\_limits | Проверка ввода верхней границы |
| Save\_initial\_data | Сохранение в файл исходных данных |
| Save\_f | Сохранение в файл результатов |
| trapezoid\_int | Вычисление интеграла методом трапеций |
| parabola\_int | Вычисление интеграла методом парабол |

## 7 Описание хода выполнения

1. В ходе работы было создано решение в интегрированной среде разработки Microsoft Visual Studio 2022. В нём был создан проект.
2. На этапе изучения математического метода я столкнулась с множеством формул для метода парабол, разобравшись, я поняла, что все формулы отличаются друг друга очень маленькой погрешностью, а значит можно брать любую.
3. В начальном варианте программы не была учтена разница между верхней и нижней границей, так как в промежутке левое значение должно быть меньше правого, я сделала дополнительную проверку границ.
4. Также в начальном варианте количество отрезков, начиная с 10, увеличивалось на 10 2 раза, этого не хватало для разницы погрешности равной 0,001, поэтому начальным количеством отрезков стало 10, а следующие умножаются на 10, последнее количество отрезков равно 1000.

## 8 Блок-схема алгоритма решения задачи

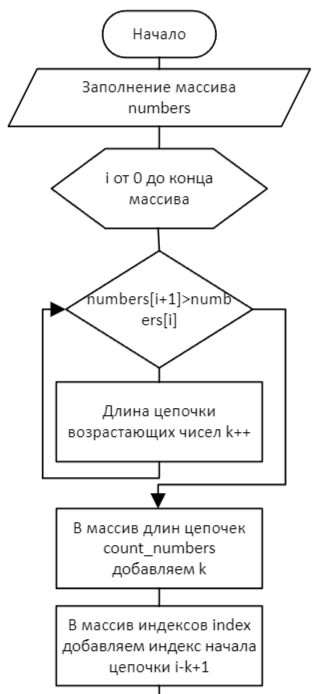
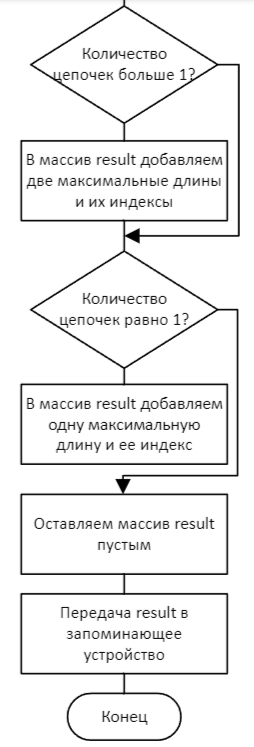


Рисунок 1 – Блок-схема алгоритма нахождения интеграла методом трапеций

## 9 Результаты работы программы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – Скриншот работы программы без работы с файлами

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Скриншот работы программы без работы с файлами

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – Скриншот проверки коэффициента при x2 в файле

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – Скриншот проверки на ввод всех значений

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 6 – Скриншот чтения и записи в файл

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 – Скриншот теста

## 10 Исходный код полученного программного решения

// Практическая работа №1.

// Вычислить интеграл функции ax^2 + bx + c методом трапеций и методом парабол.

// Студент группы 414, Кондиляброва Вероника Данииловна. 2023 год

using System.IO;

using System.Numerics;

using System.Text;

using System.Xml.Linq;

using static System.Net.Mime.MediaTypeNames;

public class find\_integral

{

static int Input\_choice()

{

bool inputSuccess = false;

int input = 0;

while (!inputSuccess)

{

string consoleInput = Console.ReadLine();

if (!int.TryParse(consoleInput, out input))

{

Console.WriteLine(

"Число a должно быть числом.");

continue;

}

if (input > 2 || input < 1)

{

Console.WriteLine(

"Попробуйте еще раз:");

continue;

}

inputSuccess = true;

}

return input;

}

static int Input\_menu()

{

bool inputSuccess = false;

int input = 0;

while (!inputSuccess)

{

string consoleInput = Console.ReadLine();

if (!int.TryParse(consoleInput, out input))

{

Console.WriteLine(

"Число a должно быть числом.");

continue;

}

if (input > 4 || input < 1)

{

Console.WriteLine(

"Попробуйте еще раз:");

continue;

}

inputSuccess = true;

}

return input;

}

static double Input\_a()

{

bool inputSuccess = false;

double input = 0;

while (!inputSuccess)

{

string consoleInput = Console.ReadLine();

if (!double.TryParse(consoleInput, out input))

{

Console.WriteLine(

"Число a должно быть числом.");

Console.Write("a = ");

continue;

}

if (input == 0)

{

Console.WriteLine("Число a не может быть нулём.");

Console.Write("a = ");

continue;

}

inputSuccess = true;

}

return input;

}

static double Input\_coef()

{

bool inputSuccess = false;

double input = 0;

while (!inputSuccess)

{

string consoleInput = Console.ReadLine();

if (!double.TryParse(consoleInput, out input))

{

Console.WriteLine(

"Число a должно быть числом.");

continue;

}

inputSuccess = true;

}

return input;

}

static double Input\_limits(double x)

{

bool inputSuccess = false;

double x2 = 0;

while (!inputSuccess)

{

Console.Write("x2 = ");

string Input\_x2 = Console.ReadLine();

if (!double.TryParse(Input\_x2, out x2))

{

Console.WriteLine(

"Число x2 должно быть числом.");

continue;

}

x2 = double.Parse(Input\_x2);

if (x2 <= x)

{

Console.WriteLine(

"Верхняя граница x2 должна быть больше нижней границы x1.");

continue;

}

inputSuccess = true;

}

return x2;

}

public static void Main()

{

Console.WriteLine("Кондиляброва Вероника\n414 группа\nВариант 8\nВычислить интеграл заданной функции методом трапеций и методом парабол.");

bool get\_result=true;

const double e = 0.001;

const int n = 10;

double a = 0;

double b = 0;

double c = 0;

double x1 = 0;

double x2 = 0;

int end = 0;

while (end != 1)

{

Console.WriteLine("Ввести данные вручную - 1, считать данные из файла - 2, получить значение площади - 3, выйти из программы - 4: ");

int user\_choice = 0;

user\_choice = Input\_menu();

switch (user\_choice)

{

case (int)Menu.Mabual\_Input:

{

Console.WriteLine("Введите коэффициенты уравнения параболы: ");

Console.Write("a = ");

a = Input\_a();

Console.Write("b = ");

b = Input\_coef();

Console.Write("c = ");

c = Input\_coef();

Console.WriteLine("Введите границы параболы: ");

Console.Write("x1 = ");

x1 = Input\_coef();

x2 = Input\_limits(x1);

Save\_initial\_data(a, b, c, x1, x2);

}

break;

case (int)Menu.File\_Input:

{

Console.Write("Введите название файла: ");

//string s;

string path = Console.ReadLine();

StreamReader f = new StreamReader(path);

if (!f.EndOfStream)

{

string s = f.ReadLine();

a = int.Parse(s);

// что-нибудь делаем с прочитанной строкой s

}

if (!f.EndOfStream)

{

string s = f.ReadLine();

b = int.Parse(s);

// что-нибудь делаем с прочитанной строкой s

}

if (!f.EndOfStream)

{

string s = f.ReadLine();

c = int.Parse(s);

// что-нибудь делаем с прочитанной строкой s

}

else

{

get\_result = false;

}

if (!f.EndOfStream)

{

string s = f.ReadLine();

x1 = int.Parse(s);

// что-нибудь делаем с прочитанной строкой s

}

else

{

get\_result = false;

}

if (!f.EndOfStream)

{

string s = f.ReadLine();

x2 = int.Parse(s);

if (x2 <= x1)

{

Console.WriteLine("Верхняя граница x2 должна быть больше нижней границы x1.");

get\_result = false;

}

// что-нибудь делаем с прочитанной строкой s

}

else

{

get\_result = false;

}

if (a != 0 && b != 0 && get\_result != false) { Console.WriteLine("Значения считаны."); }

else

{

Console.WriteLine("Файл заполнен неверно.");

}

f.Close();

}

break;

case (int)Menu.Get\_result:

{

if(a==0 && b==0)

{

Console.WriteLine("Сначала введите данные.");

break;

}

if(get\_result== false)

{

Console.WriteLine("Файл заполнен неверно.");

break;

}

if (Math.Abs(trapezoid\_int(a, b, c, x1, x2, n) - parabola\_int(a, b, c, x1, x2, n)) > e)

{

if (Math.Abs(trapezoid\_int(a, b, c, x1, x2, n \* 10) - parabola\_int(a, b, c, x1, x2, n \* 10)) > e)

{

Console.WriteLine("Максимально приближенное значение площади с количеством отрезков 1000: ");

Console.Write("Метод трапеций: ");

Console.WriteLine(trapezoid\_int(a, b, c, x1, x2, n \* 100));

Console.Write("Метод парабол: ");

Console.WriteLine(parabola\_int(a, b, c, x1, x2, n \* 100));

Save\_f(a, b, c, x1, x2, n \* 100);

}

else

{

Console.WriteLine("Максимально приближенное значение с количеством отрезков 100: ");

Console.Write("Метод трапеций: ");

Console.WriteLine(trapezoid\_int(a, b, c, x1, x2, n \* 10));

Console.Write("Метод парабол: ");

Console.WriteLine(parabola\_int(a, b, c, x1, x2, n \* 10));

Save\_f(a, b, c, x1, x2, n \* 10);

}

}

else

{

Console.WriteLine("Максимально приближенное значение с количеством отрезков 10: ");

Console.Write("Метод трапеций: ");

Console.WriteLine(trapezoid\_int(a, b, c, x1, x2, n));

Console.Write("Метод парабол: ");

Console.WriteLine(parabola\_int(a, b, c, x1, x2, n));

Save\_f(a, b, c, x1, x2, n);

}

}

break;

case (int)Menu.Exit:

{

end = 1;

}

break;

default:

Console.WriteLine(

"Ввести данные вручную - 1, считать данные из файла - 2, получить значение площади - 3, выйти из программы - 4: ");

break;

}

}

}

static void Save\_initial\_data(double a, double b, double c, double x1, double x2)

{

Console.WriteLine("Сохранить в файл? Да - 1, Нет - 2");

int choice\_save = Input\_choice();

if (choice\_save == 1)

{

Console.WriteLine("Добавить в существующий файл или создать новый - 1, Перезаписать существующий файл - 2");

choice\_save = Input\_choice();

if (choice\_save == 1)

{

Console.Write("Введите название файла: ");

string path = Console.ReadLine();

string str\_a = a.ToString();

string str\_b = b.ToString();

string str\_c = c.ToString();

string str\_x1 = x1.ToString();

string str\_x2 = x2.ToString();

string initial\_data = str\_a + "\n" + str\_b + "\n" + str\_c + "\n" + str\_x1 + "\n" + str\_x2;

// добавление в файл

using (StreamWriter writer = new StreamWriter(path, true))

{

writer.WriteLineAsync(initial\_data);

}

Console.WriteLine("Сохранено");

}

else

{

Console.Write("Введите название файла: ");

string path = Console.ReadLine();

string str\_a = a.ToString();

string str\_b = b.ToString();

string str\_c = c.ToString();

string str\_x1 = x1.ToString();

string str\_x2 = x2.ToString();

string initial\_data = str\_a + "\n" + str\_b + "\n" + str\_c + "\n" + str\_x1 + "\n" + str\_x2;

// полная перезапись файла

using (StreamWriter writer = new StreamWriter(path, false))

{

writer.Write(initial\_data);

}

Console.WriteLine("Сохранено");

}

}

}

static void Save\_f(double a, double b, double c, double x1, double x2, int n)

{

Console.WriteLine("Сохранить в файл? Да - 1, Нет - 2");

int choice\_save = Input\_choice();

if (choice\_save == 1)

{

Console.WriteLine("Добавить в существующий файл или создать новый - 1, Перезаписать существующий файл - 2");

choice\_save = Input\_choice();

if (choice\_save == 1)

{

Console.Write("Введите название файла: ");

string path = Console.ReadLine();

string integ = trapezoid\_int(a, b, c, x1, x2, n).ToString();

integ += "\n";

integ += parabola\_int(a, b, c, x1, x2, n).ToString();

// добавление в файл

using (StreamWriter writer = new StreamWriter(path, true))

{

writer.WriteLineAsync(integ);

}

Console.WriteLine("Сохранено");

}

else

{

Console.Write("Введите название файла: ");

string path = Console.ReadLine();

string integ = trapezoid\_int(a, b, c, x1, x2, n).ToString();

integ += "\n";

integ += parabola\_int(a, b, c, x1, x2, n).ToString();

// полная перезапись файла

using (StreamWriter writer = new StreamWriter(path, false))

{

writer.Write(integ);

}

Console.WriteLine("Сохранено");

}

}

}

public static double trapezoid\_int(double a, double b, double c, double x1, double x2, int n)

{

//n=10

//I=h\*( (x10 - x0)/2 + (x1 + x2 +...+ x9)

double x = 0;

double func\_sum = 0;

double I = 0;

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

{

x = x + (x2 - x1) / n;

func\_sum = a \* (double)Math.Pow(x, 2) + b \* x + c + func\_sum;

}

double func\_x1 = 0;

func\_x1 = a \* (double)Math.Pow(x1, 2) + b \* x1 + c;

double func\_x2 = 0;

func\_x2 = a \* (double)Math.Pow(x2, 2) + b \* x2 + c;

I = (x2 - x1) / n \* (((func\_x2 + func\_x1) / 2) + func\_sum);

return Math.Abs(I);

}

public static double parabola\_int(double a, double b, double c, double x1, double x2, int n)

{

//n=10

//I=h/3\*(x0 + x10 + 2 \* (x2 + x4 + x6 + x8) + 4 \* (x1 + x3 + x5 + x7 + x9)

double x\_even = (x2 - x1) / n;

double x\_odd = 2 \* (x2 - x1) / n;

double func\_odd = 0;

double func\_even = 0;

double I = 0;

for (int i = 0; i < n / 2; i++)

{

func\_even = a\* (double)Math.Pow(x\_even, 2) + b \* x\_even + c + func\_even;

x\_even = x\_even + 2 \* (x2 - x1) / n;

}

for (int i=1; i < n / 2; i++)

{

func\_odd = a \* (double)Math.Pow(x\_odd, 2) + b \* x\_odd + c + func\_odd;

x\_odd = x\_odd + 2 \* (x2 - x1) / n;

}

double h = 0;

h = (x2 - x1) / (n \* 3);

double func\_x1 = 0;

func\_x1 = a \* (double)Math.Pow(x1, 2) + b \* x1 + c;

double func\_x2 = 0;

func\_x2 = a \* (double)Math.Pow(x2, 2) + b \* x2 + c;

I = h \* (func\_x1 + func\_x2 + 2 \* func\_odd + 4 \* func\_even);

return Math.Abs(I);

}

enum Menu : int

{

Mabual\_Input = 1,

File\_Input,

Get\_result,

Exit

}

}

//Тест

namespace TestIntegral

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void integ()

{

var actual = find\_integral.trapezoid\_int(1, 1, 1, 0, 10, 1000);

Assert.AreEqual(393.3334999999895, actual);

actual = find\_integral.parabola\_int(1, 1, 1, 0, 10, 1000);

Assert.AreEqual(393.33333333332723, actual);

actual = find\_integral.trapezoid\_int(1, 2, 0, -5, 0, 1000);

Assert.AreEqual(66.61668749999838, actual);

actual = find\_integral.parabola\_int(1, 2, 0, -5, 0, 1000);

Assert.AreEqual(66.63333333333242, actual);

actual = find\_integral.trapezoid\_int(2, 3, 1, -10, -5, 1000);

Assert.AreEqual(126.18337499999696, actual);

actual = find\_integral.parabola\_int(2, 3, 1, -10, -5, 1000);

Assert.AreEqual(126.06666666666493, actual);

actual = find\_integral.trapezoid\_int(1, 1, 1, 0, 2, 100);

Assert.AreEqual(6.666800000000002, actual);

actual = find\_integral.parabola\_int(1, 1, 1, 0, 2, 100);

Assert.AreEqual(6.666666666666669, actual);

actual = find\_integral.trapezoid\_int(-2, 1, 1, 0, 5, 1000);

Assert.AreEqual(65.83337499999773, actual);

actual = find\_integral.parabola\_int(-2, 1, 1, 0, 5, 1000);

Assert.AreEqual(65.83333333333198, actual);

}

}

}