

РОСЖЕЛДОР
федеральное государственное бюджетное образовательное учреждение высшего образования
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ»
(СГУПС)

Кафедра «Информационные технологии транспорта»

Отчет по учебной практике

Проверил:
старший преподаватель

_____ Э. А. Усова
подпись

дата проверки

Выполнил:
студент гр. БИСТ-112

_____ А. Ю. Коровченко
подпись

дата сдачи на проверку

Краткая рецензия:

запись о допуске к защите

оценка по результатам защиты

подписи преподавателей и дата защиты

Новосибирск
2021

РОСЖЕЛДОР
федеральное государственное бюджетное образовательное учреждение высшего образования
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ»
(СГУПС)
Кафедра «Информационные технологии транспорта»

Задание на учебную практику

студенту Коровченко Алексею Юрьевичу Группа БИСТ-112
Вариант № 18

Тема: Численные методы.

Цель учебной практики: приобретение и закрепление профессиональных умений и навыков практической работы в области информационных технологий.

Вариант задания: Численное интегрирование. Метод трёх восьмых (формула Ньютона)

Сроки сдачи на проверку программного обеспечения: 3 неделя учебной практики.

Сроки сдачи на проверку отчета: за 3 дня до окончания учебной практики.

Сроки защиты: 4 неделя учебной практики.

Работу оформить в соответствии со стандартом организации СТО СГУПС 1.01 БИ.01-2019 «Система менеджмента качества. Письменная отчетная работа. Требования к оформлению».

График выполнения

Название документа и раздела	Ориентировочно		График выполнения (недели)
	колич. страниц записки	трудоемкость в часах	
Аналитический обзор, изучение численного метода	2	6	1
Построение математической модели, разработка алгоритма численного метода и комплекта тестов	3	4	1
Реализация и тестирование программного обеспечения	5	20	1-2
Написание пояснительной записки и разработка руководства пользователя	4	6	2
<i>Общая трудоемкость</i>	<i>15</i>	<i>36</i>	<i>-</i>

Основная литература:

1 Турчак Л.И., Плотников П.В. Основы численных методов: Учебное пособие.-2-е изд., перераб. и доп. – М.: ФИЗМАТЛИТ, 2002.-304 с.

2 Сайт СЭОР СГУПС – Система электронных образовательных ресурсов сибирского государственного университета путей сообщения (СГУПС) Учебно-методический комплекс дисциплины «Учебная практика». / Э.А. Усова. – Новосибирск : [сайт]. – URL: <http://moodle3.stu.ru/course/view.php?id=4181#section-5> (дата обращения: 28.06.2020). – Режим доступа: для авториз. пользователей. – Текст. Изображение : электронные.

Задание выдано «28» июня 2021 г.

Руководитель _____

/Э.А. Усова

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Постановка задачи.....	5
2 Описание теоретического материала	6
3 Инструмент разработки	8
4 Разработка приложения	9
4.1 Проектирование графического интерфейса приложения	9
4.2 Описание реализации индивидуальной части задания	9
4.3 Алгоритм метода вычисления значения интеграла	10
5 Тестирование, анализ результатов	12
5.1 Расчет тестового примера	12
5.2 Сравнение результатов	15
5.3 Тестирование программного обеспечения на ошибки.....	17
ЗАКЛЮЧЕНИЕ	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	21
ПРИЛОЖЕНИЕ А	22

ВВЕДЕНИЕ

Целью учебной практики является приобретение и закрепление профессиональных умений и навыков практической работы в области информационных технологий; изучение численных методов решения практических задач и возможных источников погрешностей при реализации данных методов с использованием языка программирования высокого уровня.

Целью данной работы является разработка программного обеспечения вычисления значения определенного интеграла формулой трёх восьмых (Формула Ньютона) с использованием графического интерфейса.

Для достижения поставленной цели необходимо решить следующие задачи:

- постановка задачи и построение математической модели;
- изучение численного метода решения поставленной задачи;
- разработка алгоритма программы и построение блок-схемы;
- разработка тестового примера;
- выбор инструмента разработки;
- проектирование графического интерфейса;
- программирование и отладка программы;
- проведение численного эксперимента на компьютере и анализ полученных результатов.

Данная работа состоит из теоретической и практической части. В теоретической части рассмотрен алгоритм численного метода вычисления значения определенного интеграла формулой трёх восьмых (Формулой Ньютона). В практической части – разработка программного обеспечения, тестирование созданного графического приложения и составление руководства пользователя.

В данной работе используется среда разработки Microsoft Visual Studio 2019.

Язык программирования – C#. Язык вёрстки – XAML.

1 Постановка задачи

Разработать графическое приложение для вычисления определённого интеграла методом трёх восьмых (формула Ньютона).

В программе должно быть реализовано:

- обработка исключений, возникающих из-за ошибочного ввода данных пользователем;
- ввод функции и границ определенного интеграла;
- вычисление значения на основе заданной точности пользователем;
- вывод погрешности метода;
- вывод результата работы программы;
- вывод окон информации о программе и её создателе.

2 Описание теоретического материала

Формула трёх восьмых (кубических парабол) – это формула численного интегрирования для вычисления определённых интегралов с заданной точностью. В некоторой литературе именуется как «Формула Ньютона».

Для нахождения определенного интеграла по данной формуле необходимо подсчитать значение шага исходя из границы начала и конца интервала, для этого используется формула (2.1).

$$h = \frac{b-a}{n}, \quad (2.1)$$

где h – шаг;

b – значение конца интервала;

a – значение начала интервала;

n – количество шагов.

Зная границы интервала и шаг вычисляются значения, которые принимают функция, на этом интервале. Для этого необходимо вычислить значение неизвестной переменной при определенном значении на интервале, для этого применяется формула (2.2).

$$x_i = a + ih, \quad (2.2)$$

где x_i – значение неизвестной переменной на интервале;

a – значение начала интервала;

i – номер итерации, принимаемый значения от 0 до значения конца интервала;

h – шаг, вычисляемый по формуле (2.1).

Главным условием при вычислении значения интеграла данным методом является кратность количества шагов трём так как формула Ньютона получается из построения формулы Ньютона-Котеса интерполированием функции $f(x)$ в четырёх точках с условием, что n кратно трём, то есть вычисляется по формуле $n = 3m$.

Для подсчета значения также требуется знать значение константы k_i , которое зависит от номера слагаемого (значения i). Если номер текущего

элемента кратен трём, то константа имеет значение два иначе если равен началу или концу интервала, то константа равна одному иначе константа равна трём. Подсчет происходит по формуле (2.3).

$$k_i = \begin{cases} 1, i = 0 \\ 3, i = \overline{3t - 2, 3t - 1}, t = \overline{1, \frac{n}{3}} \\ 2, i = \overline{3t}, t = \overline{1, \frac{n}{3} - 1} \\ 1, i = n \end{cases} \quad (2.3)$$

Нахождение значения определенного интеграла выполняется по формуле (2.4).

$$I_{\frac{3}{8}n} = \frac{3h}{8} \sum_{i=0}^n k_i f(x_i), \quad (2.4)$$

где $I_{\frac{3}{8}n}$ – искомое значение определенного интеграла;

n – количество шагов кратное трём;

k_i – константа, принимающая значение по формуле (2.3);

h – шаг, вычисляемый по формуле (2.1);

x_i – значение переменной на интервале.

Для подсчета погрешности применяется «Правило Рунге». Чтобы воспользоваться данным правилом необходимо найти значение численного метода для начала с числом шагов равным n , а затем с числом шагов равным $\frac{n}{2}$. Полученные значения необходимо вычесть и умножить на $\frac{1}{15}$.

Подсчет погрешности происходит по формуле (2.5).

$$\Delta_n \approx \frac{1}{15} |I_n - I_{\frac{n}{2}}|, \quad (2.5)$$

где Δ_n – погрешность при количестве шагов равным n ;

I_n – значение интеграла при количестве шагов равным n ;

$I_{\frac{n}{2}}$ – значение интеграла при количестве шагов равным $\frac{n}{2}$.

3 Инструмент разработки

Необходимо разработать графическое приложение WPF. При первом запуске изначально в программе записан тестовый пример. Пользователь может изменить подынтегральную функцию, пределы интегрирования и точность на свои записав значения в определенные TextBox. Данные значения записываются в f – строковое представление подынтегральной функции, a – нижний предел интегрирования, b – верхний предел интегрирования, eps – заданная точность. При вводе недопустимых значений (например, вместо численного значения предела ввести набор букв) программа выдает ошибку, с помощью MessageBox, о необходимости исправить данную оплошность. После корректного ввода всех значений программа выводит в специальные TextBox результат численного метода и его погрешность. В случае если была задана сложная функция и вычисление производится с задержкой имеется ProgressBar отмечающий процесс выполнения вычислений. Также в процессе вычисления можно прервать процесс специальной кнопкой внизу приложения.

Для просмотра информации о программе и ее создателе имеются две кнопки (Button) с текстом «О приложении» и «Помощь» в которых содержится вся необходимая для пользователя информация.

В правом углу приложения с помощью TextBlock указаны все возможные функции, которые можно ввести для подсчета значения интеграла.

4 Разработка приложения

4.1 Проектирование графического интерфейса приложения

Для решения поставленной задачи требуется спроектировать интерфейс WPF-приложения: предусмотреть элементы управления для ввода подынтегральной функции, значений пределов интегрирования и погрешности. Реализовать элементы для начала вычисления значения и его остановки в случае долгого процесса. Для кнопок добавить обработчики событий, которые вызывают необходимые методы в программе.

Графический интерфейс предоставлен на рисунке 4.1.

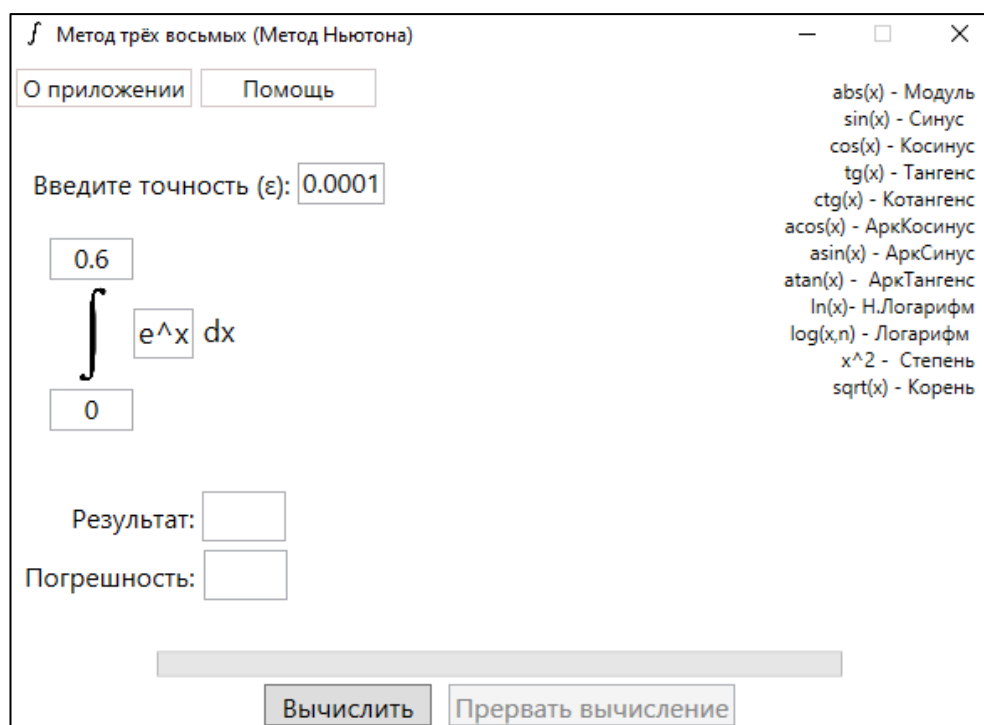


Рисунок 4.1 – Графический интерфейс WPF-приложения

4.2 Описание реализации индивидуальной части задания

Изначально данные с полей FormulaTextBox, VTextBox.Text, ATextBox.Text и EpsTextBox.Text заносятся в переменные f, b, a и eps соответственно. После этого передаются в метод CalcThread.

4.3 Алгоритм метода вычисления значения интеграла

В начале метода создаются переменные `res` – промежуточная погрешность, и `finaly` – значение численного метода при текущей погрешности. Изначально n равняется трём (обязательное условие кратности трём). Инициализируется переменная `sum`, которая изначально равна нулю, для записи результата суммы в формуле. Действия для получения результата следующие:

4.3.1 Проверяется текущая погрешность результата с заданной пользователем. Если текущая погрешность больше заданной, то продолжаются вычисления, иначе погрешность и значение метода выводятся на экран и метод завершает свою работу. При первом запуске метода текущая погрешность равняется одному, что позволяет в любом случае попасть в цикл;

4.3.2 Вычисляется значение шага по формуле (2.1). Переменная `sum` сбрасывается в значение ноль;

4.3.3 Запускается цикл, производящий операции до n включительно, начиная с нуля. Подсчитывается текущее значение x_i по формуле (2.2) и выполняется подсчет введенной функции в переменной f при данном значении x . После чего если номер итерации цикла равен нулю или n , то значение прибавляется к `sum`, иначе если номер итерации кратен трём, то к `sum` прибавляется значение, умноженное на два иначе к `sum` прибавляется значение умноженное на три. Если `sum` имеет значение `null`, значит функция была введена неверно.

Фрагмент кода данного пункта предоставлен ниже;

```
float? sum = 0;
for (int i = 0; i <= n; i++)
{
    float xi = a + h * i;
    if (i == 0 || i == n) sum += CalcFunc(f, xi);
    else if (i % 3 == 0) sum += 2 * CalcFunc(f, xi);
    else sum += 3f * CalcFunc(f, xi);
    Pg1.Dispatcher.Invoke(() => Pg1.Value += progress);
    if (sum == null)
    {
```

```

        _calc = null;
        MessageBox.Show("Корректно введите функцию");
        return;
    }
}

```

4.3.4 После подсчета суммы необходимо в соответствии с формулой (2.4) выполнить умножение суммы на $\frac{3h}{8}$. Полученный результат считается конечным при текущем значении n и записывается в переменную `finaly`;

4.3.5 Необходимо подсчитать погрешность с помощью «Правил Рунге». Для этого необходимо выполнить пункт 4.2.3, но значение переменной n зададим равной: $n = \frac{n}{2}$. Получив результат при данном значении n вычислить погрешность по формуле (2.5) и записать результат в переменную `res`. Затем увеличить переменную n на три для подсчета значения численного метода в случае не прохождения условия заданной погрешности. Фрагмент кода данного пункта предоставлен ниже;

```

float h2 = (b - a) / (n / 2f);
float? sum2 = 0;
for (int i = 0; i <= n / 2f; i++)
{
    float xi2 = a + i * h2;
    if (i == 0 || i == n / 2) sum2 += CalcFunc(f, xi2);
    else if (i % 3 == 0) sum2 += 2 * CalcFunc(f, xi2);
    else sum2 += 3 * CalcFunc(f, xi2);
}
float fin2 = (float)(sum2 * (3f * h2 / 8f));
res = Math.Abs(fin1 - fin2) * (1f / 15f);

```

4.3.6 Перейти с текущим результатом к пункту 4.2.1, где необходимо проверить погрешность с заданной и в случае выполнения условия вывести результат и погрешность на экран.

5 Тестирование, анализ результатов

5.1 Расчет тестового примера

Изначально необходимо выполнить расчёт значения метода вручную для возможности сравнения с программным решением. Наиболее точный метод можно найти применяя формулу Ньютона-Лейбница. Согласно данной формуле, определенный интеграл равен разности значений первообразной на верхнем и нижнем пределах интегрирования:

$$\int_0^{0,6} e^x dx = e^x \Big|_0^{0,6} = e^{0,6} - e^0 = 1,82211880039 - 1 = 0,82211880039$$

Значение определенного интеграла формулой Ньютона-Лейбница найдено. Следующим шагом необходимо произвести расчет с помощью формулы трёх восьмых (2.4). Изначальное значение n взять равное трём. Значение шага: $h = \frac{0,6-0}{3} = 0,2$. $f(x) = e^x$. Также необходимо задать погрешность, например $\epsilon = 0.0001$.

Все значения, необходимые для формулы, предоставлены в таблице 5.1.

Таблица 5.1 – Таблица значений численного метода при $n = 3$

i	x_i	$f(x_i)$
0	0	1
1	0,2	1,22140275816
2	0,4	1,49182469764
3	0,6	1,82211880039

Итоговое значение формулы:

$$I_{\frac{3}{8}} = \frac{3 * 0,2}{8} (1 + 3 * 1,22140275816 + 3 * 1,49182469764 + 1,82211880039) = \frac{3 * 0,2}{8} * 10,9618011678 = 0,82213508758$$

Полученное значение интеграла, используя численный метод при значении $n = 3$, равняется 0,82213508758. Требуется вычислить значение погрешности с помощью правила Рунге. Для этого необходимо количество шагов поделить на два. Можно уже на данном этапе сказать, что значения погрешности будет велико, ведь формула трёх восьмых вычисляет значения при количестве шагов кратным трём. Если поделить текущее количество шагов на два, то получится число не кратное трём, что вызовет большую погрешность в соответствии с формулой (2.4) и правилами изложенными в пункте 2.

Для примера ниже предоставлено вычисление значения при n делённым пополам и значение погрешности при изначальном количестве шагов равным трём.

$$\text{Вычисляется значение шага: } h = \frac{b-a}{n/2} = \frac{0.6}{1.5} = 0.4, f(x) = e^x.$$

Все значения, используемые в формуле, предоставлены в таблице 5.2.

Таблица 5.2 – Таблица значений численного метода при $n = 1,5$

i	x_i	$f(x_i)$
0	0	1
1	0,4	1.4918246976412703

Итоговое значение интеграла при данном количестве шагов:

$$\begin{aligned} I_{\frac{3}{8}1,5} &= \frac{3 * 0.4}{8} (1 + 1.4918246976412703) = \frac{3 * 0.4}{8} * 2,49182469764 \\ &= 0,37377370464 \end{aligned}$$

$$\begin{aligned} \text{Погрешность в данном случае равняется: } \Delta_n &\approx \frac{1}{15} |0,82213508758 - \\ &0,37377370464| = \frac{1}{15} * 0,44836138294 = 0,02989075886. \end{aligned}$$

Значения погрешности выше, чем заданное значение эпсилон изначальное. Необходимо увеличить количество шагов на три и произвести вычисление формулой трех восьмых при новом значении.

Теперь количество шагов $n = 6$. Необходимо произвести те же самые действия для этого количества шагов. При этом для вычисления погрешности новое количество шагов n равняется трем.

Вычисляется значение шага: $h = \frac{0,6-0}{6} = 0,1$, $f(x) = e^x$.

Все значения, для подсчета значения при $n = 6$, предоставлены в таблице 5.3.

Таблица 5.3 – Таблица значений численного метода при $n = 6$

i	x_i	$f(x_i)$
0	0	1
1	0,1	1,10517091808
2	0,2	1,22140275816
3	0,3	1,34985880758
4	0,4	1,49182469764
5	0,5	1,6487212707
6	0,6	1,82211880039

На основе полученных значений вычисляется значение интеграла с данным количеством шагов:

$$\begin{aligned}
 I_{\frac{3}{8}^6} &= \frac{3 * 0,1}{8} (1 + 3 * 1,10517091808 + 3 * 1,22140275816 + 2 \\
 &\quad * 1,34985880758 + 3 * 1,49182469764 + 3 * 1,6487212707 \\
 &\quad + 1,82211880039) = \frac{3 * 0,1}{8} * 21,9231953493 = 0,82211982559
 \end{aligned}$$

Для вычисления погрешности используется значение численного метода при $n = 3$ и $n = 6$.

$$\begin{aligned}
 \Delta_n &\approx \frac{1}{15} |0,82211982559 - 0,82213508758| = \frac{1}{15} * 0,00001526199 \\
 &= 0,00000101746
 \end{aligned}$$

Исходя из полученных данных видно, что значение погрешности при $n = 6$ меньше, чем значение эпсилон, а значит ответом на данный тестовый пример будет значение интеграла равное 0,82211982559.

5.2 Сравнение результатов

При запуске программы значения тестового примера изначально вводятся в необходимые поля на экране. Поставлена задача проверить значение, выдаваемое программным продуктом, и тестовым примером из пункта 5.1. Для начала необходимо проверить, что все значения введены верно. Верхний предел интегрирования – 0.6, нижний предел – 0, подынтегральная функция - e^x и необходимая точность 0.0001. Введенные данные предоставлены на рисунке 5.1.

Метод трёх восьмых (Метод Ньютона)

О приложении Помощь

Введите точность (ϵ): 0.0001

0.6

$\int e^x dx$

0

Результат:

Погрешность:

Вычислить Прервать вычисление

abs(x) - Модуль
sin(x) - Синус
cos(x) - Косинус
tg(x) - Тангенс
ctg(x) - Котангенс
acos(x) - АркКосинус
asin(x) - АркСинус
atan(x) - АркТангенс
ln(x) - Н.Логарифм
log(x,n) - Логарифм
 x^2 - Степень
sqrt(x) - Корень

Рисунок 5.1 – Введенные данные тестового примера

Нажатием левой кнопки мыши по элементу управления Button с текстом «Вычислить» получен результат при введенных тестовых данных. Результат предоставлен на рисунке 5.2.

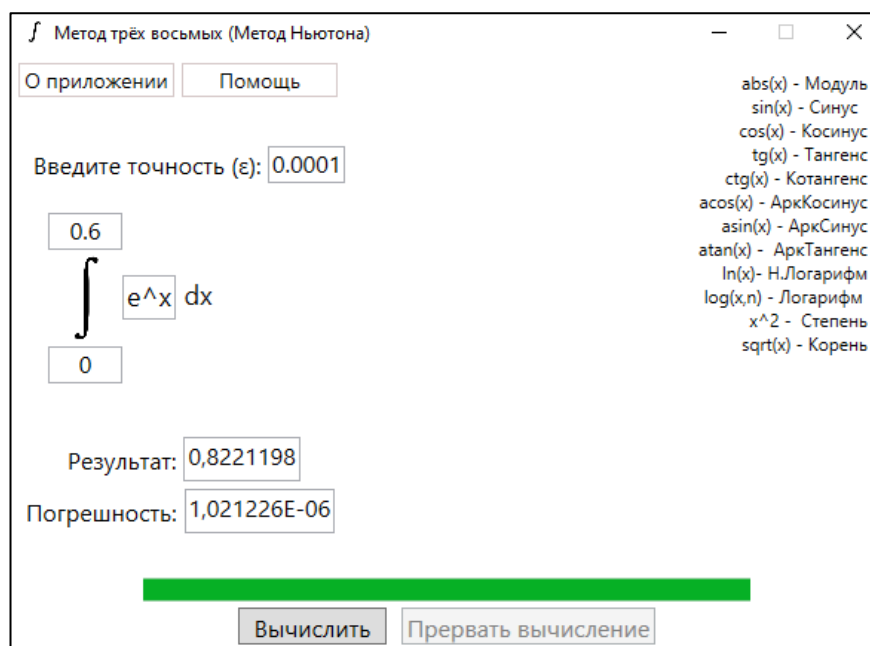


Рисунок 5.2 – Полученные значения тестового примера

Необходимо сравнить данные результаты с расчетами в пункте 5.1. Сравнение предоставлено в таблице 5.4.

Таблица 5.4 – Сравнение результатов значений

Наименование значения	Решение аналитическим методом	Решение «вручную» при $n = 6$	Решение, полученное приложением
Значение интеграла	0,82211880039	0,82211982559	0,8221198
Погрешность	-	0,00000101746	1,021226E-06 = 0,00000102122

Исходя из данных таблицы видно, что в решении полученным через приложение, количество знаков после запятой ограничено. Это связано с используемым типом данных float (Точность – 6-9 цифр). Также можно заметить, что значение погрешности отличается в сравнении с тестовым примером. Небольшое различие связано с ошибкой разрядности сетки ЭВМ. В моменте вычисления погрешности правилом Рунге идет работа с очень маленькими по модулю числами и бесконечными периодическими дробями.

Так же во время подсчета значения интеграла идет сложения различных значений (Больших и маленьких), что вызывает появление погрешности.

5.3 Тестирование программного обеспечения на ошибки

Необходимо проверить программу на обработку всех ошибок, которые может допустить пользователь в процессе эксплуатации приложения.

Для начала необходимо задать значение верхнего или нижнего предела интегрирования равного буквенному значению. Пример ошибки предоставлен на рисунках 5.3 и 5.4.

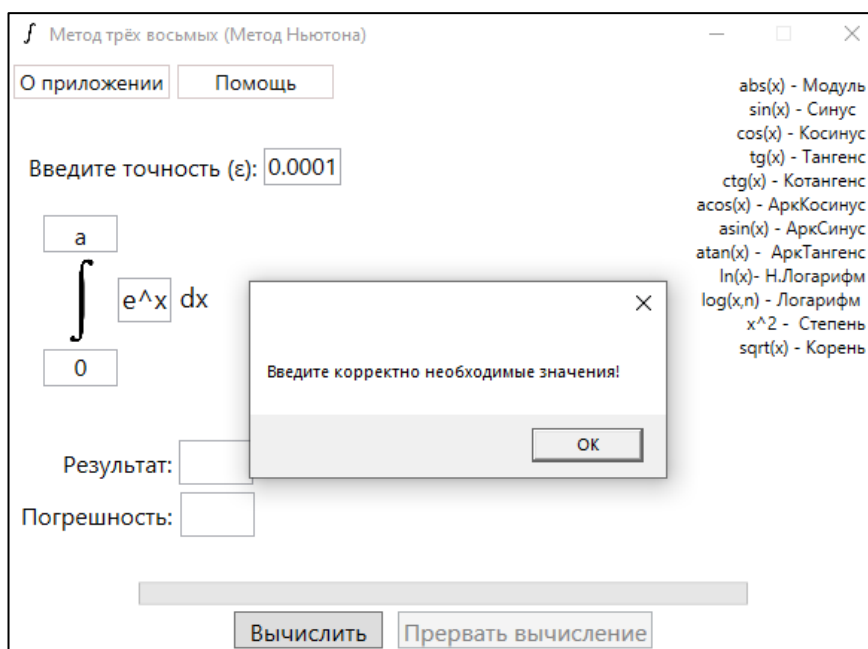


Рисунок 5.3 – Символы в верхнем пределе интегрирования

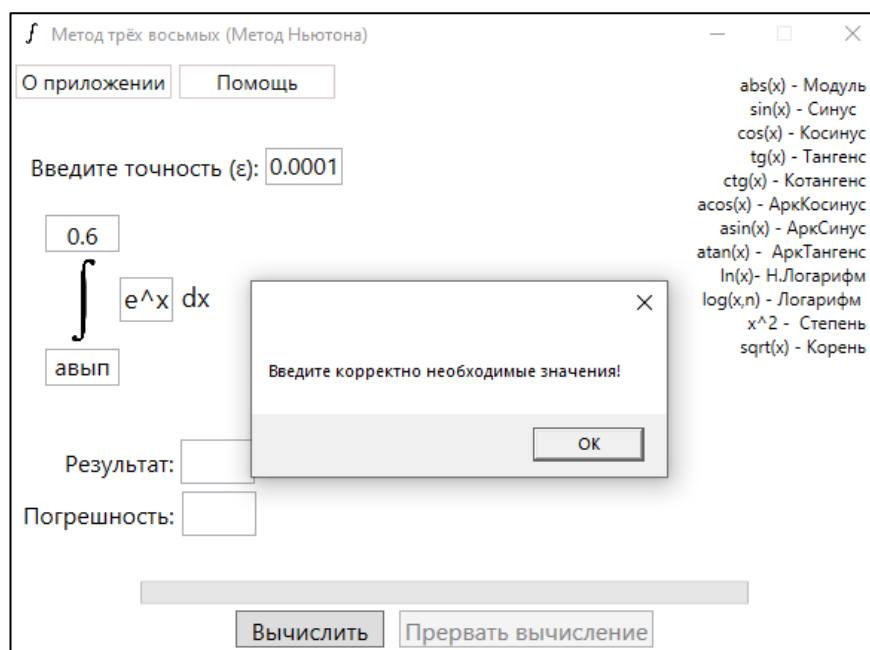


Рисунок 5.4 – Символы в нижнем пределе интегрирования

Следующим шагом ввести вместо точности неверное значение (буквы или число 5). Пример ошибки предоставлен на рисунках 5.5 и 5.6.

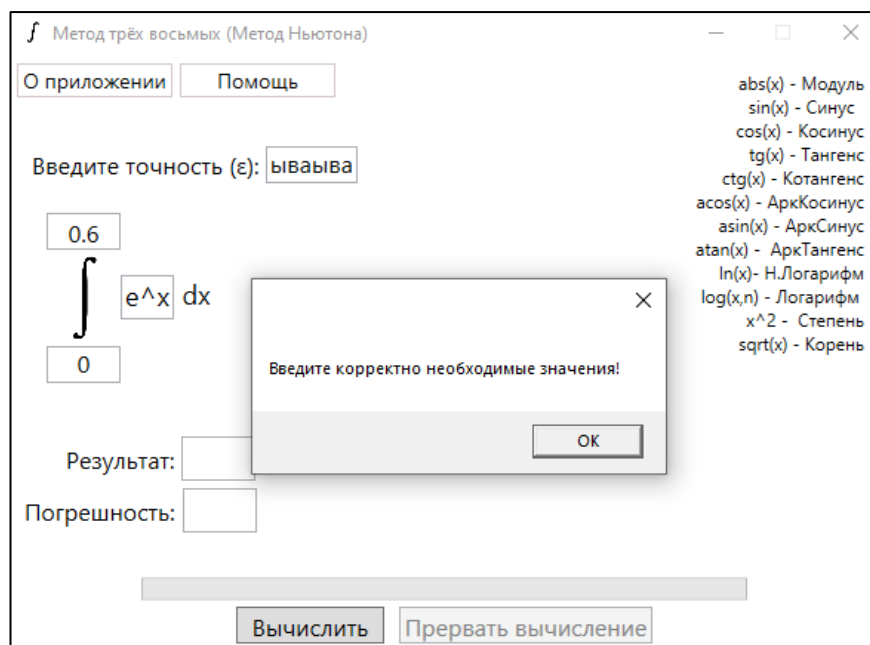


Рисунок 5.5 – Символы в значении точности

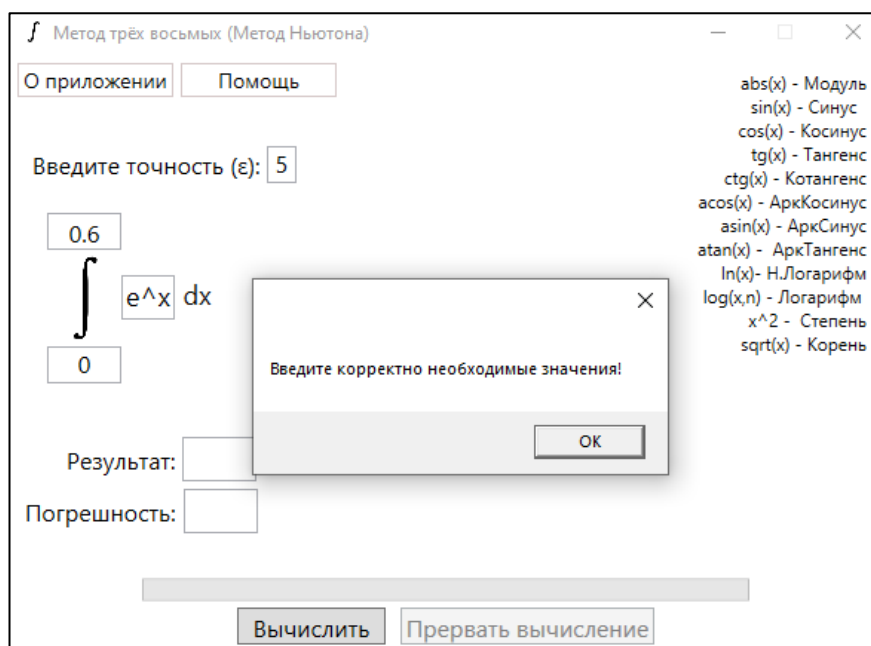


Рисунок 5.6 – Неверное значение точности

Последним тестом необходимо попробовать ввести некорректную подинтегральную функцию. Пример ошибки предоставлен на рисунке 5.7.

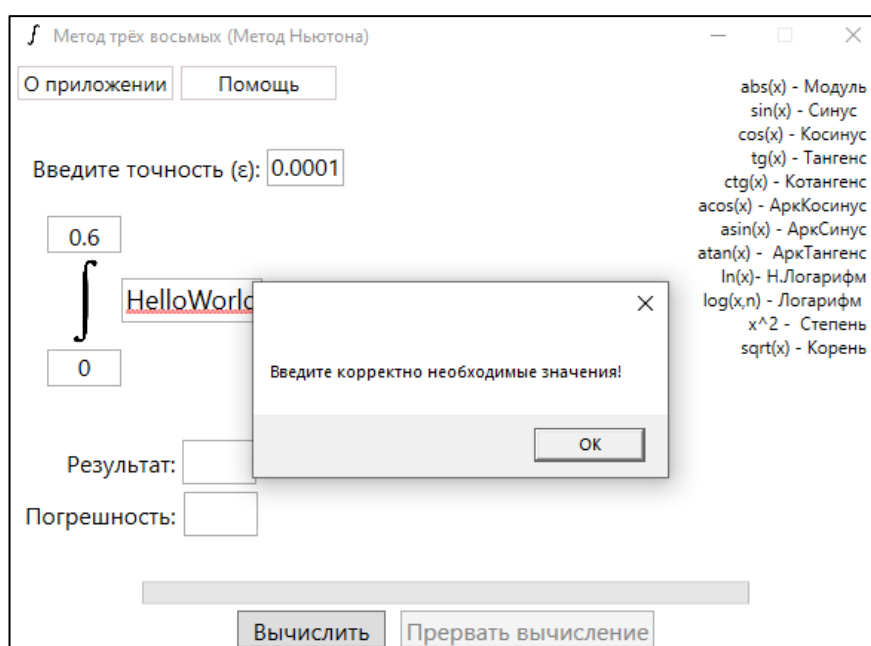


Рисунок 5.7 – Неверная подинтегральная функция

ЗАКЛЮЧЕНИЕ

В ходе выполнения учебной практики:

- изучен численный метод вычисления приближённого значения определенного интеграла методом трёх восьмых (Формула Ньютона);
- рассмотрены возможные источники погрешностей при выполнении расчетов на компьютере;
- разработано графическое приложение определения значения определенного интеграла при заданных пределах и подынтегральной функции;
- проведено тестирование разработанного программного обеспечения и выполнен анализ полученных результатов.

Тестирование разработанного графического приложения продемонстрировало, что в приложении имеется потеря точности интеграла и погрешности связанное с ошибкой разрядности сетки ЭВМ и округлением значений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Сайт СЭОР СГУПС – Система электронных образовательных ресурсов сибирского государственного университета путей сообщения (СГУПС) Учебно-методический комплекс дисциплины «Учебная практика». / Э.А. Усова. – Новосибирск : [сайт]. – URL: <http://moodle3.stu.ru/course/view.php?id=4181#section-5> (дата обращения: 29.06.2021). – Режим доступа: для авториз. пользователей. – Текст. Изображение : электронные.

2 Арушанян, И. О. Применение метода квадратур для численного решения интегральных уравнений второго рода : учебное пособие / И. О. Арушанян — Москва : МГУ, 2018. — 61 с. — Текст : электронный // МГУ : Научно-образовательный интернет ресурс НИВЦ МГУ — URL: http://num-anal.srcc.msu.ru/prac_pos/poslist/posobie%204%20iarush%202018.pdf (дата обращения: 05.07.2021). — Режим доступа: свободный.

3 Шахов, М. А. Численные методы / М. А. Шахов, Е. И. Деза. — Отдельное издание — Москва : Либроком, 2017. — 248 с. — ISBN 978-5-397-05721-9. — Текст : печатный.

ПРИЛОЖЕНИЕ А

Код программы и разметки XAML

А.1 Разметка XAML файла MainWindow.xaml

```
<Window
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    x:Class="calculateIntegral.MainWindow"
    mc:Ignorable="d"
    Title="Метод трёх восьмых (Метод Ньютона)" Height="441" Width="607"
    ResizeMode="CanMinimize"
    WindowStartupLocation="CenterScreen" Icon="Resources/math.ico">
    <Grid Margin="0,0,4,1">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="469*" />
            <ColumnDefinition Width="128*" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="269*" />
            <RowDefinition Height="150*" />
        </Grid.RowDefinitions>
        <StackPanel Grid.Row="1" Grid.ColumnSpan="2">
            <DockPanel>
                <StackPanel>
                    <WrapPanel>
                        <TextBlock HorizontalAlignment="Left" TextWrapping="Wrap"
                            Text="Результат:"
                            VerticalAlignment="Center" Margin="38,0,0,0"
                            FontSize="16" />
                        <TextBox x:Name="ResultTextBox" IsReadOnly="True" MinWidth="50"
                            MinHeight="30" FontSize="16"
                            Margin="5,0,0,0" IsTabStop="False" />
                    </WrapPanel>
                    <WrapPanel Margin="0,5,0,0">
                        <TextBlock HorizontalAlignment="Left" TextWrapping="Wrap"
                            Text="Погрешность:"
                            VerticalAlignment="Center" Margin="10,0,0,0"
                            FontSize="16" />
                        <TextBox x:Name="ErrorResultBox" IsReadOnly="True" MinWidth="50"
                            MinHeight="30" FontSize="16"
                            Margin="5,0,0,0" IsTabStop="False" />
                    </WrapPanel>
                </StackPanel>
            </DockPanel>
            <ProgressBar x:Name="Pg1" HorizontalAlignment="Center" Height="15"
                VerticalAlignment="Center"
                Width="409" Margin="0,30,0,0" LargeChange="100" />
            <WrapPanel VerticalAlignment="Center" HorizontalAlignment="Center">
                <Button x:Name="Evaluate" Content="Вычислить"
                    HorizontalAlignment="Center" VerticalAlignment="Bottom"
                    Click="Evaluate_Click" FontSize="16" Height="25" Width="100"
                    Margin="0,5,0,0" TabIndex="4" />
                <Button x:Name="Abort" Content="Прервать вычисление"
                    HorizontalAlignment="Center"
                    IsEnabled="False" />
            </WrapPanel>
        </StackPanel>
    </Grid>
</Window>
```

```

VerticalAlignment="Bottom" FontSize="16" Height="25" Width="171"
Margin="10,5,0,0"
Click="Abort_Click" TabIndex="5" />
</WrapPanel>
</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="0" HorizontalAlignment="Left"
Margin="5,5,0,0">
<WrapPanel>
<Button x:Name="InfoOpen" Content="0 приложения" FontSize="14"
Width="105" HorizontalAlignment="Left"
BorderBrush="#FFD6C9C9" Background="White" Click="infoOpen_Click"
TabIndex="6" />
<Button x:Name="InstructionsOpen" Content="Помощь" FontSize="14"
Width="105" HorizontalAlignment="Left"
BorderBrush="#FFD6C9C9" Background="White" Margin="5,0,0,0"
Click="InstructionsOpen_Click" TabIndex="7" />
</WrapPanel>
<WrapPanel>
<TextBlock Text="Введите точность (ε): " VerticalAlignment="Center"
FontSize="16"
HorizontalAlignment="Left" Margin="10,35,0,2" />
<TextBox x:Name="EpsTextBox" Height="25" MinWidth="20"
VerticalAlignment="Bottom" FontSize="16" Text="0.0001"
TextAlignment="Center" TabIndex="0"/>
</WrapPanel>
<WrapPanel HorizontalAlignment="Left" Margin="0" VerticalAlignment="Center">
<StackPanel HorizontalAlignment="Left" VerticalAlignment="Center"
Margin="20,20,0,20">
<TextBox x:Name="BTextBox" HorizontalAlignment="Left" Height="25"
TextWrapping="Wrap"
VerticalAlignment="Top" Width="50" TextAlignment="Center"
FontSize="16" TabIndex="1" Text="0.6" />
<Image HorizontalAlignment="Center" Height="65"
VerticalAlignment="Center" Width="25"
Source="Resources/intergral.png" Stretch="UniformToFill" />
<TextBox x:Name="ATextBox" HorizontalAlignment="Left" Height="25"
TextWrapping="Wrap"
VerticalAlignment="Top" Width="50" FontSize="16"
TextAlignment="Center" TabIndex="3" Text="0" />
</StackPanel>
<TextBox x:Name="FormulaTextBox" TextWrapping="Wrap"
SpellCheck.IsEnabled="True" MinWidth="20"
RenderTransformOrigin="0.767,0.653" FontSize="18"
HorizontalAlignment="Center"
VerticalAlignment="Center" Margin="0" MinHeight="30"
MaxWidth="360" TabIndex="2" Text="e^x" />
<TextBlock HorizontalAlignment="Center" Height="30" TextWrapping="Wrap"
VerticalAlignment="Center"
MinWidth="20" RenderTransformOrigin="0.767,0.653"
Margin="5,0,0,0" Text="dx"
TextAlignment="Center" FontSize="18" />
</WrapPanel>
</StackPanel>
<TextBlock Grid.Row="0" Grid.Column="1" HorizontalAlignment="Right"
VerticalAlignment="Top" TextWrapping="Wrap"
Height="205"
Width="117" TextAlignment="Right" Margin="0,10,10,0">
<Run Text="abs(x) - Модуль" /><LineBreak /><Run Text="sin(x) - Синус" /><Run
Text=" " /><LineBreak />
<Run Text="cos(x) - Косинус" /><LineBreak /><Run Text="tg(x) - Тангенс"
/><LineBreak />
<Run Text="ctg(x) - Котангенс" /><LineBreak /><Run Text="acos(x) - " /><Run
Text="АркКосинус" />

```

```

        <LineBreak /><Run Text="asin(x) - " /><Run Text="Арксинус" /><LineBreak
/><Run Text="atan(x) - " />
        <Run Text="Арктангенс" /><LineBreak /><Run Text="ln(x) - " /><Run Text="Н. "
/><Run Text="Логарифм" />
        <LineBreak /><Run Text="log(x,n) - " /><Run Text="Логарифм " /><LineBreak
/><Run Text="x^2 - " />
        <Run Text="Степень" /><LineBreak /><Run Text="sqrt(x) - " /><Run
Text="Корень" /><LineBreak /><Run />
    </TextBlock>

</Grid>
</Window>

```

A.2 Разметка XAML файла InfoWindow.xaml

```

<Window x:Class="calculateIntegral.InfoWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    Title="0 приложении" Height="480" Width="520" ResizeMode="NoResize"
    WindowStartupLocation="CenterScreen"
    Icon="Resources/math.ico">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="331*" />
            <ColumnDefinition Width="186*" />
        </Grid.ColumnDefinitions>
        <StackPanel Grid.ColumnSpan="2">
            <TextBlock TextWrapping="Wrap" Margin="5" FontSize="16"
                TextAlignment="Justify">
                <Run
                    Text="Данная программа выполняет численное интегрирование
определенного интеграла с помощью метода трех восьмых (Формула Ньютона)." />
                <LineBreak /><Run Text="Разработал студент группы БИСТ-112 Коровченко
А.Ю." /><LineBreak /><Run />
                <LineBreak /><Run Text="Используемые формулы:" />
            </TextBlock>
            <Image Source="Resources/formula.png" />
            <TextBlock TextWrapping="Wrap" FontSize="16" TextAlignment="Justify"
                Margin="5,0"><Run Text="Для вычисления погрешности используется 'Прав' /><Run
                Text="ило" /><Run Text=" Рунге'. Формула:" /></TextBlock>
            <Image Source="Resources/errorRate.png" Margin="0,0,164,0"
                Stretch="UniformToFill" />
            <TextBlock TextWrapping="Wrap" FontSize="16" TextAlignment="Justify"
                Margin="5,0">
                <Run /><LineBreak />
                <Run
                    Text="Для распознавания формулы и вычисления значения при
определенном значении x используется библиотека AngouriMath" />
            </TextBlock>
            <TextBlock Text="@Коровченко А.Ю. 2021" HorizontalAlignment="Right"
                VerticalAlignment="Bottom"
                Margin="0,5,5,5" />
        </StackPanel>
    </Grid>
</Window>

```


A.3 Разметка XAML файла InstructionsWindow.xaml

```
<Window x:Class="calculateIntegral.InstrinctionsWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        mc:Ignorable="d"
        Title="Помощь" Height="394" Width="530" ResizeMode="NoResize"
        WindowStartupLocation="CenterScreen"
        Icon="Resources/math.ico" ShowInTaskbar="False">
    <Grid>
        <StackPanel>
            <TextBlock TextWrapping="Wrap" Margin="5" FontSize="16"
                TextAlignment="Justify"><Run Text="Для начала необходимо задать формулу в соответствии с
                правилами предоставленными в правой части приложения. Далее следует указать границы
                интегрирования и точность ( $\epsilon$ )."/><LineBreak/><LineBreak/><Run Text="После ввода значения
                результат записывается в поле "Результат" в нижней части приложения."/><Run
                Text=" "/><LineBreak/><Run/><LineBreak/><Run/><LineBreak/><Run Text="Для удобства, в
                случае поиска значения функции при очень "/><Run Text="долгих вычислениях"/><Run Text=","
                имеет поле, отмечающее этап выполнения программы. Также для такого случая имеется
                возможность прервать вычисление специальной
                кнопкой."/><LineBreak/><Run/><LineBreak/><LineBreak/><Run Text="Желаем
                удачи!"/></TextBlock>
            <TextBlock Text="@Коровченко А.Ю. 2021" HorizontalAlignment="Right"
                VerticalAlignment="Bottom"
                Margin="0,5,5,5" />
        </StackPanel>
    </Grid>
</Window>
```

A.4 Код файла MainWindow.xaml.cs

```
using System;
using System.Globalization;
using System.Text.RegularExpressions;
using System.Threading;
using System.Windows;
using AngouriMath;

namespace calculateIntegral
{
    /// <summary>
    ///     Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow
    {
        private Thread _calc;
        private Thread _errorRateThread;

        public MainWindow()
        {
            InitializeComponent();
        }

        private void Evaluate_Click(object sender, RoutedEventArgs e)
        {

```

```

if (_calc is null)
{
    if (_errorRateThread is not null)
        if (_errorRateThread.IsAlive)
        {
            _errorRateThread.Abort();
            _errorRateThread = null;
        }

    string f;
    float b, a;
    float eps;
    try
    {
        f = FormulaTextBox.Text;
        b = float.Parse(BTextBox.Text.Replace('.', ','));
        a = float.Parse(ATextBox.Text.Replace('.', ','));
        if (Regex.IsMatch(EpsTextBox.Text.Replace('.', ','),
@"0\,[0]*[1]{1}$"))
        {
            eps = float.Parse(EpsTextBox.Text.Replace('.', ','));
        }
        else
        {
            MessageBox.Show("Введите корректно точность вычисления!");
            return;
        }
    }
    catch (Exception)
    {
        MessageBox.Show("Введите корректно необходимые значения!");
        return;
    }

    _calc = new Thread(() => CalcThread(f, b, a, eps));
    Pg1.Value = 0;
    _calc.Start();
    Abort.IsEnabled = true;
    ResultTextBox.Text = "";
    ErrorResultBox.Text = "";
}
else
{
    MessageBox.Show("Вычисление выполняется");
}
}

private void CalcThread(string f, float b, float a, float eps = 0.00001f)
{
    float res = 1;
    float finally = 0;
    int n = 3;
    while (eps < res)
    {
        float h = (b - a) / n;
        float progress = 100f / n;
        float? sum = 0;
        for (int i = 0; i <= n; i++)
        {
            float xi = a + h * i;
            if (i == 0 || i == n) sum += CalcFunc(f, xi);
            else if (i % 3 == 0) sum += 2 * CalcFunc(f, xi);
            else sum += 3f * CalcFunc(f, xi);
            Pg1.Dispatcher.Invoke(() => Pg1.Value += progress);
        }
    }
}

```

```

        if (sum == null)
        {
            _calc = null;
            MessageBox.Show("Корректно введите функцию");
            return;
        }
    }

    float fin1 = (float)sum * ((3f * h) / 8f);
    finally = fin1;
    float h2 = (b - a) / (n / 2f);
    float? sum2 = 0;
    for (int i = 0; i <= n / 2f; i++)
    {
        float xi2 = a + i * h2;
        if (i == 0 || i == n / 2) sum2 += CalcFunc(f, xi2);
        else if (i % 3 == 0) sum2 += 2 * CalcFunc(f, xi2);
        else sum2 += 3 * CalcFunc(f, xi2);
    }

    float fin2 = (float)(sum2 * (3f * h2 / 8f));
    res = Math.Abs(fin1 - fin2) * (1f / 15f);
    n += 3;
}
Dispatcher.Invoke(() =>
{
    ResultTextBox.Text = finally.ToString(CultureInfo.CurrentCulture);
    ErrorResultBox.Text = res.ToString(CultureInfo.CurrentCulture);
});
Dispatcher.Invoke(() => Abort.IsEnabled = false);
_calc = null;
}

private float? CalcFunc(string f, float x)
{
    float res;
    try
    {
        Entity expr = f;
        res = (float) expr.Substitute("x", x).EvalNumerical();
    }
    catch (Exception)
    {
        Dispatcher.Invoke(() => Abort.IsEnabled = false);
        return null;
    }

    return res;
}

private void Abort_Click(object sender, RoutedEventArgs e)
{
    _calc.Abort();
    _calc = null;
    Pg1.Value = 0;
    Abort.IsEnabled = false;
}

private void infoOpen_Click(object sender, RoutedEventArgs e)
{
    InfoWindow info = new InfoWindow();
    info.Show();
}

```

```

        private void InstructionsOpen_Click(object sender, RoutedEventArgs e)
        {
            InstructionsWindow ins = new InstructionsWindow();
            ins.Show();
        }
    }
}

```

A.5 Код файла InfoWindow.xaml.cs

```

namespace calculateIntegral
{
    /// <summary>
    /// Логика взаимодействия для InfoWindow.xaml
    /// </summary>
    public partial class InfoWindow
    {
        public InfoWindow()
        {
            InitializeComponent();
        }
    }
}

```

A.6 Код файла InstructionsWindow.xaml.cs

```

namespace calculateIntegral
{
    /// <summary>
    /// Логика взаимодействия для InfoWindow.xaml
    /// </summary>
    public partial class InstructionsWindow
    {
        public InstructionsWindow()
        {
            InitializeComponent();
        }
    }
}

```