Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

Кафедра информационных систем и программирования

**ТЕСТИРОВАНИЕ И ОТЛАДКА  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Отчет по лабораторной работе №10

«Программная отладка и трассировка программ»

Выполнил:

студент 3 курса

группы 19-КБ-ПР1

Кравцов Олег Юрьевич

Проверил:

ст. преп. каф. ИСП

А. Г. Волик

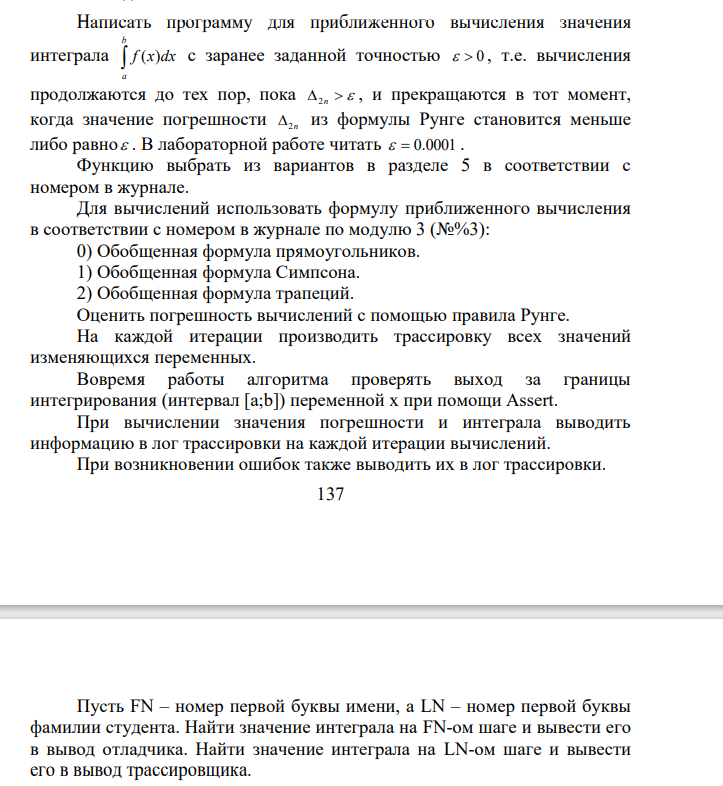
Краснодар

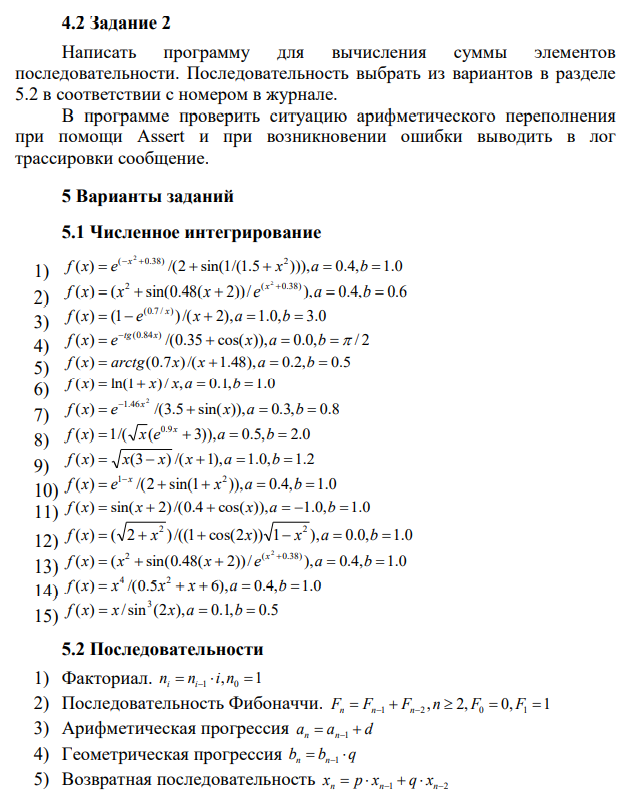
2021

**1 Цель работы**

Изучить средства и возможности программной отладки и трассировки программ.

**2 Задание**





**3 Вариант задания**

Вариант №12

**4 Текст программы**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

namespace Lr\_10\_\_ТОПО\_

{

class Program

{

static void Main(string[] args)

{

Trace.Listeners.Add(

new TextWriterTraceListener(Console.Out));

Trace.Indent();

Debug.Print("Начало отладки");

double e = 0.0001;

double a = 0.0;

double b = 1;

int n = 1;

Func<double, double> func = x => (Math.Sqrt(2 + Math.Pow(x, 2))) / ((1 + Math.Cos(2) \* x) \* Math.Sqrt(1 - Math.Pow(x, 2)));

try

{

Console.WriteLine($"Значение Фибоначчи: {Fibonacci(30)}");

}

catch (OverflowException ex)

{

Trace.Fail(ex.Message);

}

try

{

double result = LeftRectangle(func, a, b, n);

double result2;

double delta = 0;

n = 2;

do

{

result2 = LeftRectangle(func, a, b, n);

delta = Math.Abs(result2 - result) \* (1.0 / 3.0);

Trace.WriteLine(String.Format($"1-е зн. интеграла: {result}\n"));

Trace.WriteLine(String.Format($"2-е зн. интеграла: {result2}\n"));

Trace.WriteLine(String.Format($"Погрешность составила: {delta}\n\n"));

result = result2;

n \*= 2;

}

while (delta > e);

Console.WriteLine($"Результат равен: {result2}");

Debug.Print("Конец отладки");

}

catch (OverflowException ex)

{

Trace.Fail(ex.Message);

}

}

public static int Fibonacci(int index)

{

if (index <= 1)

return index;

else

return checked((Fibonacci(index - 1) + Fibonacci(index - 2)));

}

static double LeftRectangle(Func<double, double> F, double A, double B, int N)

{

var h = (B - A) / N;//h - шаг разбиения, N - кол-во отрезков, А - верхний, B - нижний предел

var sum = 0d;//сумма отрезков

for (var i = 0; i <= N - 1; i++)//высчитываем сумму отрезков интеграла

{

var x = A + i \* h;

sum += F(x);

Debug.WriteLineIf(x < A, String.Format($"x = {x} вышел за границу A"));

Debug.WriteLineIf(x > B, String.Format($"x = {x} вышел за границу B"));

}

var result = h \* sum;//значение функции

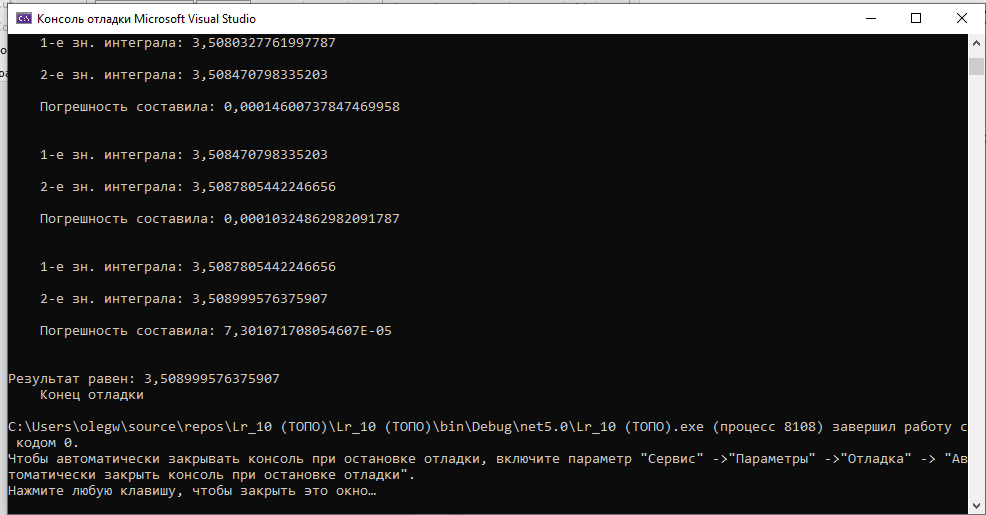
return result;

}

}

}

**5 Результаты тестирования программы**



**6 Выводы**

В результате выполнения лабораторной работы был изучен класс Debug, Trace, а также более детально была изучена процедура трассировки программы.