Оглавление

[Часть №1 Описание задания. 3](#_Toc27570707)

[Часть №2 Этапы проектирования по практической работе №2 3](#_Toc27570708)

[Раздел №1 Наименование работы 3](#_Toc27570709)

[Раздел №2 Спецификация проблемы №1: Поиск решения одномерной задачи оптимизации метод Равномерного поиска. 3](#_Toc27570710)

[Раздел №3 Спецификация (описание) метод равномерного поиска 3](#_Toc27570711)

[Раздел №4 Стадия проектирования системы для поиска решения одномерной задачи оптимизации метод Равномерного поиска 5](#_Toc27570712)

[Раздел №5 Проектирование ПО: Документирование этапов проектирования интерфейсной формы системы, реализующей метод равномерного поиска – для поиска решения одномерной задачи оптимизации. 6](#_Toc27570713)

[Раздел №6 Стадия конструирования программного обеспечения для поиска решения одномерной задачи оптимизации. 8](#_Toc27570714)

[Раздел №7: Тесты для проверки программного обеспечения, реализующего метод ESM 12](#_Toc27570715)

# Часть №1 Задание

***Even Search*** method of optimization – ***Метод равномерного поиска***»

**Дано:**

* Задается аналитическое выражение для произвольной целевой функции f(x);
* Спецификация метода оптимизации (***Even Search Method***) для нахождения оптимального значения решающей переменной;
* Структура интерфейсной формы, реализующая итерационный метод — ***Even Search Method***.

**Что требуется:**

* Разработать проект ПО для поиска **решения задач оптимизации** для произвольной заданной допустимой погрешности;
* Написать ***код*** ПО для поиска **решения задач оптимизации** для произвольной заданной допустимой погрешности;
* Доказать, что найдено **оптимальное**решение с погрешностью решения не более заданной допустимой погрешности.

# Часть №2 Этапы проектирования по практической работе №2

## Раздел №1 Наименование работы

Наименование работы – поиск решения одномерной задачи оптимизации методом Равномерного поиска (ESM)

## Раздел №2 Спецификация проблемы №1: Поиск решения одномерной задачи оптимизации метод Равномерного поиска.

Найти решение задачи оптимизации для произвольной заданной допустимой погрешности. Метод имеет постоянный размер шага h, который не зависит от номера итерации или иначе этот метод можно назвать алгоритмом с фиксированным шагом поиска. Таким образом, метод равномерного поиска генерирует стационарный одношаговый итерационный процесс.

## Раздел №3 Спецификация (описание) метод равномерного поиска

Предположим, что задана задача оптимизации следующего вида:

**Пример:** Задача типа **(3)** это задача **максимизации одномерной** целевой функции **без ограничения** на область поиска. Задана допустимая погрешность **δ** и начальная аппроксимация ***х0*** и выбираемый (настраиваемый) параметр **h.**

*Def:* Алгоритм который реализует соотношение вида **(2)** в виде следующего алгоритма:

***If f(хk )< f(хk-1 )***

***then хk+1 = хk***

***else хk+1 = хk +h; для всех k=0,1...***

Этот алгоритм называется алгоритмом равномерного поиска. Название этого алгоритма связано с тем, что на каждом шаге поиска алгоритм не имеет размер шага поиска **h,** т.е. **h=const.** Алгоритм равномерного поиска работает если ***х0*** задано слева от оптимальной точки, т.е. этот метод будет работать при

***х0 < хx* (5)**

и шаг поиска **h** должен быть меньше или равен допустимой погрешности:

**h ≤ δ (6)**

Метод равномерного поиска имеет постоянный размер шага **h,** который не зависит от номера итерации или иначе этот метод можно назвать алгоритмом с фиксированным шагом поиска.

Таким образом, метод равномерного поиска генерирует стационарный одношаговый итерационный процесс.

## Раздел №4 Стадия проектирования системы для поиска решения одномерной задачи оптимизации метод Равномерного поиска

Этап №1: Разработка блок-схемы ESM





## Раздел №5 Проектирование ПО: Документирование этапов проектирования интерфейсной формы системы, реализующей метод равномерного поиска – для поиска решения одномерной задачи оптимизации.

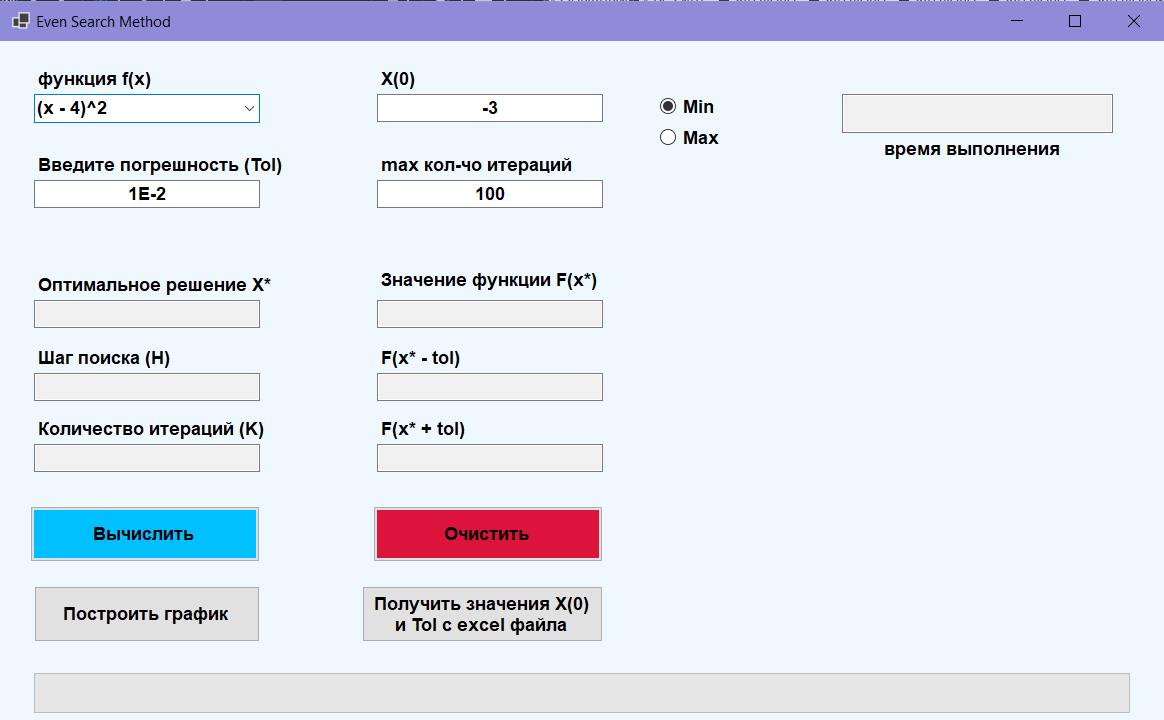


Рис 1.

Этап №2: Документирование процесса задания свойств элементов интерфейсной формы системы, выбранной для внедрения и реализующей метод равномерного поиска.

Таблица 1: Настройки управляющих свойств программной системы, реализующей метод равномерного поиска.

|  |  |  |  |
| --- | --- | --- | --- |
| Кол-во элементов  управления | Элемент Управления | Свойства | Настройки |
| 1 | radioButton1 | Appearance(text)  Checked(bool)  Design(Name) | Min  True  radioButton1 |
| 2 | radioButton2 | Appearance(text)  Checked(bool)  Design(Name) | Max  False  radioButton2 |
| 3 | Label1 | Appearance(text)  Design(Name) | функция F(x)  Label1 |
| 4 | comboBox1 | Appearance(text)  The rest (Item)  Item(Collection) | (x - 4)^2  Collection  (x - 4)^2  (4 - x^2)^2  x^3 + x^2 - x + 1  1600 \* (1 - exp(-x / 5)) - 160 \* x  x^2 - 4 \* sin(x)  4 \* exp(x)^3 - 2 \* x - 6x^2-exp(x)  -50,01\*x^3+x^2+5\*x-1 |
| 5 | Label2 | Appearance(text)  Design(Name) | X(0)  Label2 |
| 6 | textBox1 | Appearance(text)  Design(Name) | -3  textBox1 |
| 7 | Label3 | Appearance(text)  Design(Name) | Введите погрешность (Tol)  Label3 |
| 8 | textBox2 | Design(Name) | 1E-2  textBox2 |
| 9 | Label4 | Appearance(text)  Design(Name) | Max кол-чо итераций (Max)  Label4 |
| 10 | textBox3 | Design(Name) | 100  textBox3 |
| 11 | Button1 | Appearance(text)  Design(Name) | Вычислить  Button1 |
| 12 | Button2 | Appearance(text)  Design(Name) | Очистить  Button2 |
| 13 | Label5 | Appearance(text)  Design(Name) | Оптимальное решение X1  Label5 |
| 14 | textBox4 | Design(Name)  Behavior(ReadOnly) | textBox4  True |
| 15 | Label6 | Appearance(text)  Design(Name) | Значение функции YF1  Label6 |
| 16 | textBox5 | Design(Name)  Behavior(ReadOnly) | textBox5  True |
| 17 | Label7 | Appearance(text)  Design(Name) | Количество итераций (K)  Label7 |
| 18 | textBox6 | Design(Name)  Behavior(ReadOnly) | textBox6  True |
| 19 | Label8 | Appearance(text)  Design(Name) | Шаг поиска (H)  Label8 |
| 20 | Label12 | Appearance(text)  Behavior(ReadOnly) | Label12True  False |
| 21 | Label9 | Appearance(text)  Design(Name) | Время выпонения  Label9 |
| 22 | textBox8 | Design(Name)  Behavior(ReadOnly) | textBox8  True |
| 23 | Button3 | Appearance(text)  Design(Name) | Построить график  Button3 |
| 24 | progressBar1 | Design(Name)  Behavior(Visible) | progressBar1  False |
| 25 | ToolTip1 | Design(Name)  Misc(ShowAlways)  (IsBaloon) | ToolTip1  True  True |
| 26 | Label10 | Appearance(text)  Design(Name) | F(x\* - tol)  Label9 |
| 27 | textBox10 | Design(Name)  Behavior(ReadOnly) | textBox10  True |
| 28 | Label11 | Design(Name)  Behavior(ReadOnly) | F(x\* + tol)  Labell11 |
| 29 | textBox11 | Design(Name)  Behavior(ReadOnly) | textBox11  True |
| 30 | Button4 | Appearance(text)  Design(Name) | Получить значения X(0) и Tol с excel файла  Button4 |

## Раздел №6 Стадия конструирования программного обеспечения для поиска решения одномерной задачи оптимизации.

Этап №3: Код программы на C# , ассоциированный с интерфейсной формой “Form1.cs”

using parserDecimal.Parser;

using System;

using System.Windows.Forms;

using System.IO;

using Microsoft.Office.Interop.Excel;

using Excel = Microsoft.Office.Interop.Excel;

using System.Diagnostics;

namespace Even\_Search\_Method

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

string func = ""; //функция

private void button1\_Click(object sender, EventArgs e)

{

Computer computer = new Computer();

decimal x0, x1 = 0, tol, H = 0, YF0, YF1 = 0, F1,F2;

int max, k = 0, cond = 0;

func = comboBox1.Text;

func = func.ToLower();

Stopwatch swatch = new Stopwatch();

swatch.Start(); //начало подсчета времени

try

{

x0 = decimal.Parse(textBox1.Text);

YF0 = computer.Compute(func, x0);

tol = Convert.ToDecimal(Convert.ToDouble(textBox2.Text));

max = int.Parse(textBox3.Text);

}

catch

{

MessageBox.Show("Проверьте входные данные.",

"Ошибка валидации", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (max <= 0)

{

MessageBox.Show("Значение итерации должно быть больше единицы!",

"Ошибка валидации", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else if (tol <= 0)

{

MessageBox.Show("Погрешность должно быть больше нуля",

"Ошибка валидации", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

if (radioButton1.Checked)

{

H = tol;

YF0 = aziretParser.ParserDecimal.Compute(func, x0);

while (k < max)

{

k = k + 1;

progressBar1.Visible = true;

progressBar1.Maximum = k + 1;

progressBar1.Value = k;

x1 = x0 + H;

YF1 = aziretParser.ParserDecimal.Compute(func, x1);

if (YF1 >= YF0)

{

if (k == 1)

{

cond = 1;

}

x1 = x0;

YF1 = YF0;

break;

}

else

{

x0 = x1;

YF0 = YF1;

x1 = x0 + H;

YF1 = aziretParser.ParserDecimal.Compute(func, x1);

}

}

swatch.Stop();

progressBar1.Value = k;

progressBar1.Visible = true;

progressBar1.Value = 0;

textBox8.Text = (swatch.Elapsed).ToString();

textBox4.Text = x1.ToString();

textBox5.Text = YF1.ToString("0E0");

textBox6.Text = k.ToString();

textBox7.Text = H.ToString("0E0");

F1 = computer.Compute(func, x1 - tol);

textBox10.Text = F1.ToString("0E0");

F2 = computer.Compute(func, x1 + tol);

textBox11.Text = F2.ToString("0E0");

if (YF1 <= F1 && YF1 <= F2)

{

label12.ForeColor = System.Drawing.Color.Green;

label12.Visible = true;

label12.Text = "\r\n" +

"\r\n" +

"The result x\* is the minimizer of \r\nthis function because \r\n" +

"F(x\*) <= F(x\*–Tolerance) \r\n" +

"And \r\n" +

"F(x\*) <= F(x\*+Tolerance) \r\n";

}

else

{

label12.ForeColor = System.Drawing.Color.DarkRed;

label12.Visible = true;

label12.Text = "\r\n" +

"\r\n" +

"The result x\* is not the minimizer of\r\nthis function because \r\n" +

"F(x\*) <= F(x\*–Tolerance) \r\n" +

"And \r\n" +

"F(x\*) >= F(x\*+Tolerance) \r\n";

}

if (cond == 1)

{

label12.ForeColor = System.Drawing.Color.DarkRed;

DialogResult result = MessageBox.Show("Метод не выполнил ни одну итерацию, \r\n" +

"поскольку начальное значение уже является \r\n" +

"оптимальной или находится справо от оптимальной. \r\n" +

"\r\n" +

"Хотите проверить график функции?", "Внимание",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result ==DialogResult.Yes)

{

button3\_Click(sender, e);

}

}

else if (k == max)

{

DialogResult result = MessageBox.Show("Решение не может быть найдено с данной погрешностью \r\n" +

"из-за лимита количества итераций. \r\n" +

"\r\n" +

"Хотите добавит допольнительную итерацию?", "Внимание",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

max = max + max;

textBox3.Text = max.ToString();

button1\_Click(sender, e);

}

}

}

else if (radioButton2.Checked)

{

H = tol;

YF0 = computer.Compute(func, x0);

while (k < max)

{

k = k + 1;

progressBar1.Visible = true;

progressBar1.Maximum = k + 1;

progressBar1.Value = k;

x1 = x0 + H;

YF1 = computer.Compute(func, x1);

if (YF1 <= YF0)

{

if (k == 1)

{

cond = 1;

}

x1 = x0;

YF1 = YF0;

break;

}

else

{

x0 = x1;

YF0 = YF1;

x1 = x0 + H;

YF1 = computer.Compute(func, x1);

}

}

swatch.Stop();

progressBar1.Visible = false;

progressBar1.Value = 0;

textBox8.Text = (swatch.Elapsed).ToString();

textBox4.Text = x1.ToString();

textBox5.Text = YF1.ToString("0E0");

textBox6.Text = k.ToString();

textBox7.Text = H.ToString("0E0");

F1 = computer.Compute(func, x1 - tol);

textBox10.Text = F1.ToString("0E0");

F2 = computer.Compute(func, x1 + tol);

textBox11.Text = F2.ToString("0E0");

if (YF1 >= F1 && YF1 >= F2)

{

label12.ForeColor = System.Drawing.Color.Green;

label12.Visible = true;

label12.Text = "\r\n" +

"\r\n" +

"The result x\* is the maximizer of \r\nthis function because \r\n" +

"F(x\*) >= F(x\*–Tolerance) \r\n" +

"And \r\n" +

"F(x\*) >= F(x\*+Tolerance) \r\n";

}

else

{

label12.ForeColor = System.Drawing.Color.DarkRed;

label12.Visible = true;

label12.Text = "\r\n" +

"\r\n" +

"The result x\* is not the maximizer of \r\nthis function because \r\n" +

"F(x\*) >= F(x\*–Tolerance) \r\n" +

"And \r\n" +

"F(x\*) <= F(x\*+Tolerance) \r\n";

}

if (cond == 1)

{

DialogResult result = MessageBox.Show("Метод не выполнил ни одну итерацию, \r\n" +

"поскольку начальное значение уже является \r\n" +

"оптимальной или находится справо от оптимальной. \r\n" +

"\r\n" +

"Хотите проверить график функции?", "Внимание",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

button3\_Click(sender, e);

}

}

else if (k == max)

{

DialogResult result = MessageBox.Show("Решение не может быть найдено с данной погрешностью \r\n" +

"из-за лимита количества итераций. \r\n" +

"\r\n" +

"Хотите добавит допольнительную итерацию?", "Внимание",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

max = max + max;

textBox3.Text = max.ToString();

button1\_Click(sender, e);

}

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

textBox10.Clear();

textBox11.Clear();

label12.Visible = false;

}

private void button3\_Click(object sender, EventArgs e)

{

string mySheet = Path.Combine(System.Windows.Forms.Application.StartupPath, "Grafic.xlsx");

decimal b, c;

Excel.Application ExcelApp = new Excel.Application();

Microsoft.Office.Interop.Excel.Workbook wb = ExcelApp.Workbooks.Open(mySheet);

Worksheet ws = (Worksheet)wb.ActiveSheet;

ExcelApp.Visible = true;

func = comboBox1.Text;

ws.Cells[2, 2] = func;

func = func.Replace("exp", "!");

func = func.Replace("x", "D4");

func = "=" + func.Replace("!", "exp");

ws.Cells[4, 9] = textBox1.Text;

b = decimal.Parse(textBox1.Text);

c = Math.Abs(b) + 3;

ws.Cells[4, 10] = c;

ws.Range["E4", "E10003"].Value = func;

}

private void button4\_Click(object sender, EventArgs e)

{

string mySheet = Path.Combine(System.Windows.Forms.Application.StartupPath, "Grafic.xlsx");

Excel.Application ExcelApp = new Excel.Application();

Microsoft.Office.Interop.Excel.Workbook wb = ExcelApp.Workbooks.Open(mySheet);

Worksheet sh = (Worksheet)wb.ActiveSheet;

Microsoft.Office.Interop.Excel.Range cell = sh.Cells[4, 9] as Excel.Range;

string value = cell.Value2.ToString();

textBox1.Text = value;

Microsoft.Office.Interop.Excel.Range cell2 = sh.Cells[4, 10] as Excel.Range;

string value2 = cell2.Value2.ToString();

decimal a = decimal.Parse(value);

decimal b = decimal.Parse(value2);

decimal c = decimal.Parse(textBox3.Text);

decimal result = (a + b) / c;

textBox2.Text = result.ToString();

wb.Close();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

textBox10.Clear();

textBox11.Clear();

label12.Visible = false;

}

}

}

Шаг 7.1: Совершаем ряд следующих действий:

Visual Studio -> New Project -> Visual C# -> Windows Forms Application

Шаг 7.2: Создаем папку внутри проекта в обозревателе решений и называем Parser:

Далее в эту папку добавляем существующие программные модули:

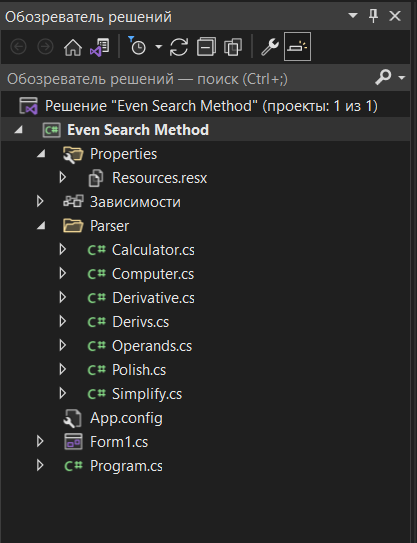


Рис 2.

Далее добавляем ссылку на библиотеки Extreme. “Numerics.Net40.dll”

“Microsoft.Office.Interop.Excel” и “office”.

Шаг №7.3: В диалоговом окне менеджера ссылок ввести имя выше указанных файлов и добавит в проект.

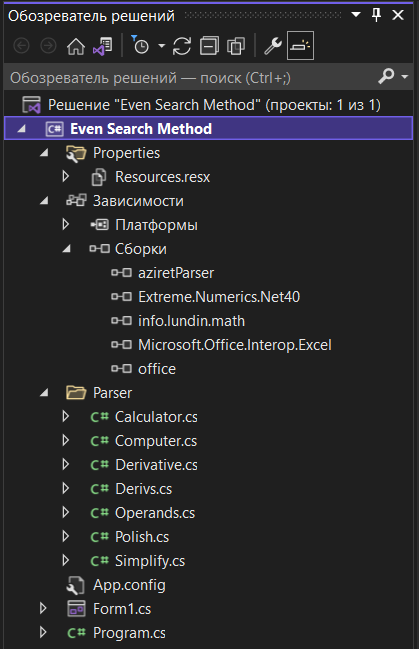


Рис 3

## Раздел №7: Тесты для проверки программного обеспечения, реализующего метод ESM.

Тест №1: Целью теста является проверка решения задачи для функции f(x) = (x – 4)^2, при заданной начальной точке X = -3 и Tol = 1E-2 (нахождение, минимума).

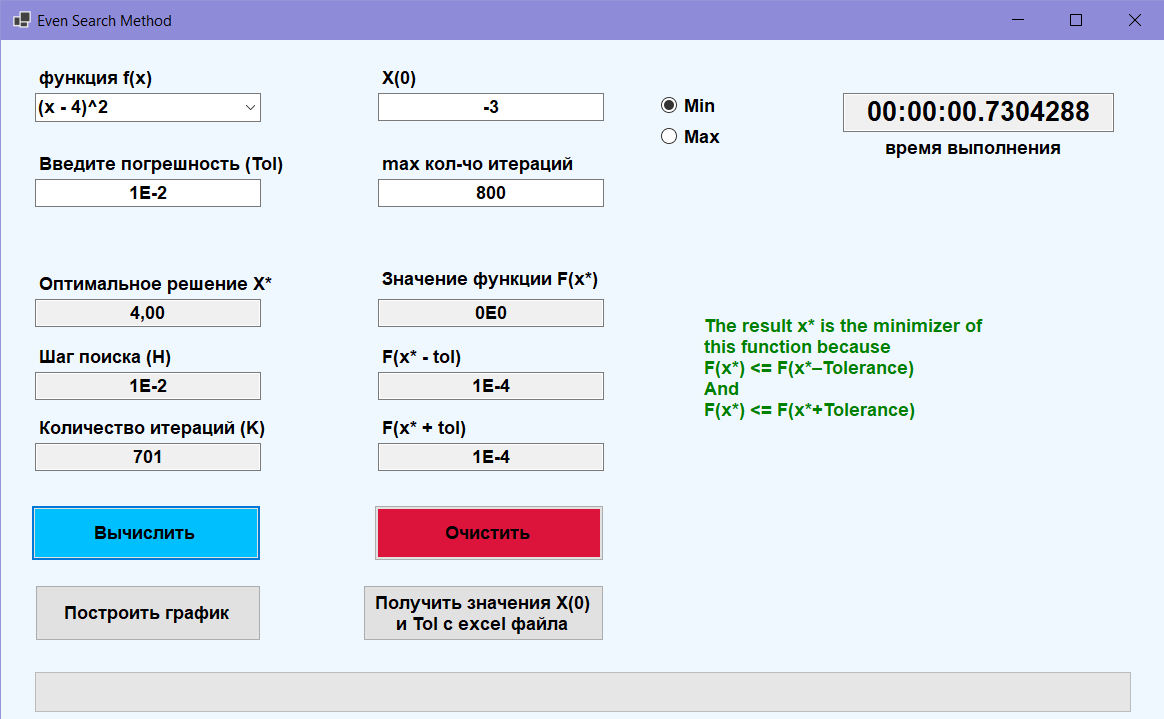


Рис 4

Тест №2: Целью теста является проверка решения задачи для функции f(x) = (x – 4)^2, при заданной начальной точке X = -3 и Tol = 1e-2 (нахождение максимума).

Если пользователь нажмет на «ДА» то открывается Excel файл и текущая функция вставляется в нужные поля.

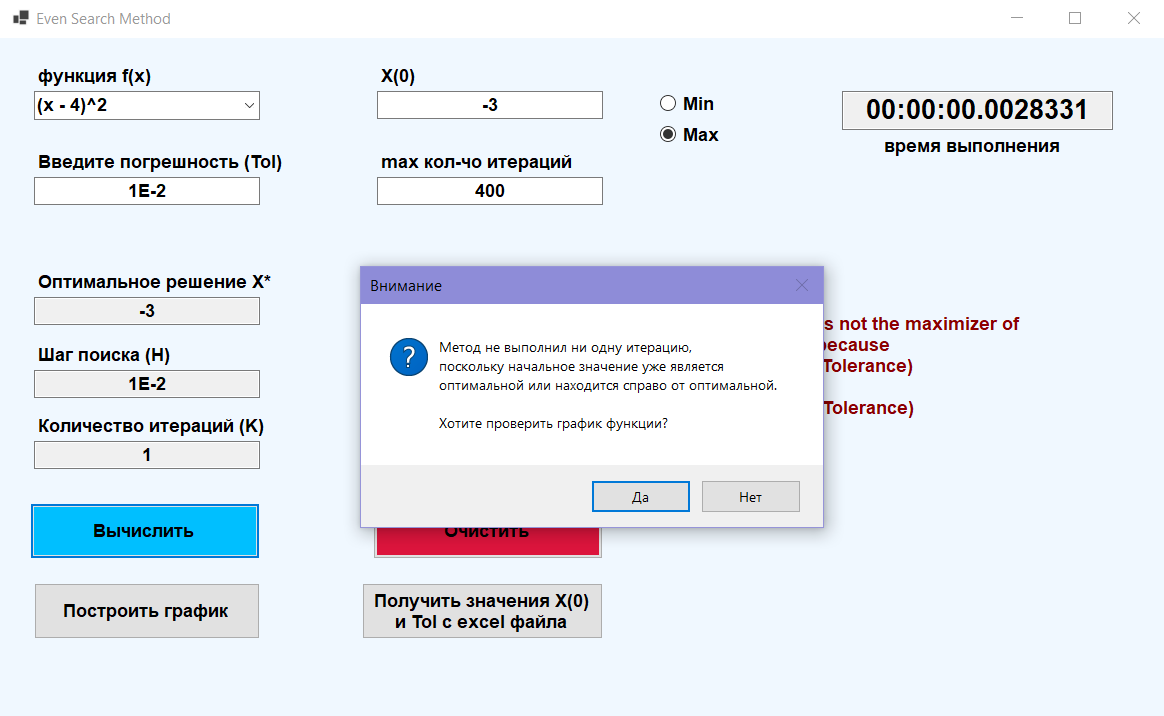


Рис 5

Тест №3: Целью теста является проверка решения при введении некорректных данных

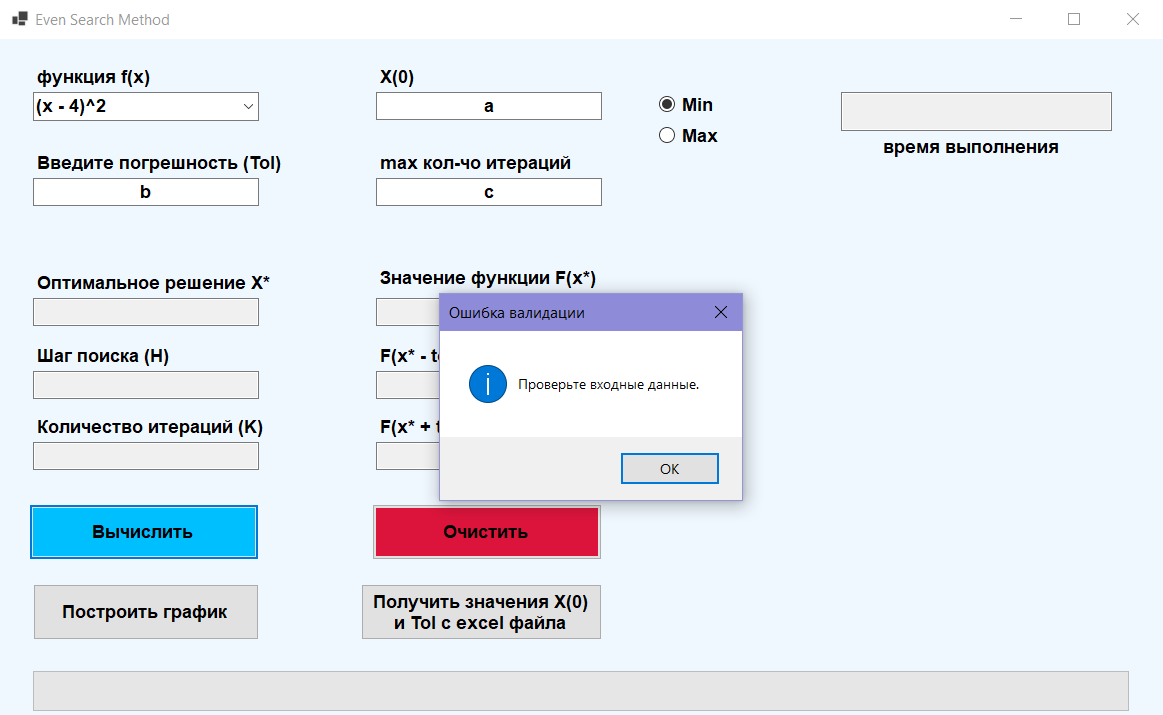


Рис 6.

Тест №4: Целью теста является проверка решения задачи для функции

f(x) = x^3 + x^2 - x + 1, при заданной начальной точке X = -2 и Tol = 1e-2 (максимум).

Если решения не найдено за указанное количество итераций,

Выходит, сообщение который видите на картине.

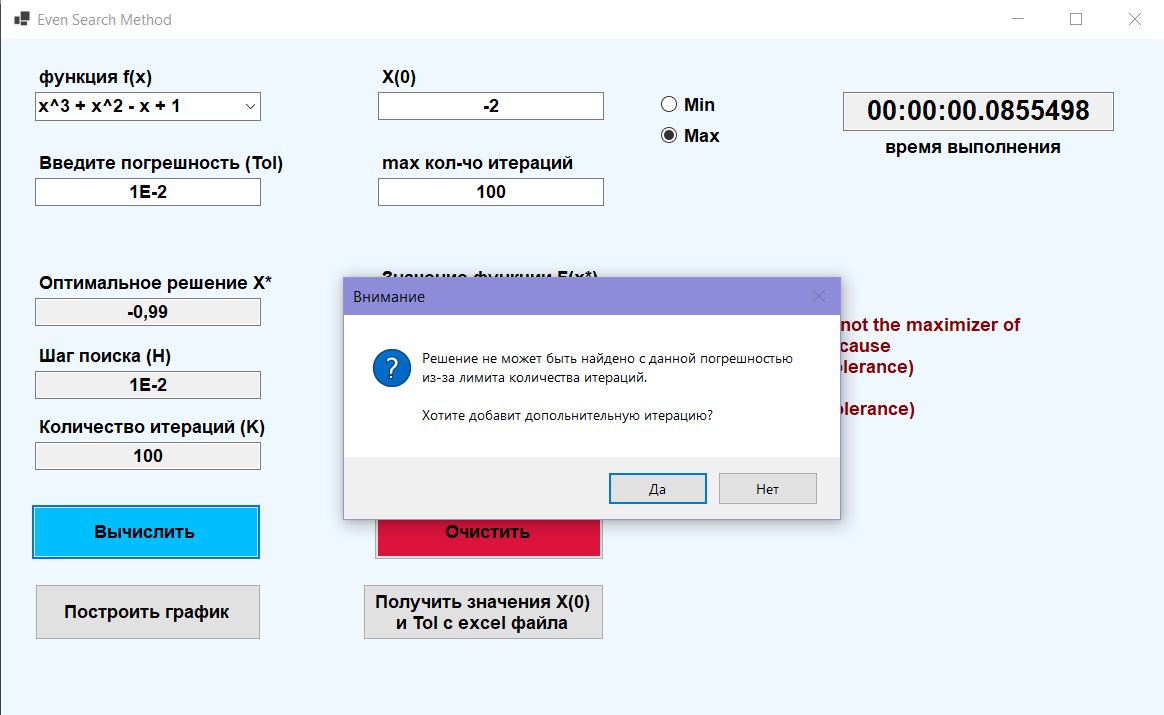


Рис 7

Если пользователь нажмет “ДА” текущее значение итерацию(max) умножается на два.

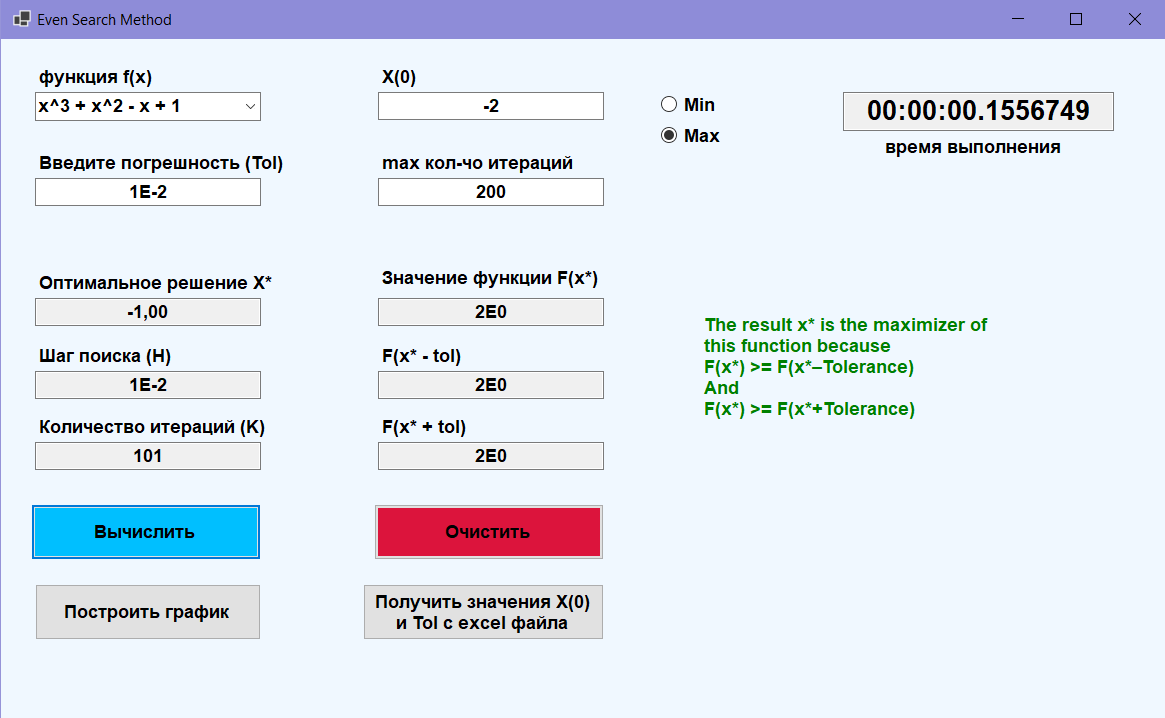


Рис 8.

Тест №5: Целью теста является проверка на входные данные х0. Если пользователь выбрал х0 с правой стороны оптимальной точки, тогда он попадает в исключительную ситуацию. В таком случае выскакивает сообщение что метод не выполнил ни одной итерации и предложит проверить график функции.

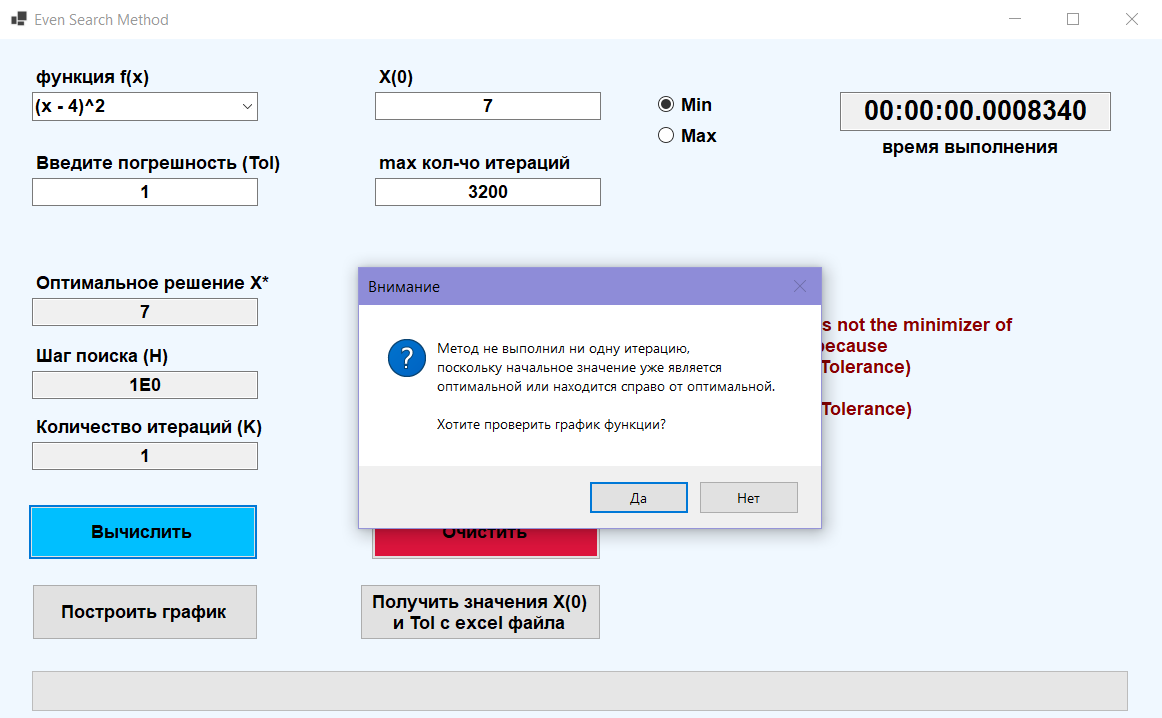


Рис 9.

При нажатии на кнопку «ДА» открывается excel файл и подставляются значения с программы.

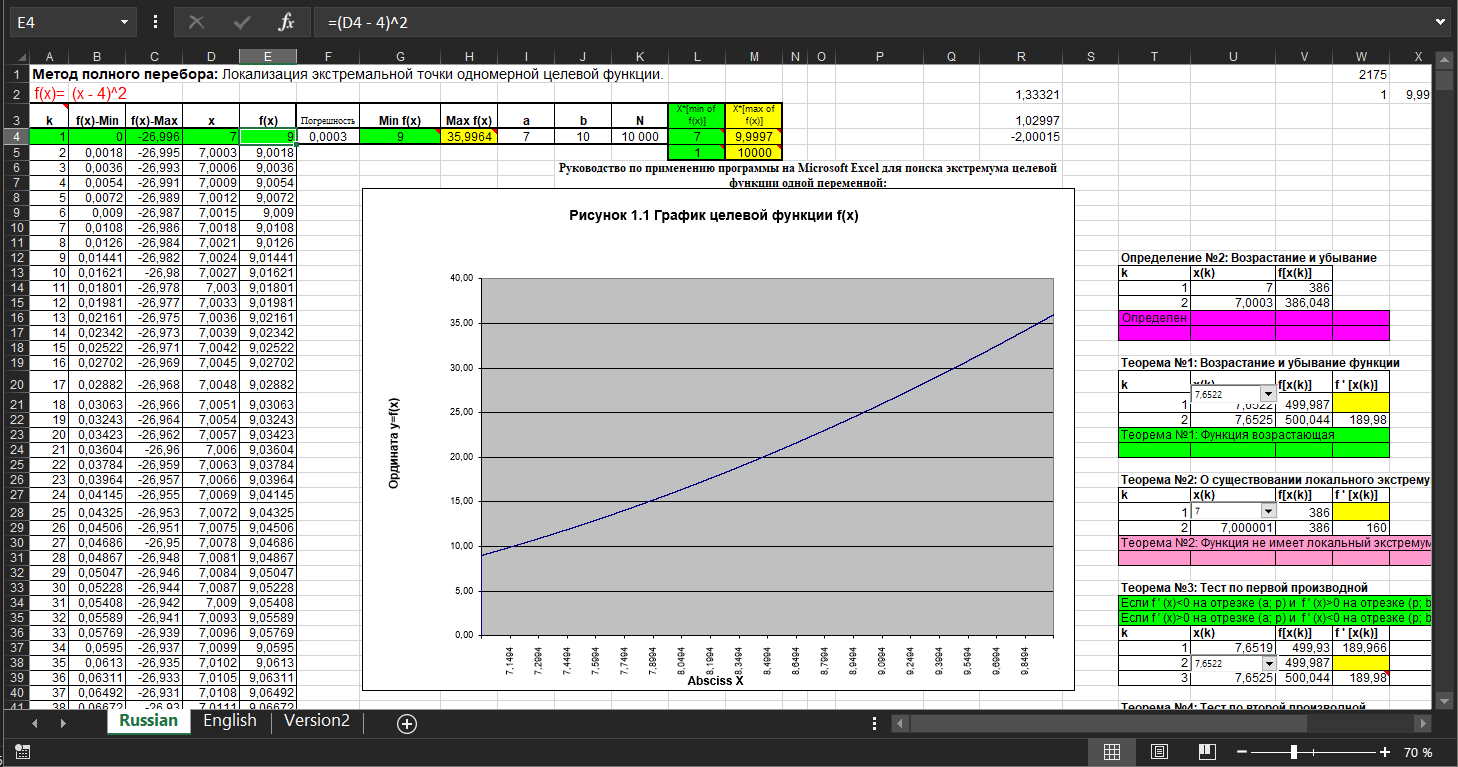


Рис 10.

После чего пользователь подбирает значение «а», так чтобы она была с левой стороны оптимальной точки.

Далее пользователь должен сохранить изменения и закрыть excel файл (обзятельно).

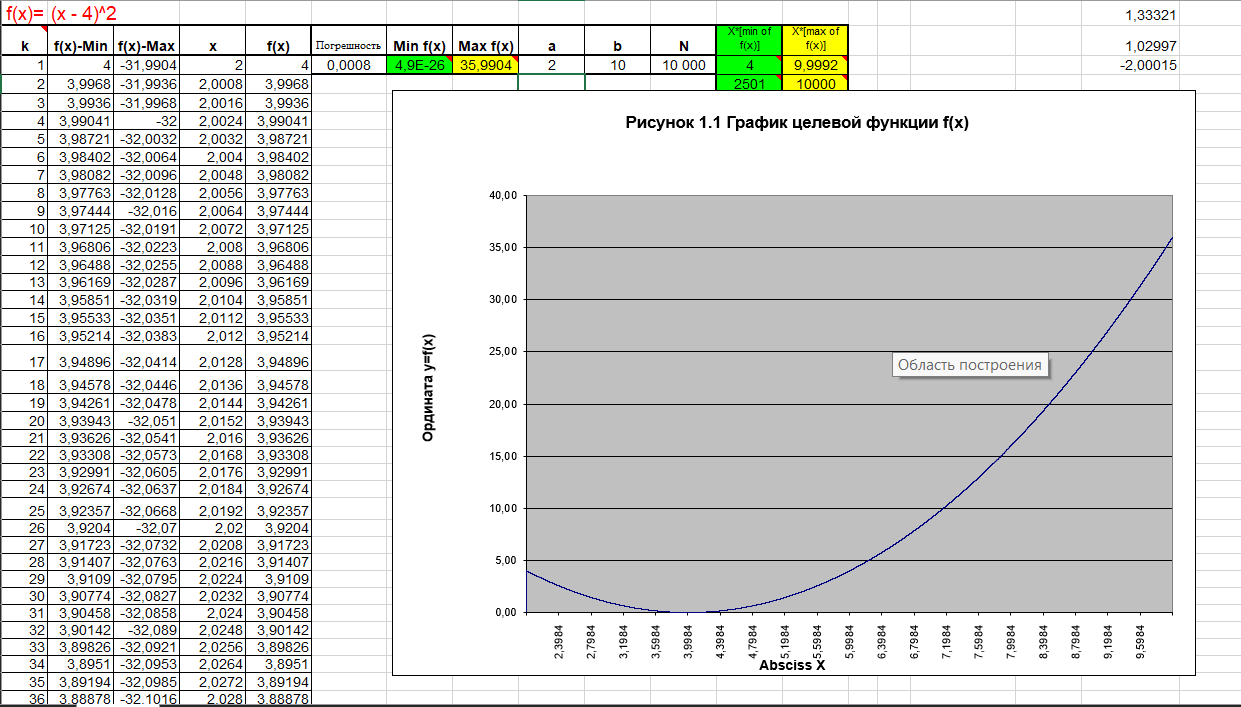


Рис 11.

После закрытия excel файла пользователь может нажать на кнопку «Получить значение x0 и Tol с excel файла» и вставить значение в TextBox1(X0) и TextBox1(Tol);

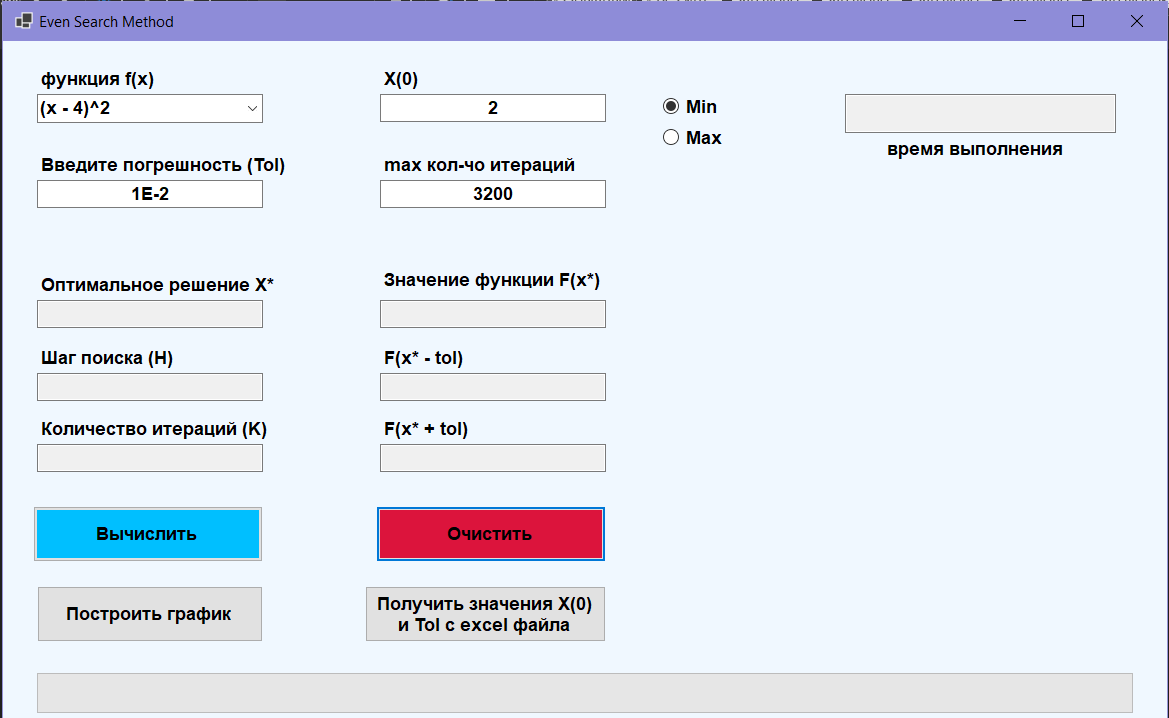


Рис 12.

При повторном нажатии на кнопку «Вычислить» программа с корректными данными находит решение!

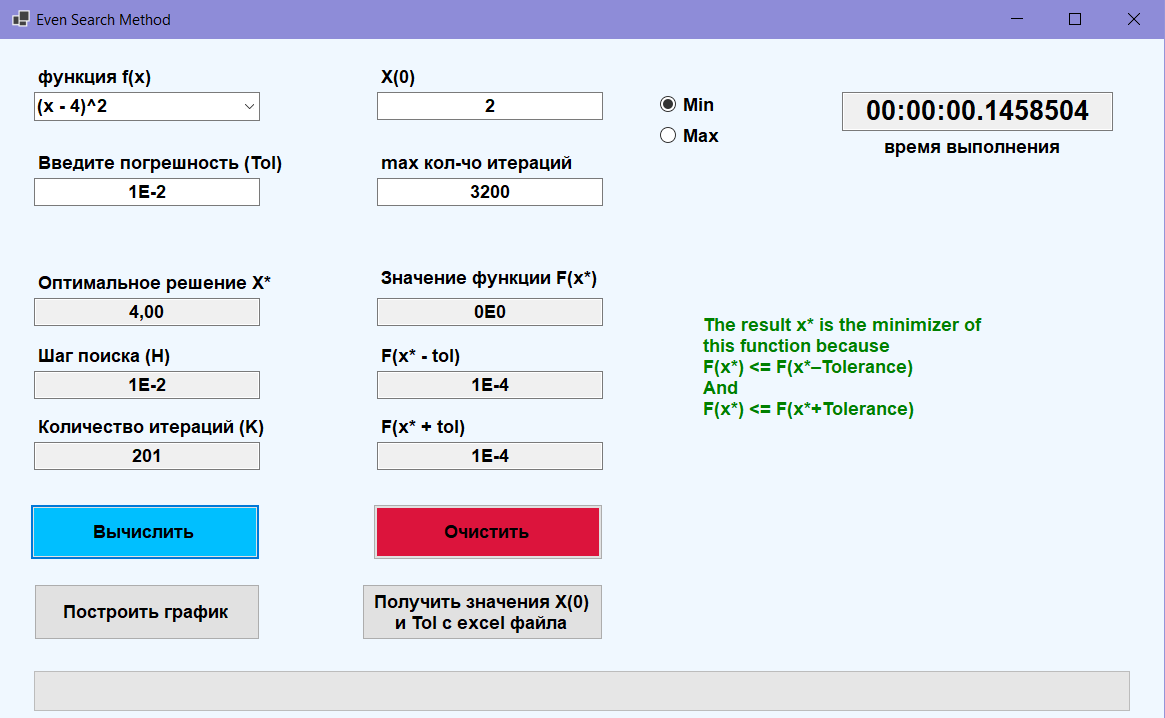


Рис 13.