

## Методика выполнения контрольного задания № 1.4

### «Вычисление всех нулей функции на заданном интервале методом дихотомии»

#### Постановка задачи

Построить алгоритм и отладить консольное **win32**–приложение, предназначенное для приближенного вычисления таблицы нулей заданной функции  $f_0(x)$ :

$$f_0(x) = \sum_{k=0}^{\infty} \frac{(-1)^k \cdot \left[ \cos\left(\frac{b}{k+1}x\right) + \sin\left(\frac{b}{k+a}x\right) \right]}{4 \cdot (2k+a) \cdot (k+1)!} \left(\frac{11x}{7}\right)^{k+1}, \quad (1)$$

на интервале  $x \in [x_0, x_n]$  с заданным количеством строк  $n$  и абсолютной погрешностью  $\varepsilon$ , где  $a$  и  $b$  – некоторые заданные значения вещественных параметров данной функции.

Для решения поставленной задачи следует воспользоваться уже имеющимися программными компонентами из библиотеки **MAC\_DLL** и при необходимости дополнить их.

Вычисленная таблица нулей функции  $f_0(x)$  должна быть «сохранена» в списке **Roots**, что генерируется в соответствующем экземпляре класса **MyTableOfFunction**.

Обратите внимание на то, что математическая функция  $f_0(x)$ , нули которой Вам предстоит определить в рамках данного Контрольного Задания 1.4, полностью совпадает с функцией, которую вы «программировали» в ходе выполнения Контрольного Задания 1.3.

#### Требования к программным компонентам

Основная расчетная программа должна быть разработана в виде отдельного проекта **MAC\_CheckTask\_1\_4** консольного приложения **win32** в рабочем пространстве **PetrenkoIN**. Собственно создание таблицы функции  $f = f_0(x)$ , должно быть реализовано в экземпляре соответствующего класса – **MyTableOfFunction** (сокращенно **MyTF**).

Для вывода вычисленной таблицы нулей функции  $f_0(x)$  в файл результатов **Test\_CT\_1\_4.txt** необходимо использовать метод **Table\_of\_Roots()**.

В имеющемся рабочем пространстве **MAC\_Petrenko** сгенерируем (добавим) новый проект **MAC\_CheckTask\_1\_4** консольного приложения.

По условию задания в классе **Main\_CT\_1\_4** мы должны сформировать статический метод **dummy\_fx()**, предназначенный для вычисления значений заданной функции  $f_0(x)$ .

Скопируем его из программы **Main\_CT\_1\_3**.

Пусть таблица параметров Контрольного Задания 1.4 имеет следующий вид:

$a$	$b$	$x_0$	$x_n$	$\varepsilon$	$n$
1.0	5.0	1.0	7.0	$10^{-10}$	900

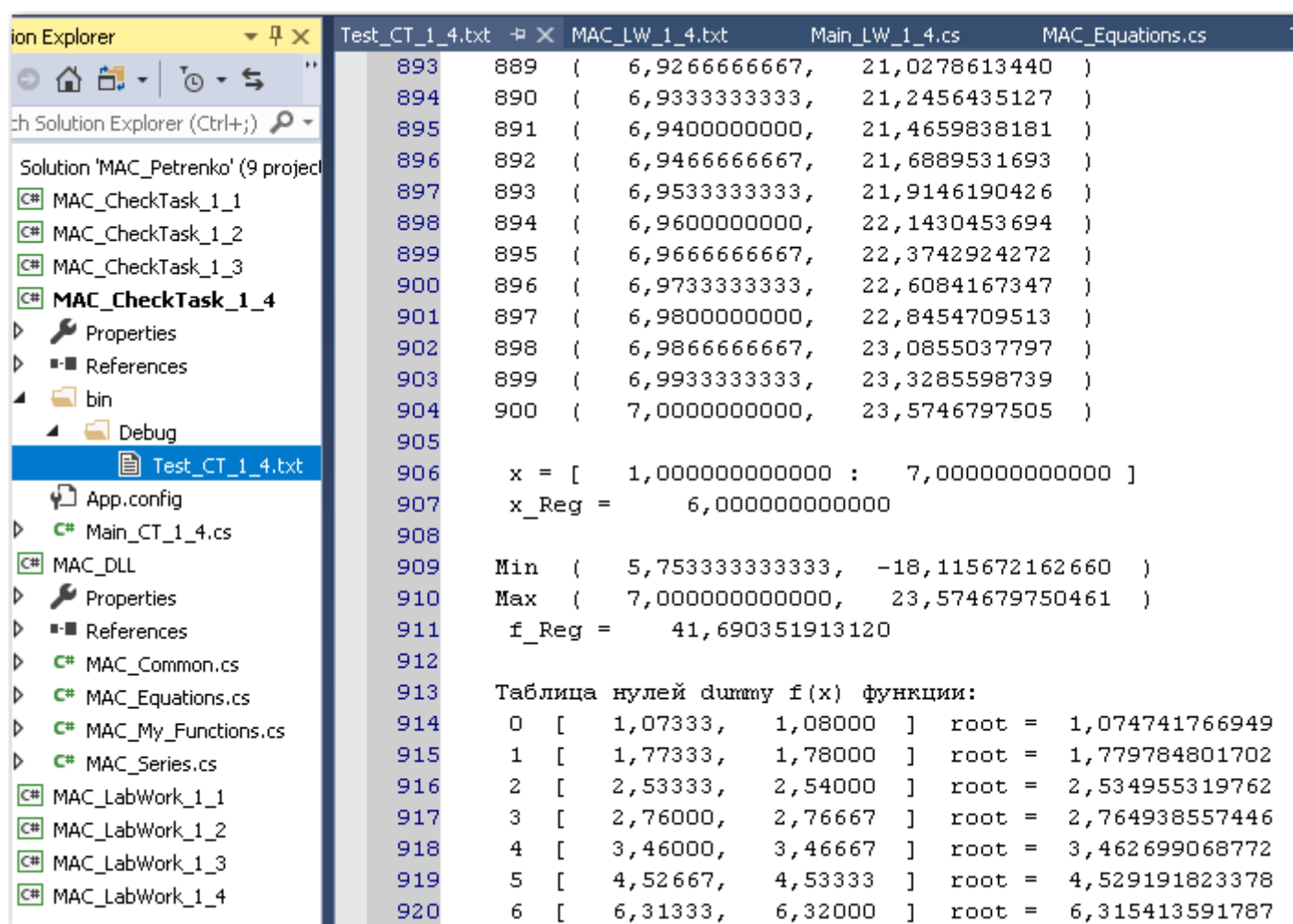
С учетом указанных значений параметров имеем код основного приложения:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using MyF = MAC_DLL.MAC_My_Functions;
7  using MyTF = MAC_DLL.MyTableOfFunction;
8  using System.IO;
9
10 namespace MAC_CheckTask_1_4
11 {
12     class Main_CT_1_4
13     {
14         public static double par_a, par_b, par_e;
15
16         static void Main(string[] args)
17         {
18             par_a = 1.0; par_b = 5.0; par_e = 1.0E-10;
19
20             MyTF TF = new MyTF(1.0, 7.0, 900, dummy_fx, "dummy f(x)");
21             TF.Roots_Correction(par_e);
22             StreamWriter SW = new StreamWriter("Test_CT_1_4.txt");
23             SW.WriteLine(TF.Table_of_Roots(" Roots of dummy f(x)"));
24             int n = TF.Roots.Count;
25             if (n > 1)
26             {
27                 double minR = TF.Roots[0].x, maxR = TF.Roots[n - 1].x;
28                 SW.WriteLine(" Minor Root = {0:F10}", minR);
29                 SW.WriteLine(" Major Root = {0:F10}", maxR);
30                 SW.WriteLine(" Distance = {0:F10}", maxR - minR);
31                 SW.Close();
32             }
33         }
34         public static double dummy_fx(double x)
35         {
36             return MyF.f0(x, par_a, par_b, par_e);
37         }
38     }
39 }

```

Файл результатов «присоединяется» к проекту:



В качестве одного из результатов вычислений в основной программе проекта **MAC\_CheckTask\_1\_4** определяется расстояние **distance** между максимальным и минимальным нулями заданной функции (корнями уравнения  $f_0(x) = 0$ ).

На консоль выведена таблица корней и ее младший и старший корни:

