

Методика выполнения контрольного задания № 1.2

«Вычисление значений степенных рядов (спецфункций)»

Рассмотрим «технологию» составления алгоритма, предназначенного для вычисления значений «модельной» спецфункции, имеющей следующее функциональное представление в виде степенного ряда:

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

где a и b – некоторые дополнительные числовые параметры.

Функция $f_0(x)$ имеет более сложное математическое представление, чем у элементарных трансцендентных функций, рассмотренных в ходе выполнения Лабораторной работы 1.2.

Однако это не мешает нам воспользоваться для функции $f_0(x)$ уже известным нам подходом при составлении вычислительного алгоритма и соответствующего ему кода.

Выполним предварительные преобразования.

Во-первых, вынесем за знак суммы множители, общие для всех слагаемых:

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

Во-вторых, выделим первое слагаемое (при $k = 0$) из общей суммы:

$$f_0(x) = \frac{11x}{28} \times \left\{ \frac{\cos(bx) + \sin\left(\frac{b}{a}x\right)}{a} + \sum_{k=1}^{\infty} \frac{(-1)^k \cdot \left[\cos\left(\frac{b}{k+1}x\right) + \sin\left(\frac{b}{k+a}x\right) \right]}{(2k+a) \cdot (k+1)!} \left(\frac{11x}{7}\right)^k \right\}.$$

Обращаем внимание на тот факт, что *часть* выражения, стоящего под знаком суммы, может быть вычислена рекуррентным путем, т.е. через соответствующее его «предыдущее» значение.

Поэтому вводим «привычное» обозначение:

$$p_k = \frac{(-1)^k}{(k+1)!} \cdot \left(\frac{11x}{7} \right)^k,$$

тогда

$$p_0 = 1 \text{ при } k = 0,$$

и

$$p_k = -p_{k-1} \cdot \frac{\tilde{x}}{k+1}, \text{ при } k > 0, \quad \text{где } \tilde{x} = \frac{11x}{7}. \quad (2)$$

Таким образом, k – е слагаемое, стоящее под знаком суммы представимо в виде

$$A_k = \frac{p_k}{2k+a} \cdot \left[\cos\left(\frac{bx}{k+1}\right) + \sin\left(\frac{bx}{k+a}\right) \right]$$

и

$$A_0 = \frac{1}{a} \cdot \left[\cos(bx) + \sin\left(\frac{bx}{a}\right) \right]. \quad (3)$$

Тогда с учетом всех обозначений заданная функция (1) может быть вычислена по следующей формуле:

$$f_0(x) = \frac{\tilde{x}}{4} \left\{ A_0 + \sum_{k=1}^{\infty} A_k \right\}. \quad (4)$$

Приступаем к непосредственному программированию представленных выше формул (2)–(4), т.е. составлению кода метода, высчитывающего значения заданной функции $f_0(x)$.

Отметим для себя сразу, что разрабатываемая функция должна принимать в качестве входных параметров четыре величины типа **double**.

Первая из них – непосредственно переменная x , две другие – a и b – математические параметры функции $f_0(x)$, последняя – eps – абсолютная погрешность вычислений.

Размещать код этой функции будем в уже имеющемся классе **MAC_My_Functions** библиотеки **MAC_DLL**.

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MAC_DLL
8  {
9      public class MAC_My_Functions
10     {
11         public static double f0(double x, double a, double b, double eps)
12         {
13             if (x == 0) return 0.0;
14             double xz = 11.0 * x / 7.0, bx = b * x;
15             double Ao = (Math.Cos(bx) + Math.Sin(bx / a)) / a;
16             double pk = 1.0, Ak = 1.0, summa = 0.0;
17
18             for (int k = 1; Math.Abs(Ak) > eps; k++)
19             {
20                 pk = -pk * xz / (k + 1.0);
21                 Ak = (Math.Cos(bx / (k + 1.0)) + Math.Sin(bx / (k + a)))
22                     * pk / (2.0 * k + a);
23                 summa += Ak;
24             }
25             return 0.25 * xz * (Ao + summa);
26         }
27
28         public static double MySin(double x, double eps) ...
29
30         public static double MyCos(double x, double eps) ...
31
32         public static double MyExp(double x, double eps) ...
33
34         public static double MySinh(double x, double eps) ...
35         public static double MyCosh(double x, double eps) ...
36     }
37 }

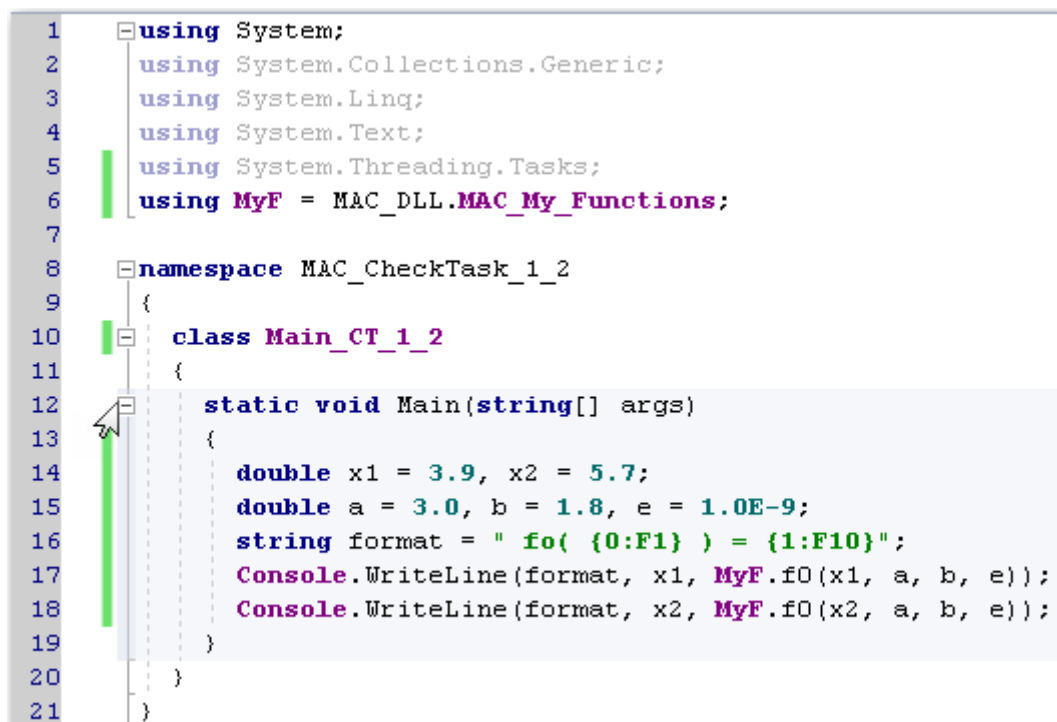
```

Вычисления значений данной функции будем выполнять в рамках отдельного проекта консольного приложения **MAC_CheckTask_1_2**, который добавим в имеющееся рабочее пространство **PetrenkoIN**.

Для удобства сразу переименуем класс **Program** в класс **Main_CT_1_2** и стандартным способом добавим библиотеку **MAC_DLL** в раздел **References** данного проекта.

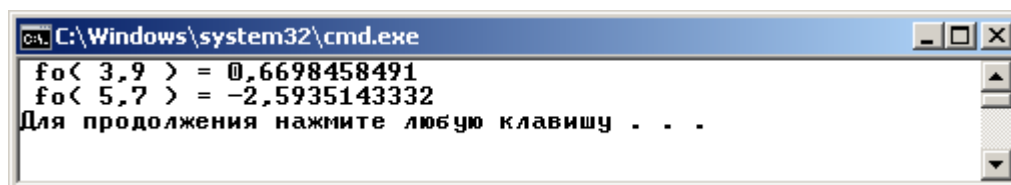
Пусть нам требуется вычислить два значения исследуемой функции $f_0(x)$, соответствующие значениям $x_1 = 3.9$ и $x_2 = 5.7$ ее аргумента, при заданных значениях $a = 3.0$ и $b = 1.8$ ее параметров. При этом абсолютная погрешность вычисления значений функции $f_0(x)$ должна быть не хуже, чем 10^{-9} .

Окончательный вид приложения **MAC_CheckTask_1_2** может быть следующим:



```
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
8  namespace MAC_CheckTask_1_2
9  {
10     class Main_CT_1_2
11     {
12         static void Main(string[] args)
13         {
14             double x1 = 3.9, x2 = 5.7;
15             double a = 3.0, b = 1.8, e = 1.0E-9;
16             string format = " fo( {0:F1} ) = {1:F10}";
17             Console.WriteLine(format, x1, MyF.f0(x1, a, b, e));
18             Console.WriteLine(format, x2, MyF.f0(x2, a, b, e));
19         }
20     }
21 }
```

Имеем искомый результат выполнения основной программы:



```
C:\Windows\system32\cmd.exe
fo< 3.9 > = 0.6698458491
fo< 5.7 > = -2.5935143332
Для продолжения нажмите любую клавишу . . .
```

На этом методическая часть руководства заканчивается.

Уяснив «технологию» составления алгоритма для функции $f_0(x)$ вида (1), Вы должны выполнить аналогичное проектирование для индивидуального варианта функции и значений ее параметров.

Разрабатываемая функция должна быть также размещена в классе **MAC_My_Functions** библиотеки **MAC_DLL** под именем **my_f**.

Для программирования вычислений следует использовать проект **MAC_CheckTask_1_2**.

Ваш индивидуальный вариант, как и все остальные контрольные задания, выдается Вам на специальном бланке.

На левой стороне бланка размещается собственно Ваше задание.

На правой стороне бланка приводятся контрольные результаты вычислений Вашей функции, соответствующие значениям параметров задачи, *отличающимся* от тех, что размещены на левой части бланка.

Это поможет Вам выполнить самостоятельную отладку разрабатываемого алгоритма и кода для заданной функции.

Если отладка выполнена успешно – Вы получили контрольные результаты, то Вы можете выполнить вычисления для значений параметров задачи из «Таблицы контрольных параметров» и внести соответствующие результаты вычислений в итоговую «Таблицу контрольных результатов».

Образец заполнения такой таблицы приводится на правой стороне бланка контрольного задания (где данные соответствуют тестовым вычислениям).