Лабораторная работа по теме
«*Тема 4.6. Программирование алгоритмов итеративных циклических структур*»

Перейти к [ЛП 4.5] [ЛП 4.7] [Огл.]

Цель данной лабораторной работы состоит в освоении формализации при решении задач на компьютере, а также в изучении средств, приемов и получении практических навыков разработки, написания и отладки проектов, использующих итеративные циклические структуры.

4.6.1. Вопросы, подлежащие изучению

1. Алгоритмы организации итеративных циклических структур: цикл с предусловием; цикл с постусловием.
2. Базовые алгоритмы, использующие итеративные циклические структуры: алгоритм вычисления суммы (или произведения) членов бесконечной последовательности; алгоритмы вычислений по итеративным формулам.
3. Оператор, реализующий выполнение итеративного цикла, `Do/Loop` и его основные конструкции: `Do While…Loop; Do Until…Loop; Do...Loop While…; Do…Loop Until… ; Do... Exit Do…Loop.`

4.6.2. Задание

1. ***Выбрать вариант задания*** из таблицы 4.6-1 по усмотрению преподавателя.
2. ***Провести формализацию*** поставленной задачи.
3. ***Составить схему алгоритма*** решения поставленной задачи.
4. ***Разработать интерфейс*** пользователя. В этом интерфейсе предусмотреть отображение на форме номера итерации и значения вычисляемого члена бесконечной последовательности или корня уравнения.
5. ***Написать программный код*** процедур пользователя в соответствии со схемами алгоритмов. Обмен данными между процедурами должен осуществляться через параметры, без использования глобальных переменных.
6. ***Написать программный код*** проекта. ***Событийная процедура*** должна содержать только операторы вызова пользовательских (общих) процедур.
7. ***Выполнить созданный проект***.
8. ***Получить решение***.
9. ***Обосновать правильность полученных результатов*** на заранее разработанных тестах.

4.6.3. Варианты задания

Таблица 4.6-1

1)	Вычислить с точностью **ε = 0.00001** константу Эйлера (основание натурального логарифма), воспользовавшись разложением в ряд: $$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!} + \ldots$$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.

2)	Вычислить и вывести те члены последовательности, $x, \frac{x^2}{2!}, \frac{x^3}{3!}, \ldots \frac{x^n}{n!}, \ldots$ значения которых больше **ε = 0.001** при **x = 0.2**.
3)	Вычислить **arctg(x)** с точностью **ε = 0.0001**, воспользовавшись разложением в ряд: $arctg(x) = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3 \cdot x^3} - \frac{1}{5x^5} + \ldots + \frac{(-1)^{n+1}}{(2n+1) \cdot x^{2n+1}} + \ldots$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции при x=1.5.
4)	Вычислить с точностью **ε = 0.00001** значение функции $y = \sqrt{x}$ при **x = 2**, воспользовавшись рекуррентной формулой: $y_{i+1} = 0.5\left[y_i + \frac{x}{y_i}\right]; \quad i = 0,1,2\ldots; y_0 = \frac{x}{2}$. Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
5)	Вычислить константу π с точностью до **ε = 0.00001**, воспользовавшись разложением в ряд: $arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots + (-1)^n \frac{x^{2n+1}}{2n+1} + \ldots$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
6)	Вычислить с точностью **ε = 0.00001** значение функции $y = \frac{1}{\sqrt{x}}$ при **x = 2**, воспользовавшись формулой: $y_{i+1} = 1.5y_i - 0.5xy_i^3; i = 0,1,2,\ldots; y_0 = 1$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции
7)	Вычислить **sin 0.5** с точностью **ε = 0.0001**, воспользовавшись разложением в ряд: $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \ldots$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
8)	Вычислить e^{-x} с точностью **ε = 0.00001**, воспользовавшись разложением в ряд: $e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \ldots + (-1)^n \frac{x^n}{n!} + \ldots$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
9)	Вычислить **cos 0.6** с точностью **ε = 0.00001**, воспользовавшись разложением в ряд: $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots + (-1)^n \frac{x^{2n}}{(2n)!} + \ldots$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.

10)	Вычислить с точностью **ε = 0.0001** корень уравнения $x - \cos x = 0$, воспользовавшись формулой: $\mathbf{x_{i+1} = \cos x_i ; i = 0,1,2..., x_0 = 0}$. Проверить правильность решения подстановкой найденного корня в уравнение.
11)	Вычислить и вывести те члены последовательности, $\frac{x^2}{2!}, -\frac{x^3}{3!}, ..., (-1)^n \frac{x^n}{n!}, ...$ значения которых по модулю больше **ε = 0.001** при **x = 0.5**.
12)	Вычислить $\ln\frac{1+x}{1-x}$ при |x|<1 с точностью до **ε = 0.0001**, воспользовавшись разложением в ряд: $\ln\frac{1+x}{1-x} = 2\cdot[x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + ... + \frac{x^{2n+1}}{(2n+1)} + ...]$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
13)	Вычислить корень уравнения $f(x) = x^4 + 2x^2 - x - 1 = 0$ с точностью **ε=0.0001**, воспользовавшись итерационной формулой $$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}; i = 0,1,2,...; x_0 = 0$$ Проверить правильность решения подстановкой найденного корня в уравнение.
14)	Вычислить значение $\sqrt{2}$ с точностью **ε = 0.00001** , воспользовавшись представлением в виде в виде цепной дроби: $$\sqrt{2} - 1 = \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{...}}}$$ Значение дроби равно пределу числовой последовательности, члены которой вычисляются по рекуррентной формуле до достижения заданной точности $$\mathbf{a_n = \frac{1}{2 + a_{n-1}}, n = 1,2,3,..., \quad a_0 = 0.5.}$$. Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
15)	Вычислить и вывести те члены последовательности, $$\frac{x^3}{3}, \frac{-x^5}{15}, ..., (-1)^{n+1}\frac{x^{2n+1}}{4n^2 - 1}$$ значения которых по модулю больше **ε = 0.001** при **x = 0.3**.
16)	Вычислить **ln(x)** с точностью **ε = 0.0001**, воспользовавшись разложением в ряд: $\ln(x) = 2(\frac{(x-1)}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + ... + \frac{(x-1)^{2n+1}}{(2n+1)(x+1)^{2n+1}} + ...)$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции при x=1.5.

17	Вычислить **sh 0.3** с точностью до **ε= 0.00005**, воспользовавшись разложением в ряд: $\mathbf{sh(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \ldots + \frac{x^{2n+1}}{(2n+1)!} + \ldots}$ Сравнить результат со значением, полученным с помощью встроенной функции для вычисления **e^x**, используя соотношение: $sh(x) = \frac{e^x - e^{-x}}{2}$
18)	Вычислить корень уравнения **x-0.5(sinx^2-1)=0** с точностью **ε = 0.001,** воспользовавшись итерационной формулой: $x_{i+1} = 0.5(\sin x_i^2 - 1);\ \ i = 0,1,2,\ldots;\ \ x_0 = -0.25$ Проверить правильность решения подстановкой найденного корня в уравнение.
19	Вычислить **ln(2)** с точностью **ε = 0.001**, воспользовавшись представлением в виде ряда: $\ln 2 = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \ldots + (-1)^{n+1}\frac{1}{n} + \ldots$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
20)	Вычислить с точностью **ε = 0.00001** корень уравнения $f(x) = tg(x) - x = 0$ воспользовавшись итерационной формулой $\mathbf{x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, i = 0,1,2,\ldots, x_0 = 4.6.}$ Проверить правильность решения подстановкой.
21)	Вычислить ch 0.7 с точностью до **ε = 0.00005**, воспользовавшись разложением в ряд: $ch(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots + \frac{x^{2n}}{(2n)!} + \ldots$ Сравнить результат со значением, полученным с помощью встроенной функции e^x, используя соотношение: $ch(x) = \frac{e^x + e^{-x}}{2}$
22)	Вычислить приближенное значение бесконечной суммы с точностью **ε=0,0001** (справа от суммы дается выражение для проверки полученного результата): $1 + 2x + 3x^2 + 4x^3 + \ldots + nx^{n-1}$ (для |x|<1 сумма равна $\frac{1}{(1-x)^2}$)
23)	Вычислить $\ln\frac{x+1}{x-1}$ при |x|>1 с точностью до **ε = 0.0001**, воспользовавшись разложением в ряд: $\ln\frac{x+1}{x-1} = 2\cdot[\frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \ldots + \frac{1}{(2n+1)x^{2n+1}} + \ldots]$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции.
24)	Вычислить **ln(x+1)** с точностью **ε = 0.0001**, воспользовавшись разложением в ряд: $\ln(x+1) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots + (-1)^n\frac{x^{n+1}}{n+1} + \ldots$ Сравнить результат со значением, полученным с помощью соответствующей встроенной функции при x=0.5.
25)	Вычислить и вывести те члены последовательности, $3x, 8x^2, \ldots n(n+2)x^n, \ldots$, значения, которых больше **ε = 0.01,** при **x = 0.6.**

26)	Найти наименьшее целое положительное **n**, при котором: $$1+\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{3}}+\ldots+\frac{1}{\sqrt{n}}+\ldots>4$$
27)	Пусть $y_0=0;\ y_k=\dfrac{y_{k-1}+1}{y_{k-1}+2};\ k=1,2,\ldots$. Дано действительное число **ε>0**. Найти первый член y_n, для которого выполнено условие $y_n-y_{n-1}<\varepsilon$.
28)	Вычислить приближенное значение бесконечной суммы с точностью **ε=0,0001** (справа от суммы дается ее точное значение, с которым можно сравнить полученный результат): $1+\frac{1}{2^2}+\frac{1}{3^2}+\ldots+$ **π²/6-1.**
29)	Вычислить приближенное значение бесконечной суммы с точностью **ε=0,0001** (справа от суммы дается ее точное значение, с которым можно сравнить полученный результат): $1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\ldots$ **π/4**
30)	Вычислить приближенное значение бесконечной суммы с точностью **ε=0,0001** (справа от суммы дается ее точное значение, с которым можно сравнить полученный результат): $\frac{1}{1\cdot 3}+\frac{1}{2\cdot 4}+\frac{1}{3\cdot 5}+\ldots$ **3/4.**
31)	Вычислить приближенное значение бесконечной суммы с точностью **ε=0,0001** (справа от суммы дается ее точное значение, с которым можно сравнить полученный результат): $\frac{1}{1\cdot 2\cdot 3}+\frac{1}{2\cdot 3\cdot 4}+\frac{1}{3\cdot 4\cdot 5}+\ldots+$ **1/4.**
32	Вычислить $\frac{\sin(x)}{x}$ с точностью **ε = 0.0001**, воспользовавшись разложением в ряд: $\frac{\sin x}{x}=1-\frac{x^2}{3!}+\frac{x^4}{5!}-\frac{x^6}{7!}+\ldots+\frac{(-1)^n\cdot x^{2n}}{(2n+1)!}$. Сравнить результат со значением, полученным с помощью соответствующей встроенной функции при x=0.5.
33)	Даны действительные числа **x, ε (x≠0, ε>0).** Вычислить с точностью ε: $\sum_{k=0}^{\infty}\frac{(-1)^k x^{4k+1}}{(2k)!\,(4k+1)}$
34)	Даны действительные числа **x, ε (x≠0, ε>0).** Вычислить с точностью ε: $\sum_{k=0}^{\infty}\frac{x^{2k}}{2^k k!}$
35)	Дано действительное **b<0.** Последовательность **a_1, a_2, …**образована по следующему закону: **$a_1=b$; $a_k=(a_{k-1}+1)/(1-\sin^2 k)$, k=2, 3, … .** Найти первый неотрицательный член последовательности.

4.6.4. Содержание отчета

1. Тема и название работы.
2. Задание на разработку проекта и вариант задания.
3. Формализация задания.
4. Разработка приложения
   - **4.1.** Графический интерфейс пользователя;
   - **4.2.** Таблица свойств объектов;
   - **4.3.** Схемы алгоритмов решаемой задачи;
   - **4.4.** Программный код с использованием процедур.
5. Результаты выполнения проекта.
6. Доказательство правильности работы программы.

4.6.5. Пример выполнения задания-1

1. *Тема и название работы*

Программирование алгоритмов итеративных циклических структур – Вычисление с точностью **$\varepsilon=10^{-5}$** корня заданного уравнения.

2. *Задание на разработку проекта и вариант задания*

Создать проект **Проект-4-6-1-Лаб** для вычисления с точностью $\varepsilon=10^{-5}$ корня уравнения $f(x)=x^3-2x^2+x-3=0$, воспользовавшись итерационной формулой

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}; \quad i = 0,1,2,\ldots, x_0 = 2.2.$$

Проверить правильность решения подстановкой найденного корня в уравнение.

Составить схему алгоритма и написать программный код в соответствии с заданием. Если необходимо, предварительно провести формализацию.

3. Формализация и уточнения задания

Вычислим производную **$f'(x)=3x^2-4x+1$**. Обозначим **x** – текущее приближение к корню, **a** – предыдущее приближение, **f** – значение функции **f(x)** для предыдущего значения, **p** – значение производной **f'(x)** для предыдущего значения, **i** – номер итерации, совпадающий с номером текущего приближения к корню уравнения, **y** – значение функции **f(x)** для найденного с заданной точностью корня уравнения.

Будем считать, что заданная точность **ε** обеспечена, если модуль разности между текущим и предыдущим значениями корня меньше точности **ε**, то есть для нашего случая **$|x-a|<\varepsilon$**.

Для решения поставленной задачи необходимо реализовать процедуру `Sub Kop( )`, которая в качестве входных параметров получает начальное значение **$x_0=2.2$** и точность **$\varepsilon=10^{-5}$**, и возвращает найденный корень **x_l**. Эта процедура для вычисления корня по заданной формуле должна использовать две процедуры `Function`: одна – `Funy()`, вычисляющая значение **f(x),** а другая – `Fproiz( )`– значение производной этой функции **f'(x).** Заметим, что процедуру `Sub Kop( )`можно было оформить как `Function`, так как она возвращает только одно значение – вычисленный корень уравнения.

4. Разработка приложения

4.1. Разработка графического интерфейса пользователя

Разработанная форма интерфейса пользователя приведена на рис. 4.6-1.

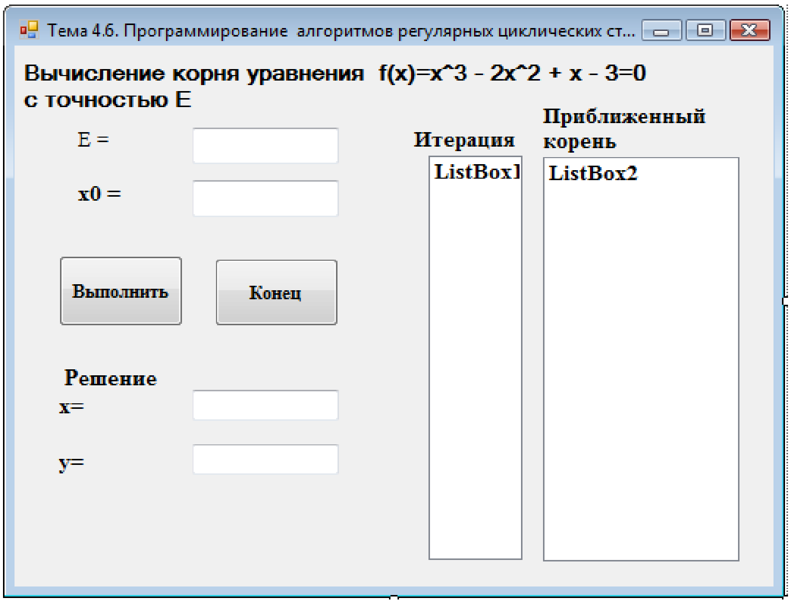

Рис. 4.6-1

4.2. Установка свойств объектов

Свойства объектов управления разработанной формы должны быть приведены в таблице 4.6-2.

Таблица 4.6-2

Объект	Свойство	Значение свойства
`Form1`	*Text*	*Тема 4.6. Программирование алгоритмов итеративных циклических структур*
`Label1`	*Name*	*Label1*
	Text	*Вычисление корня уравнения f(x)=x^3-2x^2+x-3=0*
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 10 пунктов*
`Label2`	*Name*	*Label2*
	Text	*E=*
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
`Label3`	*Name*	*Label3*
	Text	*X0=*
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
`Label4`	*Name*	*Label4*
	Text	*Итерация*
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Обычный, 8 пунктов*
`Label5`	*Name*	*Label5*
	Text	*Приближенный корень*
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Обычный, 8 пунктов*
`Label6`	*Name*	*Label6*

	Text	*Решение x= y=*
	ForeColor	*Черный*
	Font	*Arial, Жирный, 12 пунктов*
TextBox1	*Name*	*TextBox1*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Обычный, 8 пунктов*
TextBox2	*Name*	*TextBox2*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
TextBox3	*Name*	*TextBox3*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
TextBox4	*Name*	*TextBox4*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
ListBox1	*Name*	*ListBox1*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
ListBox2	*Name*	*ListBox2*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
Button1	*Name*	*Button1*
	Text	*Выполнить*
Button2	*Name*	*Button2*
	Text	*Конец*

4.3. Разработка схемы алгоритма

Схема алгоритма нахождения корня уравнения представлена на рис. 4.6-2.

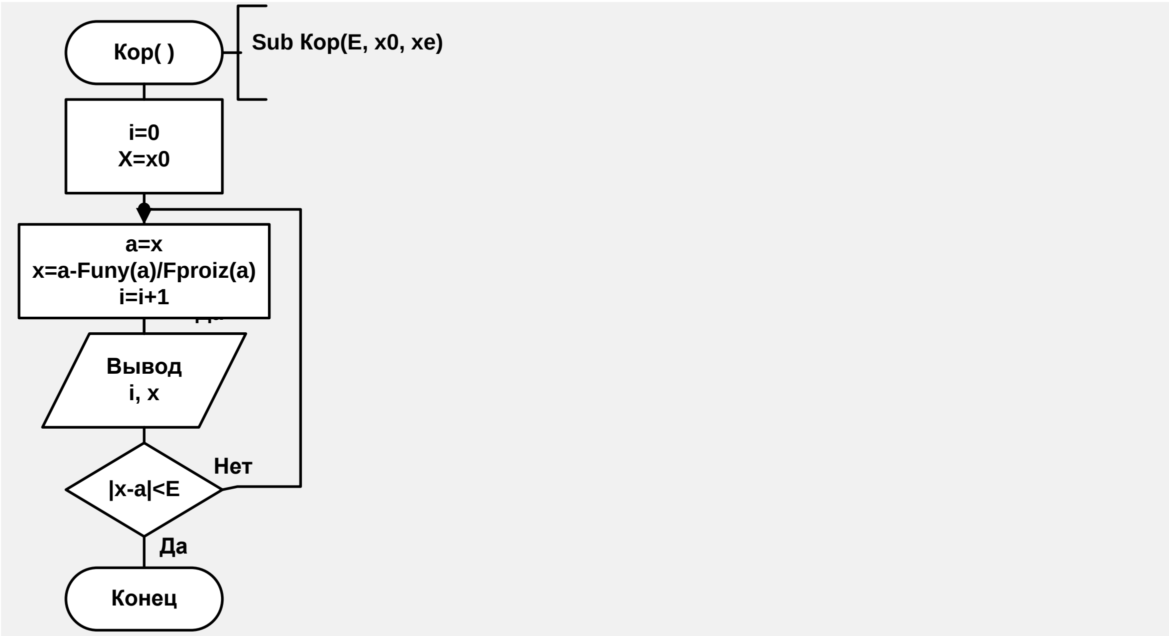

Рис. 4.6-2

4.4. Написание программного кода с использованием процедур

Программный код решаемой задачи представлен на рис. 4.6-3.

```
Option Strict On
Option Explicit On
Imports System.Math

Public Class Form1

   'Функция ввода исходн. данных из  TextBox
    Function vvod(ByVal T As TextBox) As Double
        Return Val(T.Text)
    End Function

   'Процедура  вывода вещественного результата  в TextBox
    Sub vivod(ByVal Z As Double, ByVal T As TextBox)
        T.Text = CStr(Z)
    End Sub


    'Процедура  вывода вещественного результата  в ListBox
    Sub vivodList(ByVal Z As Double, ByVal LB As ListBox)
        LB.Items.Add(CStr(Z))
    End Sub

    ' Процедура  вывода целого результата  в ListBox
    Sub vivodListint(ByVal Z As Integer, ByVal LB As ListBox)
        LB.Items.Add(CStr(Z))
    End Sub

    'процедура-Function, вычисляющая производную
    Public Function FProiz(ByVal x As Double) As Double
        Dim p As Double
        p = 3 * x ^ 3 - 4 * x + 1
        Return p
    End Function

'процедура-Function, вычисляющая заданную функцию
    Public Function Funy(ByVal x As Double) As Double
```

```
        Dim f As Double
        f = x ^ 3 - 2 * x ^ 2 + x - 3
        Return f
    End Function

   ' Процедура решения задачи поиска корня
    Public Sub Kop(ByVal E As Double, ByVal x0 As Double, _
                   ByRef xe As Double)
        Dim x, a As Double
        Dim i As Integer
        i = 0 : x = x0
        Do
            a = x : x = a - Funy(a) / FProiz(a) : i = i + 1
            vivodListint(i, ListBox1)
            vivodList(x, ListBox2)
        Loop Until Abs(x - a) < E
        xe = x
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles Button2.Click
        End
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles Button1.Click
        Dim EE, x0x0, xn, y As Double
        EE = vvod(TextBox1)
        x0x0 = vvod(TextBox2)
        Kop(EE, x0x0, xn)
        vivod(xn, TextBox3)
        y = Funy(xn)
        vivod(y, TextBox4)
    End Sub

End Class
```
Рис. 4.6-3

5. *Выполнение проекта и получение результатов*

Выполним проект на компьютере и получим следующий результат, приведенный на рис. 4.6-4.

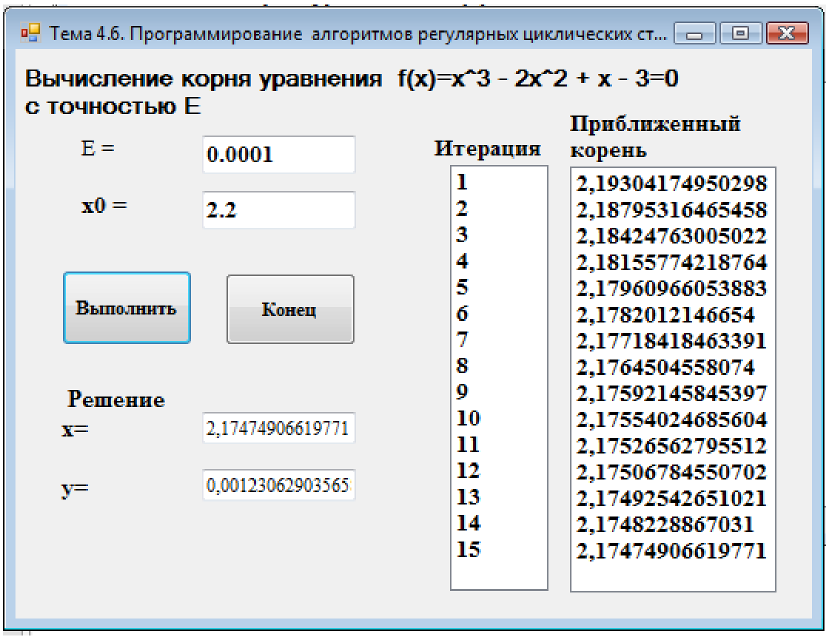

Рис. 4.6-4

6. *Доказательство правильности работы программ*

Значение функции при подстановке корня в уравнение **f(x)= 0.00012315320113**. Это говорит о том, что значение функции **f(2.17457839205816)=0.00012315320113** близко к нулю.

4.6.6. Пример выполнения задания-2

1. *Тема и название работы:*

Программирование алгоритмов итеративных циклических структур – Вычисление членов заданной последовательности, значения которых по модулю больше заданного числа.

2. *Задание на разработку проекта и вариант задания*

Создать проект **Проект-4-6-2-Лаб** для вычисления и отображения на экране тех членов последовательности

$$\frac{x-1}{1!};\ \frac{(x-1)^2}{2!};\ \frac{(x-1)^3}{3!};\ \ldots;\ \frac{(x-1)^n}{n!};\ldots,$$

значения которых по модулю больше ε=0.0001, при x=1.5.

3. *Формализация и уточнение задания*

Для решения поставленной задачи необходимо вывести рекуррентную формулу вычисления члена последовательности.

Очевидно, что выражение для n-го члена заданной последовательности имеет вид:

$$a_n = \frac{(x-1)^n}{n!}.$$

Тогда формула для (n+1) члена последовательности имеет вид:

$$a_{n+1} = \frac{(x-1)^{n+1}}{(n+1)!}.$$

Имея в виду, что (n+1)!=n! · (n+1), получим

$$\frac{a_{n+1}}{a_n} = \frac{(x-1)^{n+1}}{(n+1)!}\cdot\frac{n!}{(x-1)^n} = \frac{(x-1)}{n+1}.$$

Откуда получаем следующую рекуррентную формулу

$$a_{n+1} = a_n\cdot\frac{x-1}{n+1};\qquad a_1 = \frac{x-1}{1!};$$ -начальный член последовательности при n=1.

4. *Разработка приложения*

4.1. Разработка графического интерфейса пользователя

Разработанная форма интерфейса пользователя приведена на рис. 4.6-5.

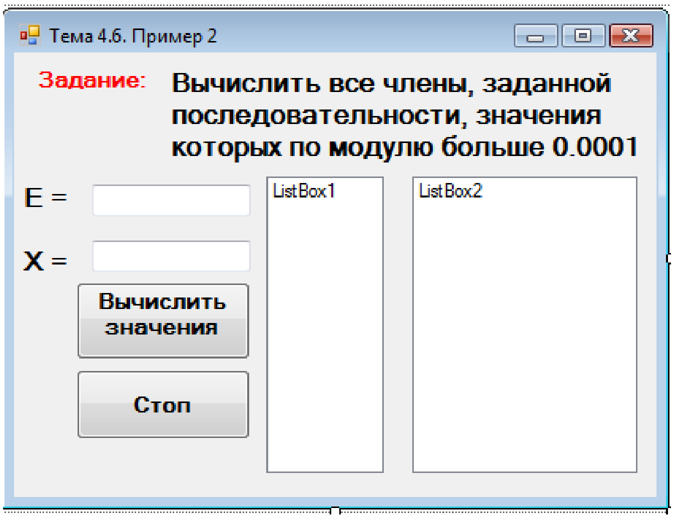

Рис. 4.6-5

4.2. Установка свойств объектов

Свойства объектов управления разработанной формы приведены в таблице 4.6-3.

Таблица 4.6-3

Объект	Свойство	Значение свойства
`Form1`	*Text*	*Тема 4.6. Пример 2.*
`Label1`	*Name*	*Label1*
	Text	*E=*
	ForeColor	*Синий*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
`Label2`	*Name*	*Label2*
	Text	*X=*
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
`Label3`	*Name*	*Label3*
	Text	*Задание:*
	ForeColor	*Красный*
	Font	*Microsoft Sans Serif, Жирный, 10 пунктов*
`Label4`	*Name*	*Label4*
	Text	*Вычислить все члены, заданной последовательности, значения которых по модулю больше 0.0001*
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
`TextBox1`	*Name*	*TextBox1*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 10 пунктов*
`TextBox2`	*Name*	*TextBox2*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
`ListBox1`	*Name*	*ListBox1*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
`ListBox2`	*Name*	*ListBox2*
	Text	
	ForeColor	*Черный*
	Font	*Microsoft Sans Serif, Жирный, 8 пунктов*
`Button2`	*Name*	*Button2*
	Text	*Вычислить значения*
`Button1`	*Name*	*Button1*
	Text	*Стоп*

4.3. Разработка схемы алгоритма

Схемы алгоритмов представлены на рис. 4.6-6.

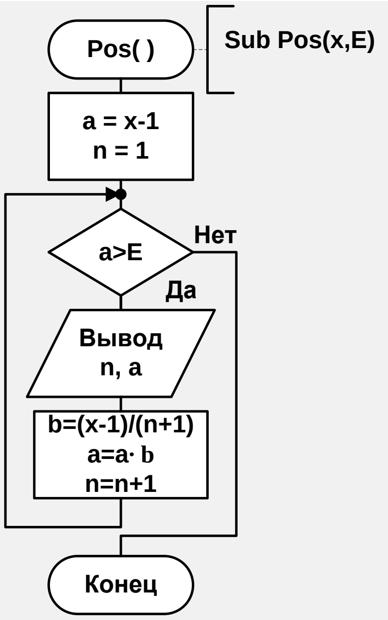

Рис. 4.6-6

4.4. Написание программного кода с использованием процедур

Программный код решаемой задачи представлен на рис. 4.6-7.

```
Option Strict On
Option Explicit On
Imports System.Math
Public Class Form1

    'Функция ввода исходн. данных из  TextBox
    Function vvod(ByVal T As TextBox) As Double
        Return Val(T.Text)
    End Function

    ' Процедура  вывода вещественного  результата  в ListBox
    Sub vivodList(ByVal Z As Double, ByVal LB As ListBox)
        LB.Items.Add(CStr(Z))
    End Sub

    ' Процедура вывода целого результата в ListBox
    Sub vivodint(ByVal Z As Integer, ByVal LB As ListBox)
        LB.Items.Add(CStr(Z))
    End Sub

    ' Процедура выч-я и вывода членов послед-ти с зад. точн.
    Private Sub Pos(ByVal x As Double, ByVal E As Double)
        Dim n As Integer
        Dim a As Double
        a = x – 1 :  n = 1
        Do While a > E
            vivodint(n, ListBox1)
            vivodList(a, ListBox2)
            a = a * (x - 1) / (n + 1)
            n = n + 1
        Loop
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles Button1.Click
```

```
        End
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles Button2.Click
        Dim EE, x0, y As Double
        EE = vvod(TextBox1)
        x0 = vvod(TextBox2)
        Pos(x0, EE)
    End Sub

End Class
```

Рис. 4.6-7

5. *Выполнение проекта и получение результатов*

Выполним проект на компьютере и получим следующий результат, приведенный на рис. 4.6-8.

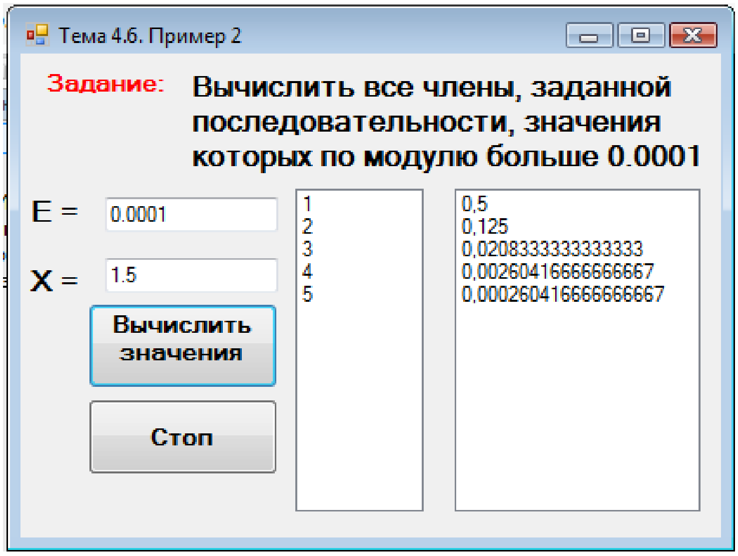

Рис. 4.6-8

6. *Доказательство правильности работы программ*

Все выведенные значения последовательности больше 0.0001.

4.6.7. Контрольные вопросы по теме «Программирование итеративных циклических структур»

1. Что представляет собой итеративный цикл?
2. Что представляет собой цикл с предусловием?
3. Что представляет собой цикл с постусловием?
4. Что представляют собой циклы Do и каковы их разновидности?
5. Что такое рекуррентная формула?
6. В чем отличие организации регулярных и итеративных циклов?
7. Какой оператор предназначен в языке **VB** для организации итеративных циклов?
8. В чем отличие использования в операторе Do конструкции While и Until?
9. Каким образом можно выйти из цикла до его завершения?
10. Как определяется число повторений операторов тела цикла в итеративной циклической структуре?
11. Какие операторы могут находиться в теле цикла оператора Do…Loop?
12. Сколько раз могут выполняться операторы тела цикла Do…Loop?
13. Какой оператор используется для программирования циклических алгоритмических структур с неизвестным числом повторений?
14. Какова алгоритмическая структура цикла итеративного типа?
15. Что за оператор Do While…Loop?
16. Что за оператор Do Until…Loop?
17. Что за оператор Do …Loop While?
18. Что за оператор Do …Loop Until?
19. Сколько раз будет выполняться тело цикла, если при программировании циклической структуры используется оператор Do While…Loop?
20. Сколько раз будет выполняться тело цикла, если при программировании циклической структуры используется оператор Do …Loop While?
21. Сколько раз будет выполняться тело цикла, если при программировании циклической структуры используется оператор Do …Loop Until?
22. Какие алгоритмы можно реализовать с использованием итеративных циклических структур?
23. Что записывается после ключевых слов While или Until в операторе итеративного цикла?

Перейти к [ЛП 4.5] [ЛП 4.7] [Огл.]