## Тема 4.5

## Программирование алгоритмов регулярных циклических структур

## и циклических структур цикл в цикле

### 4.5.1. Средства программирования регулярных циклических структур

Напомним, что ***циклическими называются структуры, в которых предусмотрена возможность многократного повторения выполнения участка алгоритма.*** Этот участок алгоритма называется **телом цикла**. Различают циклические структуры двух видов: с заранее известным и с заранее неизвестным числом повторений цикла.

Циклические структуры, в которых число повторений цикла заранее известно или может быть определено до начала выполнения цикла, называются ***регулярными*** (рис. 4.В.3-9)

В блоке организации цикла используется специальная пере­менная, которая предназначена для определения условия останова цикла – **i**. Эта переменная называется ***параметром цикла***. Блоки, следующие за заголовком цикла, составляют ***тело цикла***. Тело цикла выполняется для всех значений параметра цикла **i**, начинаю­щегося со значения **m1**и изменяющегося с шагом **h** до значения **m2**.

  Циклическая структура, в которой число повторений цикла заранее неизвестно, а определяется только в процессе выполнения алгоритма, называется ***итеративной циклической структурой****.* В зависимости от места расположения условия продолжения цикла (или выхода из цикла) ***итеративные циклические алгоритмы*** подразделяются на два вида: ***с предусловием*** и ***с постусловием***

Таким образом, алгоритмы регулярной циклической структуры характеризуются известным числом повторений: иногда это число за­дается явно, иногда его предварительно вычисляют.

К базовым ал­горитмам, использующих регулярные циклические структуры относятся: ***алгоритмы вычисления значений функций одной или нескольких переменных***, где аргументы изменяются в известном диапазоне и с заданным шагом изме­нения; алгоритмы ***вычисления конечных сумм и произведений*** и многие дру­гие. Программирование такой алгоритмической структуры на языке VBосуществляется операторам **For...Next** (Рис. 4.5.1-1).

Здесь *параметр* цикла – это переменная, для которой значение вычисленного *выражение1* является начальным значением; значение вычисленного *выражение2* – конечным значением; а значение вычисленного *выражение3* – шаг изменения параметра цикла. Причем ***шаг изменения параметра*** может быть, как положительным, так и отрицательным. По умолчанию значение шага равно **1.** *Блок Операторов*, находящихся между конструкциями **For** и **Next**, называется ***телом цикла***, количество повторений которого опре­деляется числом возможных значений параметра цикла. Перед каждым повторением цикла, в том числе первым входом в цикл, текущее значение ***параметра*** цикла сравнивается со значением *выражение2* (конечным значением). После каждого выполнения операторов ***тела цикла*** к переменной (***параметру*** цикла) прибавляется значение *выражение3* (шаг) и снова производится сравнение параметра с конечным значением. ***Тело*** цикла перестает выполняться, когда значение ***параметра*** станет больше конечного значения (шаг положителен) или меньше конеч­ного значения (шаг отрицателен). Если условие выхода из цикла выполняется при начальном значении параметра, то тело цикла пропускается, и управление сразу же передается оператору, следующему за **Next**. Для выхода из цикла до его завершения мо­жно использовать оператор **Exit For**.

|  |
| --- |
| **For** *параметр = выражение1*  **To** *выражение2*[**Step**  *выражение3*]  *Блок Операторов*  [**Exit For**]  … **Next** *параметр* |

*Рис. 4.5.1-1. Синтаксис опрератор* **For...Next**

При организации вложенных циклов (цикл в цикле) используются пары операторов **For…Next,** которые не должны «пересекаться» друг с другом. Другими словами, каждой конструкции **For** должна соответствовать конструкция **Next**. При этом конструкция **Next** для внут­реннего цикла должна предшествовать конструкции **Next** внешнего цикла.

Для преждевременного выхода из каждого цикла по условию можно использовать оператор **Exit For**.

**Пример 4.5.1-1. Примеры реализации регулярных циклических алгоритмов.**

В приведенных ниже примерах, реализующих различные варианты организации регулярных циклических алгоритмов, число повторений цикла – n может быть определено через граничные значения и шаг изменения параметра следующим обра­зом: , где m1 и m2, соответственно, нижняя и верхняя граница изменения параметра цикла, а h – шаг изменения параметра. Квадратные скобки здесь обозначают получение целой части.

В примере на рис. 4.5.1-2 тело цикла будет выполняться 10 х 2=20 раз, при условии что переменная **g** не будет равна 2, в примере на рис. 4.5.1-3 тело цикла будет выполняться 5 раз.

|  |
| --- |
| **For i = 1  To  10  For  j = 1  To 2**  *'тело внутреннего цикла*  **If  g = 2  Then  Exit  For     Next  j**  **Next i** |

*Рис. 4.5.1-2. Фрагмент использования вложенных операторов* **For...Next**

**Примера 4.5.1-1**

|  |
| --- |
| **For a = -5 To -7 Step -0.5**  **m = СStr(Sin(a))**  **MsgBox("m=" & m)**  **Next a** |

*Рис. 4.5.1-3. Фрагмент использования оператора* **For...Next** *с отрицательными параметрами* **Примера 4.5.1-1**

В примере на рис. 4.5.1-4 тело цикла будет выполняться три раза, так как шаг по умолчанию равен единице; первая итерация: **i=1, А=3**; вторая итерация: **i=2, А=6;** третья итерация: **i=3, А=18**.

|  |
| --- |
| **Dim i, A As Integer**  **A = 3**  **For i = 1 To 3**  **A = A \* i**  **Next i** |

*Рис. 4.5.1-4 Фрагмент использования оператора* **For...Next**для нахождения произведения с шагом 1 **Примера 4.5.1-1**

|  |
| --- |
| **Dim i, A As Integer**  **A = 3**  **For i = 3 To 1 Step -1**  **A = A \* i**  **Next i** |

*Рис. 4.5.1-5. Фрагмент использования оператора* **For...Next**для нахождения произведения с шагом -1 **Примера 4.5.1-1**

На рис. 4.5.1-5 тело цикла будет выполняться три раза: первая ите­рация: **i=3, А=9**; вторая итерация: **i=2, А=18;** третья итерация**: i=1, A=18**.

Изначально цикл на рис. 4.5.1-6 должен выполнить 100 итераций*,* но, поскольку при значении счетчика, равном 50, выполняется оператор **Exit For**, будет произведено всего 50 итераций.

|  |
| --- |
| **Dim i As Integer**  **For i = 1 To 100**  **If i = 50 Then Exit For**  **Next i** |

*Рис. 4.5.1-6. Фрагмент использования оператора* **For...Next**с оператором **Exit For** **Примера 4.5.1-1**

|  |
| --- |
| **Dim i As Integer**  **For i = 1 To 1000**  **Next i**  *' Пустой цикл может использоваться для за­держки времени* |

*Рис. 4.5.1-7. Пример использования оператора* **For...Next**для реализации пустого цикла **Примера 4.5.1-1**

До сих пор мы использовали элемент управления **TextBox**для ввода или вывода **одной** строки, а для вывода нескольких строк текста использовали элемент управления **ListBox*.*** Однако в элемент управления **TextBox**можно выводить и несколько строк текста. Для вывода нескольких строк в элемент управления **TextBox**надо установить свойство Multilineэлемента управления **TextBox**в значении *True*, а свойство ScrollBars– в значение *Vertical*. При использовании этих простых настроек объект однострочного текстового поля становится объектом многострочного текстового поля с полосами прокрутки, облегчающими доступ к нему.

**Пример 4.5.1-2. Создать проект, демонстрирующий реализацию регулярных циклических структур.**

1. Запустить VS и создать новый проект с именем ***Пример 5.1-2****,* а на появившейся форме двойным щелчком мышью создать элемент управления **Button** в верхний левый угол формы. В случае с элементом управления **Button***,* а также и со многими други­ми, двойной щелчок мышью является быстрым способом создания на форме объекта со стандартными размерами. Теперь этот объект перетащить туда, куда его требуется поместить, и настроить его свойства. перетащить туда, куда его требуется поместить, и настроить его свойства.
2. Перетащить объект кнопки вправо и поместить его в центре верхней части формы.
3. Открыть окно **Свойства**, а затем установить свойство Textкнопки в значении *Цикл.*
4. Дважды щелкнуть мышью в окне **Элементы управления** на элементе управления **TextBox***.* VS создаст на форме небольшой объект текстового поля.
5. Установить свойство Multiline этого объекта текстового поля в значении *True*, а затем установить его свойство ScrollBars в значении *Vertical.* Эти настройки подготавливают текстовое поле к отображению не­скольких строк.
6. Переместить текстовое поле так, чтобы оно располагалось ниже кноп­ки, и увеличить его так, чтобы оно заняло примерно треть формы.
7. Дважды щелкнуть мышью на форме.

Появится окно **Редактор кода**.

1. В это окно добавить процедуру **Cykl()** и ее вызов из процедуры события **Button1\_Click()**, приведенные на рис. 4.5.1-8.

|  |
| --- |
| **Public Class Form1**  **Sub Cykl(ByVal Line As String)**  **Dim i As Integer**  **For i = 1 To 10**  **TextBox1.Text = TextBox1.Text & Line & i & vbCrLf**  **Next i**  **End Sub**  **Private Sub Button1\_Click(ByVal sender As System.Object, \_**  **ByVal e As System.EventArgs) Handles Button1.Click**  **Dim Line As String = "Line "**  **Cykl(Line)**  **End Sub**  **End Class** |

*Рис. 4.5.1-8. Программный код проекта* ***Пример 5.1-2***

1. После объявления переменной, используется цикл **For...Next**, который десять раз отображает в объекте текстового поля строку   
   **Line i**, где **i** – текущее значение переменной-счетчика (другими словами, от **Line 1** до **Line 10**). Символы конкатенации строк (**&**) в текстовом поле объединяют воедино части каждой строки. Вна­чале в объект добавляется все значение текстового поля, которое хра­нится в свойстве Text, так, что предыдущие строки при добавлении новых строк не удаляются. Затем для отображения новой строки и перево­да курсора к левой границе и на следующую строку, объединяются строка **Line**, текущий номер строки и символ перевода строки **vbCrLf**. Оператор **Next** завершает цикл.
2. Обратить внимание, что при вводе начала цикла **For** VS ав­томатически добавляет в конец цикла оператор **Next**.
3. Чтобы сохранить изменения, необходимо щелкнуть на кнопке   
    Сохранить все на стан­дартной панели инструментов.
4. Щелкнуть на кнопке *Начать отладку* на панели инстру­ментов.
5. Щелкнуть на кнопке ***Цикл*** формы.

Цикл **For...Next** отобразит в текстовом поле 10 строк, как показано на рис. 4.5.1-9.



*Рис. 4.5.1-9. Результат работы проекта* ***Пример 5.1-2***

1. Снова щелкнуть на кнопке ***Цикл*** формы.
2. Цикл **For...Next** отобразит на форме еще 10 строк. Чтобы увидеть все строки, можно воспользоваться вертикальной полосой про­крутки. При каждом повторении цикла в объект тек­стового поля добавляется 10 новых строк.
3. Необходимо обратить внимание, что в объекте текстового поля не останется места. Даже если отображается всего не­сколько строк текста, на это потребуется время. Многостроч­ный объект имеет практическое ограничение на размер текста в 32 Кб. Если требуется вывести текст большего размера или получить дополнительные опции форматирования, используй­те элемент управления **RichTextBox** из окна **Элементы управления** – похожий, но более мощный элемент управления для отображения и обработ­ки текста.

**Пример 4.5.1-3. Написать процедуры ввода/вывода, которые могут использоваться в алгоритмах циклических структур.**

**vvodDbl1( )** и **vivodDbl1() *–*** процедуры ввода и вывода данных типа **Double** в текстовые поля,

**vvodSng3( )** и **vivodSng3()** ***–*** процедуры ввода и вывода данных типа **Single** в текстовые поля,

**vvodInt4( )** и **vivodInt4()** ***–*** процедуры ввода и вывода данных типа **Integer** в текстовые поля,

**vvodLng5()** и **vivodLng5()** ***–*** процедуры ввода и вывода данных типа **Long** в текстовые поля,

**vivodDblFx6()** и **vivodLngFx7()** ***–*** процедуры форматного вывода значения одной переменной типа **Double** и **Long**,

**vivodDblFxy8()** ***–*** процедуры форматного вывода значения двух переменных типа **Double**.

Процедуры **vvodDbl1()**, **vivodDbl1( )**, **vvodSng3()**, **vivodSng3(), vvodIDbl2()** и **vivodDbl2()** приведены в **Теме 4.3**, процедуры **vvodInt4()** и **vivodInt4()** приведены в **Теме 4.4**, а процедуры **vvodLng5(),  
vivodLng5(), vivodDblFx6(), vivodLngFx7(), vivodSngFxy8(),   
vvodInt9(),vivodInt9()** представлены на рис. 4.5.1-10 – 4.5.1-16.

|  |
| --- |
| *'процедура ввода данных типа Long из текстового поля TextBox*  **Function vvodLng5(ByVal T As TextBox) As Long**  **Return CLng(T.Text)**  **End Function** |

*Рис. 4.5.1-10. Программный код процедуры* **vvodLng5()** *ввода данных типа* **Long** *из текстового поля* **TextBox Примера 4.5.1-3**

|  |
| --- |
| *'Процедура вывода данных типа Long в TextBox*  **Sub vivodLng5(ByVal l As Long, ByVal T As TextBox)**  **T.Text = CStr(l)**  **End Sub** |

*Рис. 4.5.1-11. Программный код процедуры* **vivodLng5()** *вывода данных типа* **Long** *в* **TextBox Примера 4.5.1-3**

|  |
| --- |
| *'Процедура форматного вывода данных типа Double в TextBox*  **Sub vivodDblFx6(ByVal x As Double, ByVal T As TextBox)**  **T.Text = T.Text + Format(x, "0.000 ")**  **End Sub** |

*Рис. 4.5.1-12. Программный код процедуры* **vivodDblFx6()** *форматного вывода данных типа* **Double** *в* **TextBox Примера 4.5.1-3**

|  |
| --- |
| *'Процедура форматного вывода данных типа Long в TextBox*  **Sub vivodLngFx7(ByVal x As Long, ByVal T As TextBox)**  **T.Text = T.Text + Format(x, "0")**  **End Sub** |

*Рис. 4.5.1-13. Программный код процедуры* **vivodLngFx7()** *форматного вывода данных типа* **Long** *в* **TextBox Примера 4.5.1-3**

|  |
| --- |
| *'Процедура форматного вывода значений 2 переменных Single в TextBox*  **Sub vivodSngFxy8(ByVal x As Single,ByVal y As Single, \_**  **ByVal T As TextBox)**  **T.Text = T.Text + Format(x, "0.000 ") + Space(8) + \_**  **Format(y, "0.000 ") + vbCrLf**  **End Sub** |

*Рис. 4.5.1-14. Программный код процедуры* **vivodSngFxy8()**

*Форматного вывода значений 2 переменных типа* **Single** *в* **TextBox Примера 4.5.1-3**

|  |
| --- |
| *'процедура-Function ввода данных типа Integer из функции InputBox()*  **Function vvodInt9(ByVal p As String, ByVal T As TextBox) \_**  **As Integer**  **T.Text = InputBox(P)**  **Return CInt(T.Text)**  **End Function** |

*Рис. 4.5.1-15. Программный код процедуры* **vvodInt9()** *ввода данных типа* **Integer** *из функции* **InputBox()   
Примера 4.5.1-3**

|  |
| --- |
| *'Процедура вывода данных типа Integer функцией MsgBox()*  **Sub vivodInt9(ByVal I As Integer, \_**  **ByVal S As String, ByVal T As TextBox)**  **T.Text = MsgBox(CStr(I), S)**  **End Sub** |

*Рис. 4.5.1-16. Программный код процедуры* **vivodInt9()**

*вывода данных типа* **Integer** *функцией* **MsgBox()**

**Примера 4.5.1-3**

В вышеприведенных примерах для организации вывода данных, используется форматированный вывод данных с использованием функции **Format()**, описание которой приведено в табл.4.2.4-2.

### Базовые алгоритмы

### регулярных циклических структур и примеры их программирования

***Алгоритмы регулярной циклической структуры*** позволяют описать широкий класс задач, из которых можно выделить следующие:

* Ввод и вывод членов последовательности (Пример 4.5.2-3).
* Вычисление значений функций одной переменной с заданным диапазоном и шагом их изменения при заданных условиях (Пример 4.5.2-2).
* Вычисление значений функций двух переменных с заданными диапазонами и шагами их изменения при заданных условиях с заданными критериями (Пример 4.5.2-8).
* Вычисление конечных сумм и произведений значений функции n-го членов последовательности при заданных условиях (Примеры 4.5.2-3 – 4.5.2-4, 4.5.2-6).
* Вычисление n-го члена последовательности с заданными критериями (Пример 4.5.2-5).
* Определе­ние наибольшего (наименьшего) значения функции одной и более пере­менных с заданными критериями (Пример 4.5.2-7);

Рассмотрим последовательность данных **a0, a1,…,ai,…,an**.

В одном случае член последовательности **ai**можно определить, вычисляя выражение **ai**.

В других случаях член последовательности **ai**можно определить через k предыдущих членов той же последовательности: **ai = f(a i-k, a i-k+1,… a i-1), для k ≤ i.** То есть, для нахождения **i**-го члена последовательности должны быть заданы члены **a0, a1, …, ai-1**. Такое определение называется ***ите­рацией*** или ***рекуррентной формулой,*** а алгоритмы вычисления произвольного члена последовательно­сти – ***итерационными.*** Как пра­вило, итерационные алгоритмы реализуются с помощью циклов – как ре­гулярных, так и итеративных.

В простейшем случае используется выражение **ai=f(ai-1)** для i**>0, a0=c**, где **с** – известное значение нулевого члена последовательности. Например, для арифметической прогрессии **ai=ai-1+d, a0=a**, а для геометрической – **bi=bi-1 ∙q**, **b0=b**. При программировании итераций такого типа используется выражение **a=f(a)**. Использование этого выражения основано на ***специфике операции присваивания***. Сначала вычисляется правая часть выражения при предыдущем значении **a,** равном **ai-1**, а затем **a** из­меняется на вычисленное значение, тем самым получается текущее значение **a,** равное **ai**.

С помощью рекуррентных формул удобно находить суммы и про­изведения членов последовательности. Обозначим Sk – сумму **k** членов последовательности **a1,… ai,… an**. Очевидно, что **Sk=S k-1 + ak**для **k=1,…, n**; начальное значение суммы следует обнулить - **S0=0**. Аналогично произведение **k** членов этой последовательности **Pk=Pk-1 ∙ak**, для **k=1,…,n**; начальное значение произведения надо установить равным единице, **P0=1**.

Заметим, что с использованием регулярных циклических структур можно находить суммы с конечным числом слагаемых и произведение с конечным, т.е. с заранее заданным числом сомножителей. Вычисление суммы и произведения неизвестного заранее числа членов последовательности осуществляется с помощью итеративных циклов и будет рассмотрено в следующей теме.

В регулярных циклах число повторений должно быть определено заранее. При этом, в одних случаях это число задано явно (константой или вводимым значением переменной), а в других случаях его надо предварительно вычислить.

Наконец, возможны и такие случаи, когда число повторений цикла не фиксируется в алгоритме в явном виде, а определяется неявно граничными значениями и шагом изменения не­которых переменных.

**Пример 4.5.2-1. Написать процедуру-Sub, которая вычисляет значение функции y(x) = sin(x) при значе­ниях x, изменяющихся на отрезке [a; b] с шагом h.**

Другой формулировкой данной задачи может быть следующая.

**Написать процедуру-Sub, которая формирует таблицу значений функции y(x)= sin(x) при изме­нении x от a до b с шагом h. Значения a, b, h – вводимые величины**.

|  |  |
| --- | --- |
|  | **Sub Pr521()**  **Dim a, b, h As Single**  **Dim x, y As Single**  **a = vvodSng3(TextBox1)**  **b = vvodSng3(TextBox2)**  **h = vvodSng3(TextBox3)**  **For x = a To b Step h**  **y = Sin(x)**  **vivodSng3Fxy8(x, y, TextBox4)**  **Next x**  **EndSub** |

*Рис. 4.5.2-1. Схема алгоритма и программный код процедуры* **Pr521(),** в которойиспользуется в качестве параметра цикла переменная вещественного типа с вводом/выводом **Примера 4.5.2-1**

Схема и код программы представлены на рис. 4.5.2-1.

Процедура **Pr521()** может быть вызвана, например, как показано на   
рис. 4.5.2-2.

|  |
| --- |
| **Pr521( )** |

*Рис. 4.5.2-2. Пример вызова процедуры* **Pr521()**

Если процедура не будет содержать ввода данных, то она будет выглядеть, как показано на рис. рис. 4.5.2-3.

|  |  |
| --- | --- |
|  | **Sub Pr523(ByVal a As Single, \_**  **ByVal b As Single, \_**  **ByVal h As Single)**  **Dim x As Single, y As Single**  **For x = a To b Step h**  **y = Sin(x)**  **vivodSng3Fxy8(x, y, TextBox4)**  **Next x**  **End Sub** |

*Рис. 4.5.2-3. Схема алгоритма и программный код процедуры* **Pr523(),**

в которойиспользуется в качестве параметра цикла переменная вещественного типа выводом данных **Примера 4.5.2-1**

Процедура вызова **Pr523()** может быть, например, как показано на   
рис. 4.5.2-4.

|  |
| --- |
| **Dim aa, bb, hh As Single**  **aa=vvodSng3(TextBox1)**  **bb=vvodSng3(TextBox2)**  **hh=vvodSng3(TextBox3)**  **Pr523(aa, bb, hh)** |

*Рис. 4.5.2-4. . Пример вызова процедуры* **Pr523()**

В алгоритме решения данной задачи используется регулярная циклическая структура, которая в программе реализована оператором **For …Next**, а в качестве параметра цикла используется вещественный аргумент функции – **х**.

Такой способ организации цикла опасен тем, что в результате погрешностей округления последнее зна­чение параметра может оказаться не равным в точности значению верхней границы **b**, и если оно окажется больше **b,** то выполнение цикла прекратится, a в результате таблица значений функции будет содержать на одну точку меньше. Так, например, если **a=0, b=1** и **h=0.1**, то последнее, выведенное в цикле значение **x**, оказывается равным 0.9000001, т.е. точка **x=b** в таблицу не попадет. Такой способ организации цикла в данной задаче можно использовать только, если переменные **a, b, h** и **x** целого типа.

|  |  |
| --- | --- |
|  | **Sub Pr525(ByVal a As Single, \_**  **ByVal b As Single, \_**  **ByVal h As Single)**  **Dim x, y As Single**  **Dim n, i As Integer**  **n = CInt((b - a) / h) + 1**  **x = a**  **For i = 1 To n**  **y= Sin(x)**  **x = x + h**  **vivodSngFxy8(x, y, TextBox4)**  **Next i**  **End Sub** |

*Рис. 4.5.2-5. Схема алгоритма и программный код процедуры* **Pr525(),**

*в которой**используется в качестве параметра цикла переменная целого типа, а очередное значение* ***х*** *вычисляется через предыдущее путем добавления шага* **Примера 4.5.1-3**

Избежать подобных неприятностей можно путем предварительного вычисления числа повторений цикла и использования в качестве параметра цикла ***переменной целого типа***, назначение которого – подсчет количества повторений цикла (рис. 4.5.2-5 и рис. 4.5.2-6). Различие между алгоритмами состоит только в способах вычисления текущего значения аргумента **x**. В первом алгоритме очередное значение **х** вычисляется через предыдущее путем добавления шага, а во втором алгоритме х вычисляется через параметр цикла.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **Sub Pr526(ByVal a As Single, \_**  **ByVal b As Single, \_**  **ByVal h As Single)**  **Dim x, y As Single**  **Dim n, i As Integer**  **n = CInt((b - a) / h) + 1**  **For i = 1 To n**  **x = a + (i - 1) \* h**  **y = Sin(x)**  **vivodSngFxy8(x, y, TextBox4)**  **Next i**  **End Sub** | |

*Рис. 4.5.2-6. Схема алгоритма и программный код процедуры* **Pr526(),**

*в которой**используется в качестве параметра цикла переменная целого типа, а очередное значение х вычисляется через предыдущее путем добавления шага* **Примера 4.5.1-3**

***Обратите внимание на то, что при большом диапазоне (в интервале* [a; b] *переменная* a *имеет маленькое значение, а переменная* b *большое) c маленьким шагом* h, *вычисление по формуле* x=x+h *дает большую погрешность, чем формула* x=a+(i-1)\*h *(даже при использовании данных типа* Double)*.***

**Пример 4.5.2-2. Написать процедуру-функцию, которая вычисляет сумму положительных значений функции y=cos(8x) при изменении x от 0 до π с шагом π/16.**

Алгоритм решения данной задачи относится к алгоритмам вычисления конечных сумм (рис.4.5.2-7).

Процедура-функция Pr**527()** может быть вызвана из любой другой процедуры или из модуля формы, например, как показано на рис. 4.5.2-8.

Алгоритм использует цикл с параметром целого типа. Для уменьшения влияния ошибок округления при таком спо­собе вычисления текущего значения аргумента используется представление вещественных чисел с удвоенной точностью. Количество повторений цикла можно вычислить заранее по вышеприведенной формуле или непосредственно программой перед входом в цикл

.

|  |  |
| --- | --- |
|  | **Function Pr527(ByVal n As Integer,\_**  **ByVal h As Double) As Double**  **Dim x, y, s As Double**  **Dim i As Integer**  **s = 0**  **x = 0**  **For  i = 1 To n**  **Y = Cos(8 \* x)  If y > 0 Then s = s + y  x = x + h  Next i**  **Return s**  **End Function** |

*Рис.4.5.2-7. Схема алгоритма и программный код процедуры* **Pr527() Примера 4.5.2-2**

|  |
| --- |
| **Dim ss, hh As Double**  **Dim nn As Integer**  **h = Math.PI / 16**  **n = CInt((Math.PI -0) / h) + 1** *'nn=17*  **ss = Pr527(nn, hh)**  **vivodDblFx6(ss, TextBox1)** |

*Рис. 4.5.2-8. Пример вызова процедуры* **Pr527()**

Алгоритм решения данной задачи относится к алгоритмам вычисления конечных сумм (рис.4.5.2-7).

Процедура-функция Pr**527()** может быть вызвана из любой другой процедуры или из модуля формы, например, как показано на рис. 4.5.2-8.

Алгоритм использует цикл с параметром целого типа. Для уменьшения влияния ошибок округления при таком спо­собе вычисления текущего значения аргумента используется представление вещественных чисел с удвоенной точностью. Количество повторений цикла можно вычислить заранее по вышеприведенной формуле или непосредственно программой перед входом в цикл

.

Так как при таком способе организации цикла (через целый параметр), аргумент **x** уже не будет автоматически изменяться при каждом повторении цикла, то в тело цикла добавляется оператор **x=x+h**, который обеспечивает изменение **x** на величину шага **h** при каждом повторении цикла. При этом начальное значение аргумента **x** должно быть установлено перед первым входом в цикл (**x=0**). Начальное значение суммы также следует установить равным нулю до цикла (**s=0**), чтобы после нахождения первого подходящего значения **y>0** значение суммы стало равно этому значению. Заметим, что если в задаче требуется вычислить конечное произведение, то начальное значение переменной, которая будет накапливать произведение, следует установить равным единице (**p=1**).

**Пример 4.5.2-3. Написать процедуру-Sub, которая вычисляет сумму и произведение n членов последовательности по формуле   
fi = -i ∙ fi-1,** а **f0 = 1**.

Схема алгоритма и код программы приведены на рис. 4.5.2-9.

|  |  |
| --- | --- |
|  | **Sub Pr529(ByVal n As Integer, \_**  **ByRef s As Long, \_**  **ByRef p As Long)**  **Dim f As Long**  **Dim I As Integer**  **f = 1 : s = 0 : p = 1**  **For i = 2 To n**  **f = -i \* f**  **s = s + f**  **p = p \* f**  **Next i**  **End Sub** |

*Рис. 4.5.2-9. Схема алгоритма и программный код процедуры* **Pr529() Примера 4.5.2-3**

Процедура Pr**529()** может быть вызвана из любой другой процедуры или из модуля формы, например, как показано на рис. 4.5.2-10.

|  |
| --- |
| **Dim ss, pp As Long**  **Dim nn As Integer**  **nn= vvodInt4(TextBox1)**  **Pr529(nn, ss, pp)**  **vivodLngFx7(ss, TextBox2)**  **vivodLngFx7(pp, TextBox3)** |

*Рис. 4.5.2-10. Пример вызова процедуры* **Pr529()**

**Пример 4.5.2-4. Написать процедуру-Function, которая вычисляет   
n-й член последовательности по формуле fi= fi-1 + fi-2, а f0 = f1 = 1.**

Это задача вычисления чисел Фибоначчи, которые определяются по заданной формуле. В данном примере каждый очередной член последовательности зависит от двух предыдущих. Обозначим **b=f0, с=f1.** Тогда очередной член последовательности **a=b+c**. Далее операторы **b=c** и **c=a** изменяют значения **b** и **c** для очередного вычисления **a**. Таким образом, значение предыдущего члена последовательности присваивается предшествующему ему члену последовательности (**b=c**), а значение только что вычисленного члена **a** присваиваем предыдущему (**c=a**)**.** Начальное значение параметра цикла **i=3**, так как вычисление начинается с 3-го члена (**f0=f1=1** задано).

Схема алгоритма и код программы приведены на рис. 4.5.2-11.

|  |  |
| --- | --- |
|  | **Function Pr5211(ByVal n As Integer)**  **As Integer**  **Dim a, b, c As Integer**  **Dim i As Integer**  **b = 1**  **c = 1**  **For i = 3 To n**  **a = b + c**  **b = c**  **c = a**  **Next i**  **Return a**  **End Func*t*ion** |

*Рис. 4.5.2-11. Схема алгоритма и программный код процедуры* **Pr5211() Примера 4.5.2-4**

Процедура – функция **Pr5211()** может быть вызвана из любой другой процедуры или из модуля формы, например, как показано на рис. 4.5.2-12.

|  |
| --- |
| **Dim aa As Integer**  **Dim nn As Integer**  **nn = vvodInt4(TextBox1)**  **aa = Pr5211(nn)**  **vivodInt4(aa, TextBox2)** |

*Рис. 4.5.1-12. Пример вызова процедуры* **Pr5211()**

**Пример 4.5.2-4. Написать процедуру-Function, которая вычисляет значение факториала заданного числа n.**

По определению **f=n!=1 ∙ 2 ∙ …∙ (n-1) ∙ n** (при **n≥1**) и **f=0!=1**(при **n=0**).

Таким образом, вычисление факториала сводится к задаче вычисления произведения значений целых чисел, изменяющихся от 1 до **n** с шагом 1.

Накопление произведения будем производить по рекуррентной формуле: **fi = fi-1 ∙ i; ; f0 = 1**.

Схема алгоритма и код программы приведены на рис. 4.5.2-13

|  |  |
| --- | --- |
|  | **Function Pr52-13(ByVal n As Integer) As Long**  **Dim f As Long**  **Dim i As Integer**  **f = 1**  **For i = 2 To n**  **f = f \* i**  **Next i**  **Return f**  **End Function** |

*Рис. 4.5.2-13. Схема алгоритма и программный код процедуры* **Pr5213() Примера 4.5.2-4**

|  |
| --- |
| **Dim nn As Integer**  **Dim ff As Long**  **Nn = vvodInt4(TextBox1)**  **ff = Pr5213(nn)**  **vivodLngFx7(ff, TextBox2)** |

*Рис. 4.5.2-14. Пример вызова процедуры* **Pr5213()**

Процедура – функция **Pr5213()** может быть вызвана, например, как показано на рис. 4.5.2-14.

**Пример 4.5.2-5. Написать процедуру-подпрограмму, которая находит наименьшее значение функции y=c∙e-dx∙sin(2x+5), где c, d – входные параметры, а x изменяется на отрезке [a; b] с шагом h.**

Схема алгоритма и код программы приведены на рис. 4.5.2-15.

Алгоритм нахождения наименьшего (наибольшего) значения функции основан на последовательном сравнении очередного значения функции с текущим наименьшим (наибольшим) значением функции. В качестве начального наименьшего значения **ymin** следует использовать число, близкое к наибольшему числу из диапазона возможных значений того типа данных, который описывает рассматриваемую функцию, чтобы наверняка выполнилось условие **y < ymin** при первом проходе цикла. То есть, за **ymin** можно принять число**,** близкое к наибольшему из диапазона значений данных типа **Doublе**, т.е **Double.MaxValue**.

|  |  |
| --- | --- |
|  | **Sub Pr5215(ByVal n As Integer, \_**  **ByVal a As Double, \_**  **ByVal c As Double, \_**  **ByVal d As Double, \_**  **ByVal h As Double, \_**  **ByRef xmin As Double, \_**  **ByRef ymin As Double)**  **Dim i As Integer**  **Dim x, y As Double**  **ymin = Double.MaxValue**  **x=a**  **For i =1 To n**  **y =c\*Exp(-d\*x)\*Sin(2\*x+5)**  **If y<ymin Then**  **ymin=y : xmin=x**  **End If**  **x =x+h**  **Next i**  **End Sub** |

*Рис. 4.5.2-15. Схема алгоритма и программный код процедуры* **Pr5215() Примера 4.5.2-5**

Соответственно, за начальное значение **ymax** можно принять число **Double.MinValue**, т.е. число, близкое к наименьшему числу из диапазона значений **Double**, для того чтобы при первом проходе цикла гарантированно выполнялось условие **y > ymax**. Для определения значения аргумента **xmin (xmax)**, соответствующего найденному наименьшему (наибольшему) значению функции при каждом переопределении текущего наименьшего (наибольшего) значения функции надо одновременно переопределять текущее значение **xmin (xmax)**.

Процедура – функция **Pr5215()** может быть вызвана из любой другой процедуры или из модуля формы, например, как показано на рис. 4.5.2-16.

|  |
| --- |
| **Dim nn As Integer**  **Dim aa, bb, hh, cc, dd, ym, xm As Double**  **aa = vvodDbl2("Введите значение a=",TextBox1)**  **bb = vvodDbl2("Введите значение b=",TextBox2)**  **hh = vvodDbl2("Введите значение h=",TextBox3)**  **cc = vvodDbl2("Введите значение c=",TextBox4)**  **dd = vvodDbl2("Введите значение d=",TextBox5)**  **nn = CInt((bb-aa)/hh)+1**  **Pr5215(nn, aa, cc, dd, hh,xm, ym)**  **vivodDblFx6(xm, TextBox6)**  **vivodDblFx6(ym, TextBox7)** |

*Рис. 4.5.2-16. Пример вызова процедуры* **Pr5215()**

**Пример 4.5.2-7. Получить таблицу значений функции z(x,y)=x2+y2 в заданных диапазонах значений аргументов , если шаг изменения x равен hx, а шаг изменения y равен hy.**

Схема алгоритма и код программы приведены на рис. 4.5.2-17.

|  |  |
| --- | --- |
|  | **Sub Pr5217(ByVal n As Integer, \_**  **ByVal m As Integer, \_**  **ByVal a As Double, \_**  **ByVal hx As Double, \_**  **ByVal c As Double, \_**  **ByVal hy As Double)**  **Dim x, y, z As Double**  **Dim j, i As Integer**  **x = a**  **For i = 1 To n**  **y =c**  **For j = 1 To m**  **z = x^2 + y^2**  **vivodDblFx6(x, TextBox7)**  **vivodDblFx6(y, TextBox8)**  **vivodDblFx6(z, TextBox9)**  **y = y + hy**  **Next j**  **x = x + hx**  **Next i**  **End Sub** |

*Рис. 4.5.2-17. Схема алгоритма и программный код процедуры* **Pr5217() Примера 4.5.2-7**

Процедура **Pr5217()** может быть вызвана из любой другой процедуры или из модуля формы, например, как показано на рис. 4.5.2-18.

|  |
| --- |
| **Dim nn, mm As Integer**  **Dim a, b, hx, c, d, hy As Double**  **a = vvodDbl2("Введите значение a", TextBox1)**  **b = vvodDbl2("Введите значение b", TextBox2)**  **hx = vvodDbl2("Введите значение hx",TextBox3)**  **c = vvodDbl2("Введите значение c", TextBox4)**  **d = vvodDbl2("Введите значение d", TextBox5)**  **hy = vvodDbl2("Введите значение hy",TextBox6)**  **nn = CInt((b-a )/ hx) + 1**  **mm = CInt((d-c) / hy) + 1**  **Sub Pr5217(nn, mm, a, hx, c, hy)** |

*Рис. 4.5.2-18. Пример вызова процедуры* **Pr5217()**

**Пример 4.5.2-8. Создать проект *Пример 4.5.2-8*, который выводит в текстовое окно значения функции y=+cosx3 на отрезке [a; b] с шагом h.**

Форма должна содержать три элемента **Label** для записи подсказок вводимых переменных и три текстовых поля **TextBox** для ввода значений этих переменных. На форме должно быть также текстовое поле для вывода значений функции и аргумента. Для этого текстового поля необходимо задать свойство MultiLine, равное ***True***, а свойство ScrollBars– в значении *Vertical* для возможности прокрутки текстового поля по вертикали. Помимо этого, на форме должна быть кнопка **Button1**, с которой необходимо связать программный код.

Форма должна быть такой, как на рис. 4.5.2-19.

Результаты работы проекта представлены на рис. 4.5.2-19.

Программный код проекта приведен рис. 4.5.2-20.

****

*Рис. 4.5.2-19. Форма проекта* ***Пример 45.2-8*** с результатами

|  |
| --- |
| **Option Explicit On**  **Option Strict On**  **Imports System.Math**  **Public Class Form1**  **Function vvodDbl1(ByVal T As TextBox) As Double**  **Return Val(T.Text)**  **End Function**  **Sub vivodDblFxy8(ByVal x As Double, ByVal y As Double, \_**  **ByVal T As TextBox)**  **T.Text = T.Text + Format(x, "0.000 ") + Space(8) + \_**  **Format(y, "0.000 ") + vbCrLf**  **End Sub**  **Sub TabFun(ByVal a As Double, ByVal b As Double, \_**  **ByVal h As Double)**  **Dim x, y As Double**  **Dim i, n As Integer**  **x = a**  **n = CInt(((b - a) / h) + 1)**  **For i = 1 To n**  **y = Sqrt(x + 5) + Cos(x ^ 3)**  **vivodDblFxy8(x, y, TextBox4)**  **x = x + h**  **Next i**  **End Sub**  **Private Sub Button1\_Click(ByVal sender As System.Object,\_**  **ByVal e As System.EventArgs) Handles Button1.Click**  **Dim aa, bb, hh As Double**  **aa = vvodDbl1(TextBox1) : bb = vvodDbl1(TextBox2)**  **hh = vvodDbl1(TextBox3) : TabFun(aa, bb, hh)**  **End Sub**  **Private Sub Button2\_Click(ByVal sender As System.Object,\_**  **ByVal e As System.EventArgs) Handles Button2.Click**  **End**  **End Sub**  **End Class** |

*Рис. 4.5.2-20. Программный код процедуры проекта* **Примера 5.2-8**

### 4.5.3. Тестовые задания

1. **Оператор For – это:**
2. оператор выбора;
3. оператор регулярного цикла
4. оператор итеративного цикла
5. составной оператор
6. **Переменная, изменяющая свое значение при каждом новом вхождении в цикл,**

**называется**

1. параметром цикла
2. шагом цикла
3. индексом
4. размером
5. **Многократно повторяющаяся часть алгоритма называется**
6. телом цикла
7. выбором
8. перебором
9. шагом цикла
10. **Циклическая алгоритмическая структура может быть**
11. регулярная, итеративная
12. регулярная, разветвляющаяся
13. итеративная, разветвляющаяся
14. нет верного ответа
15. **В регулярной циклической структуре число повторений операторов тела цикла**
16. заранее известно или может быть предварительно вычислено
17. может быть, известно или неизвестно заранее
18. заранее неизвестно
19. нет верного ответа
20. **Тело цикла в операторе For…Next**
21. может ни разу не выполниться
22. обязательно выполнится хотя бы 1 раз
23. выполняется несколько раз
24. **Для досрочного прекращения регулярного цикла используется оператор**
25. **ExitFor**
26. **Exit**
27. **Break**
28. **Exit For**
29. **Для того чтобы операторы тела цикла выполнились необходимое число раз, параметр цикла**
30. должен быть переменной вещественного типа
31. должен быть переменной целого типа
32. должен быть переменной строкового типа
33. должен быть переменной целого или вещественного типа
34. **Если в операторе регулярного цикла For…Next слово Step отсутствует, это означает, что после каждого выполнения тела цикла**
35. параметр цикла увеличивается на 1
36. параметр цикла уменьшается на 1
37. параметр цикла изменяется произвольно
38. параметр цикла не изменяется
39. **Для нахождения наименьшего значения числовой функции от аргумента, значение которого изменяется на заданном интервале, за начальное наименьшее значение следует принять**
40. число, близкое к наибольшему числу из диапазона типа данных рассматриваемой функции
41. число, близкое к наименьшему числу из диапазона типа данных рассматриваемой функции
42. 0
43. нет верного ответа
44. **Для нахождения наибольшего значения числовой функции от аргумента, значение которого изменяется на заданном интервале, за начальное наибольшее значение следует принять**
45. число, близкое к наибольшему числу из диапазона типа данных рассматриваемой функции
46. 0
47. нет верного ответа
48. число, близкое к наименьшему числу из диапазона типа данных рассматриваемой функции
49. **Для нахождения значения произведения числовой функции от аргумента, значение которого изменяется на заданном интервале, за начальное значение произведения следует принять**
50. 1
51. число, близкое к наибольшему числу из диапазона типа данных рассматриваемой функции
52. число, близкое к наименьшему числу из диапазона типа данных рассматриваемой функции
53. 0
54. нет верного ответа
55. **Для нахождения значения суммы числовой функции от аргумента, значение которого изменяется на заданном интервале, за начальное значение суммы следует принять**
56. число, близкое к наибольшему числу из диапазона типа данных рассматривае

мой функции

1. число, близкое к наименьшему числу из диапазона типа данных рассматривае-

мой функции

1. 0
2. 1
3. нет верного ответа
4. **Значение переменной M после выполнения фрагмента программы**

|  |
| --- |
| **M = 0**  **For N = 1 To 3**  **For K = 1 To 2**  **M = M + 1**  **Next**  **Next N** |

**будет равно**

1. 6
2. 13
3. 12
4. нет верного ответа

### 4.5.4. Лабораторная работа по теме *«Программирование алгоритмов регулярных циклических структур и циклических структур цикл в цикле»*

**Цель** данной работы состоит в получении навыков разработки проектов, использующих алгоритмы регулярных циклических структур, структур цикл в цикле и освоении стандартного элемента управления **ListBox**.

### Вопросы, подлежащие изучению

1. Циклические алгоритмические структуры.
2. Операторы для программирования регулярных циклов.
3. Базовые алгоритмы регулярных циклических структур.
4. Базовые алгоритмы циклических структур цикл в цикле.
5. Элемент управления **ListBox**. Добавление, вставка и удаление элементов списка.

### Общее задание на разработку двух проектов

1. ***Изучите вопросы программирования алгоритмов регулярных циклических структур и циклических структур цикл в цикле*** *(****Тема 5****).*
2. **Создайте приложение**, состоящее из двух проектов. Первый проект для построения таблицы значений функции y=f(x) одного аргумента, а второй проект для построения таблицы значений функции z=f(x,y) двух аргументов
3. **Выберите вариант задания** из табл. 4.4.5-1 для создания 1-го проекта и из табл. 4.4.5-2 для создания 2-го проекта.
4. ***Проведите формализацию*** *поставленной задачи*.
5. ***Составьте схему алгоритма*** решения поставленных задач.
6. ***Разработайте интерфейсы*** пользователя
7. ***Напишите программный код*** *процедур пользователя для двух проектов в соответствии со схемами алгоритмов. Использовать в качестве параметра цикла переменную целого типа. Обмен данными между процедурами должен осуществляться через параметры, без использования глобальных переменных.*
8. ***Напишите программный код*** двух проектов. ***Событийная процедура*** должна содержать только операторы вызова пользовательских (общих) процедур.
9. ***Подготовте тесты*** *для контрольного решения задачи.*
10. ***Выполните созданные проекты***.
11. ***Получите результаты*** *выполнения проектов.*
12. ***Обоснуйте правильность*** *полученных результатов на заранее*

*разработанных тестах для всех ветвей программы*.

### Варианты индивидуальных заданий

Таблица 4.4.5-1

|  |  |
| --- | --- |
| **№** | **Задача** |
| **1)** | Постройте таблицу значений функции **y=f(x)** при изменении **x** на отрезке [a;b] c шагом h. |
| **2)** | Постройте таблицу значений функции **y=f(x)** и найдите наибольшее значение функции **y=f(x)** при изменении **x** на отрезке **[a;b]**c шагом **h**.  **Y=3cos(2x+1).** Отрезок [-], шаг h=. |
| **3)** | Постройте таблицу и найдите наибольшее значение функции **y=f(x)** при изменении **x** на отрезке **[a;b]**c шагом h.  **Y=2.** Отрезок**[-1,1],** шаг **h=0.1** |
| **4)** | Постройте таблицу и вычислите сумму значений функции **y=f(x)** при **y>0** при изменении **x** на отрезке **[a;b]** с шагом **h**.  Отрезок [-], шаг **h=** |
| **5)** | Постройте таблицу значений функции **y=f(x)** при изменении **x** на отрезке**[a;b]**c шагом **h**  Отрезок **[-3;3],** шаг **h=0 .25** |
| **6)** | Постройте таблицу и найдите наибольшее значение функции **y=f(x)** при изменении **x** на отрезке**[a; b]**c шагом **h.**  Y=0.5. Отрезок **[0,10],** шаг **h=0.5** |
| **7)** | Постройте таблицу и вычислите произведение значений функции **y=f(x)** при **y>0** при изменении **x** на отрезке **[a; b]** с шагом **h**.  **Y=x** Отрезок **[0;10],** шаг **h=0.5** |
| **8)** | Постройте таблицу значений функции **y=f(x)** при изменении **x** на отрезке**[a; b]**c шагом **h**.  **Y=** Отрезок**[-2;2],** шаг **h=0.25** |
| **9)** | Постройте таблицу и найти наибольшее значение функции **y=f(x)** при изменении **x** на отрезке **[a;b]**c шагом **h**.  **Y=** Отрезок**[0.1;1.5],**шаг **h=0.1** |
| **10)** | Построите таблицу и вычислить сумму значений функции **y=f(x)** при **y<0** и при изменении **x** на отрезке **[a,b]** с шагом **h**.  **Y=0.5-0.1-sinx Отрезок [0,2], шаг h=** |
| **11)** | Постройте таблицу значений функции **y=f(x)** при изменении **x** на отрезке[**a; b]**c шагом **h**.  **Y=** Отрезок**[-2;2],** шаг **h=0.25** |
| **12)** | Постройте таблицу и вычислите произведение значений функции **y=f(x)** при **y<0** при изменении **x** на отрезке **[a; b]** с шагом **h.**  **Y=.** Отрезок**[-1.5,1.5**], шаг **h=0.15** |
| **13)** | Постройте таблицу и найдите наибольшее значение функции **y=f(x)** при изменении **x** на отрезке **[a; b]**c шагом **h.**  . Отрезок**[-5,5],** шаг **h=0.5** |
| **14)** | Постройте таблицу и вычислите сумму значений функции **y=f(x)** при **0.5<y<1.5** при изменении **x** на отрезке **[a; b]** с шагом **h**.  **Y=1+cos10x.** Отрезок **[],** шаг **h=.** |
| **15)** | Постройте таблицу и найдите наименьшее значение функции **y=f(x)** при изменении **x** на отрезке **[a; b]**c шагом **h**.  **Y=x+1/x.** Отрезок **[0.1;1.5],** шаг **h=0.1** |
| **16)** | Постройте таблицу и вычислите произведение значений функции **y=f(x)** при **y>0** при изменении **x** на отрезке **[a; b]** с шагом **h**.  **Y=.** Отрезок**[-0.5;1.5],** шаг **h=0.1** |
| **17)** | Постройте таблицу значений функции **y=f(x)** при изменении **x** на отрезке **[a; b]**c шагом **h**.  **Y=** Отрезок**[2;12],** шаг **h=0.5** |
| **18)** | Постройте таблицу значений функции **y=f(x)** при изменении **x** на отрезке **[a; b]**c шагом **h**.  **Y=** Отрезок**[-2;6],** шаг **h=0.5** |
| **19)** | Постройте таблицу и найдите наибольшее значение функции **y=f(x)** при изменении **x** на отрезке **[a;b]**c шагом **h**.  **Y=.** Отрезок**[2;4],** шаг **h=0.1** |
| **20)** | Постройте таблицу значений функции **y=f(x)** при изменении **x** на отрезке **[a;b]**c шагом **h**.  **Y=**Отрезок**[-3;3],**шаг **h=0.3** |
| **21)** | Постройте таблицу и найдите наибольшее значение функции **y=f(x)** при изменении **x** на отрезке **[a; b]**c шагом **h**.  . Отрезок**[6;8]**шаг **h=0.2** |
| **22)** | Постройте таблицу и вычислите сумму значений функции **y=f(x)** при **y<1.2** при изменении x на отрезке **[a; b]**c шагом **h**.  **Y=sin(4x)-2.** Отрезок [-], шаг **h=** |
| **23)** | Постройте таблицу значений функции **y=f(x)** при изменении **x** на отрезке **[a; b]**c шагом **h**.  **Y=** Отрезок**[-4;4**], шаг **h=0.5** |
| **24)** | Постройте таблицу и вычислите произведение значений функции **y=f(x)** при **y>0** при изменении **x** на отрезке **[a; b]**c шагом **h**.  **Y=.** Отрезок**[-3;0],** шаг **h=0.15** |
| **25)** | Постройте таблицу и найдите наибольшее значение функции **y=f(x)** при изменении **x** на отрезке **[a; b]**c шагом **h**.  **Y=arctg(x)-.** Отрезок **[0.1;1.5]**,шаг **h=0.1** |
| **26)** | Постройте таблицу и вычислите сумму значений функции **y=f(x)** при **y>0** при изменении **x** на отрезке **[a; b]**c шагом **h**.  . Отрезок [0; ],шаг **h=** |
| **27)** | Постройте таблицу и найдите наибольшее значение функции **y=f(x)** при изменении **x** на отрезке **[a; b**] c шагом **h**.  **.** Отрезок**[0.1;1.5],** шаг **h=0.1** |
| **28)** | Постройте таблицу и вычислите произведение значений функции **y=f(x**) при **y>0** при изменении **x** на отрезке **[a; b]** с шагом **h**.  **Y=.** Отрезок**[1.9;2.1],**шаг **h=0.01** |
| **29)** | Постройте таблицу и вычислите наименьшее значение функции y=f(x) при изменениях **х** на отрезке **[a; b]** с шагом **h.**  **Y=5\*sin(x+1)^2**. Отрезок[**-**], шаг **h=** |
| **30)** | Постройте таблицу и вычислите наибольшее значение функции y=f(x) при изменениях х на **[a; b]**c шагом h.  **Y=5\*sin(x+1)^4- cos(x)**. Отрезок[-], шаг **h=** |

Таблица 4.4.5.2

|  |  |
| --- | --- |
| **№** | **Задача** |
| **1)** | табл. 5-1 для создания 1-го проекта и найдите наибольшее значение функции при изменении **x** на отрезке **[a;b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  **a= - π b= π c= - π d= π h1=π/8 h2=π/4** **z(x,y)=** 3 cos2(2\*x+y+1) |
| **2)** | Постройте таблицу значений функции **z=f(x,y)** и найдите сумму и количество положительных значений функции при изменении **x** на отрезке **[a;b]** с шагом **h1** и **y** на отрезке [**c; d]** с шаго**м** **h2**.  **a= - π b= π c= - π d= π h1=π/6 h2=π/4**  **z(x,y)= 1/(x+2\*π-y) – sin(x)** |
| **3)** | Постройте таблицу значений функции **z(x,y)** при изменении **x** на отрезке **[a;b]** с шагом **h1** и **y** на отрезке [**c; d]** с шагом **h2**.  **а= - 1 b= 2 h1= 0.1 c = - 1 d=2 h2=0.2**  **x^2 + y^2** если 0≤ x ≤ 1 и 0 ≤ y ≤ 1 ,  z(x,y)= x - y если -1≤ x ≤0 и -1≤ y ≤0 ,  1 – x - y в противном случае |
| **4)** | Постройте таблицу значений функции **z=f(x,y)** и найдите произведение и количество положительных значений функции при изменении **x** на отрезке **[a;b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.  **a= 0 b= 10 c= - 2 d= 2 h1= 0.5 h2=0.2 z(x ,y )=x^4 +y- 10\*x^2 – 30\*x -25** |
| **5)** | Постройте таблицу значений функции **z=f(x,y)** и найдите наименьшее значение функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  **a= 0.1 b= 1.5 c= 1 d=2 h1=0.1 h2=0.1**  **z(x,y) = x + 1/x + (1 - y)/2** |
| **6)** | Постройте таблицу значений функции **z=f(x,y)** и найти сумму и количество отрицательных значений функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  **a= 0 b= 2π c=0 d=2π h1=π/8 h2=π/8**  **z(x,y)= y – 0.1\*sin(x)/2** |
| **7)** | Постройте таблицу значений функции **z(x,y)** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  **a= -3 b= 4 c = -2 d=2 h1= 0.25 h2=0.25**  e^x + y если x € [-2;2] и y € [-1;1],  z(x,y)= x+y+4 если -3 <x<-2 и - 2< y< - 1 ,  0 в противном случае |
| **8)** | Постройте таблицу значений функции **z=f(x,y)** и найдите произведение и количество отрицательных значений функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.  **a= -1.5 b= 1.5 c= -2 d= 2 h1= 0.2 h2= 0.2 z(x,y)= x^8-y- 0.4\*x^3 – 1.2** |
| **9)** | Постройте таблицу значений функции **z=f(x,y)** и найдите наименьшее значение функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  **a= 0.1 b=1.5 h1=0.1 c=2 d=3 h2=0.1 z(x,y) =** |
| **10)** | Постройте таблицу значений функции **z(x,y)** и найдите сумму и количество положительных значений функции при изменении **x** на отрезке **[a; b] с** шагом **h1** и **y** на отрезке [**c; d]** с шагом **h2.**  **a= -π b= π h1=π/8 c= - π d= π h2=π/**6  **z(x,y)= sin(4x+y) – 2\*x** |
| **11)** | Постройте таблицу значений функции **z(x,y)** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.  **a= 1 b=5 h1= 0.2 c=0 d=6 h2=0.4**  **x +y если x € [2;4] и y € [1;5],**    **z(x,y)=  если 1< x <2 и 0<**y<1,  **x - y в противном случае** |
| **12)** | Постройте таблицу значений функции **z=f(x,y)** и найдите произведение и количество отрицательных значений функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  **a=1.9 b=2.1 h1=0.01 c=2 d=3 h2=0.1**  **z(x,y)=** |
| **13)** | Постройте таблицу значений функции **z=f(x,y)** и найдите наименьшее значение функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  **a= 2.1 b=3 h1=0.1 c=2 d=3 h2=0.2 z(x,y) =** |
| **14)** | Постройте таблицу значений функции **z=f(x,y)** и найдите сумму и количество значений функции  **>0.1** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  **a=0 b= π h1= π/16 c=0 d=3 h2=0.1** z**(x,y)=** |
| **15)** | Постройте таблицу значений функции **z(x,y)** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.  **a= -2 b= 5 h1=0.5 b= 0 c=3 h2= 0.1**  **если x€[-1;4] и y€ [1;2]**  z**(x,y)= x – 3 + y если -2 < x < -1 и 0< y <1**  **в противном случае** |
| **16)** | Постройте таблицу значений функции **z=f(x,y)** и найдите произведение и количество значений функции **z(x,y) < 1** при изменении **x** на отрезке [**a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.  a= -π b= π h1=π/8 c= -π d= π h2=π/8  **z(x,y)=** |
| **17** | Постройте таблицу значений функции **z=f(x,y)** и найдите наибольшее значение функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[с; d]** с шагом **h2.**  a=-1.9 b= 2.9 h1=0.03 c= 1 d=2 h2=0.1 z(x,y)= Y**+.** |
| **18)** | Постройте таблицу значений функции **z=f(x,y)** и вычислите наименьшее значение функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.  **a= -3 b=3 h1=0.2 c= 1 d=2 h2= 0.1**  **z(x,y)= + y** |
| **19)** | Постройте таблицу значений функции **z=f(x,y)** и найдите сумму и количество значений функции при **z>0** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a= -1.9 b= 2.9 h1=0.3 c=2 d=3 h2=0.1 **z(x,y) =- y** |
| **20)** | Постройте таблицу значений функции **z=f(x,y)** при изменении **x** на отрезке **[a; b]**c шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a= -4 b=4 h1=0.5 c=2 d=3 h2=0.1  **z(x,y)=** |
| **21)** | Постройте таблицу значений функции **z=f(x,y)** и найдите сумму и количество значений функции при **z<1.5** при изменении **x** на отрезке **[a; b]** c шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a= -π b=π h1=π/8 c= -π d=π h2=π/8 **z(x,y)=sin(4x+y) - 2** |
| **22)** | Постройте таблицу значений функции **z=f(x,y)** и найдите наибольшее значение функции при изменении x на отрезке [**a; b]**c шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a= 6 b=8 h1= 0.2 c=2 d=3 h2=0.1  **z(x,y)=** |
| **23)** | Постройте таблицу значений функции **z=f(x,y)** и найдите произведение и количество значений функции при **z<3** при изменении **x** на отрезке **[a; b]** с шагом **h** и **y** на отрезке **[c; d]** с шагом **h2.**  a= 1.9 b=2.1 h1=0.01 c=2 d=3 h2=0.02 **z(x,y)=. + y** |
| **24)** | Постройте таблицу значений функции **z=f(x,y)** при изменении **x** на отрезке**[a; b]** c шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a=0 b=3 h1=0.1 c=1 d=4 h2=0.2  x+y-1 если x€ [1;2] и y€ [2;3]  z(x,y)= x-y если 0<x<1 и 3<y<4  12.3 в противном случае |
| **25)** | Постройте таблицу значенийфункции **z=f(x,y)** и найдите наименьшее значение функции при изменении x на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a= -π b=π h1=π/8 c=2 d=3 h2=0.1  **z(x,y)= 5\*sin(2\*x+1)^2 + y** |
| **26)** | Постройте таблицу значений функции **z=f(x,y)** и найдите сумму и количество значений функции **z(x,y)>0** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a= -2 b=5 h1= 0.2 c=0.1 d=0.2 h2=0.01 **z(x , y)= x + y - x/y** |
| **27)** | Постройте таблицу значений функции **z=f(x,y)** и найдите произведение и количество значений функции **z(x,y) < 0** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.  a= 20 d=30 h1= 5 c= 10 d=50 h2=5 **z(x,y)=17.5 + x – y** |
| **28)** | Постройте таблицу значений функции **z=f(x,y)** и найдите наибольшее значение функции при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a= - 5 b=5 h1= 0.5 c= 0.1 d=0.3 h2=0.02 **z(x,y) =** |
| **29)** | Постройте таблицу функции **z=f(x,y)** и найдите сумму и количество значений функции при **z>0** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.  a= -2 b=2 h1=0.5 c=2 d=3 h2=0.1 **z(x,y)= x/y + (x+y)/2 -** |
| **30)** | Постройте таблицу значений функции **z=f(x,y)** при изменении **x** на отрезке**[a; b]** c шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2.**  a= 0.1 b=0.8 c= 0.2 d= 0.6 h2=0.05  **Sin(x+y) если x€[0.5; 0.7] и y € [0.3;0.4]**  z(x,y**)= x-y если 0.1<x<0.5 и y € [0.4;0.6]**    1 в противном случае |

### Содержание отчёта

1. Фамилия, имя студента, номер группы, номер варианта.
2. **Реализация 1-го задания:**
3. Тема и название лабораторной работы.
4. Задание на разработку задачи.
5. Формализация и уточнение задания.
6. Элементы, разрабатываемого проекта:
7. графический интерфейс пользователя;
8. таблица свойств объектов;
9. схемы алгоритмов;
10. программный код проекта;
11. Результаты выполнения первого проекта.
12. Доказательства правильности результатов.
13. **Реализация 2-го задания:**
14. Тема и название лабораторной работы.
15. Задание на разработку задачи.
16. Формализация и уточнение задания.
17. Элементы, разрабатываемого проекта:
18. графический интерфейс пользователя;
19. таблица свойств объектов;
20. схема алгоритма;
21. программный код проекта;
22. Результаты выполнения проекта второго проекта.
23. Доказательства правильности результатов.

### Пример выполнения 1-го 2-го заданий

1. **Фамилия, имя студента, номер группы, номер варианта.**

Иванов И., БИН1405, вариант 13.

1. **Реализация 1-го задания:**
   1. **Тема и название лабораторной работы:**

Программирование алгоритмов регулярных циклических структур.

Формирование таблицы значений функции **y=f(x)** на отрезке  
**[a; b]** с шагом **h**.

* 1. **Задание на разработку проекта:**

Создайте проект ***Проект 5.4-1* для формирования таблицы** значений функции **f(x)=sin(x+2)\*12+cos(x\*4)\*15** при изменении **x** на отрезке **[a;b]** с шагом **h** **(а=2, b=8, h=0.5)**. Найдите наибольшее и наименьшее значения функции на этом же отрезке, а также число положительных и отрицательных значений функции.

* 1. **Формализация и уточнение задания**

Для решения задачи табуляции функции на заданном интервале следует использовать алгоритмическую структуру регулярного типа, т.к. заданы начальное и конечное значения аргумента, а также шаг его изменения. Таким образом, в данном случае можно заранее определить количество вычисленных значений.

* 1. **Элементы, разрабатываемого проекта:**

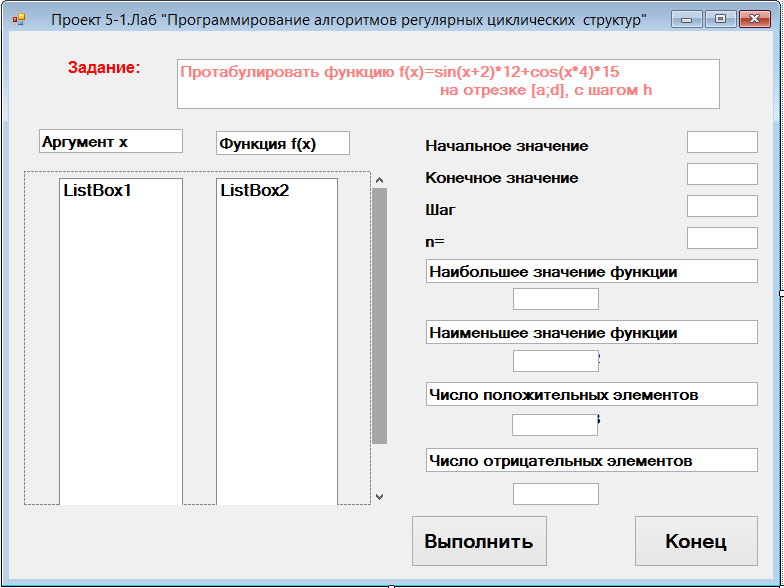
1. **Графический интерфейс пользователя:**

Форма приложения может иметь такой же вид, как на   
 рис. 4.5.4-1.

При размещении первого элемента управления **ListBo**x в форму на экране появляется пустой прямоугольник, внутри которого находится текст **ListBox1**. В этом списке нет ни полосы прокрутки, ни элементов. Вертикальная полоса прокрутки появляется автоматически, в том случае, когда новый добавляемый элемент не может быть размещен в выделенной для списка области экрана. В данном проекте два элемента **ListBox** располагаются на элементе управления (служит для объединения нескольких элементов в одну группу) и поэтому будут иметь одну общую полосу прокрутки.

Однако данные можно занести в список и на стадии разработки проекта. Для этого надо поместить элементы списка в свойство **Items** (Collections), которое имеется в окне **Свойства** элемента управления **ListBox**, щелкнув кнопку с тремя точками и составив нужный список в открывшемся диалоговом окне. Каждая строка этого свойства задает один элемент, для перехода на следующую строку надо нажать клавишу <Enter>, после набора всех элементов – щелкнуть ОК.

На кнопке с именем **Button1** поместите надпись:   
***Выполните***, а на кнопке **Button2** – надпись ***Конец***.



*Рис 4.5.4*-*1. Форма проекта* 1-го задания ***Проект 5-1:*** Формирование таблицы значений функции **y=f(x)** на отрезке  
**[a; b]** с шагом **h**

1. **Таблица свойств объектов:**

Определите, установите и сведите в таблицу табл. 4.5.4-3

свойства всех объектов.

Таблица 4.5.4-3

|  |  |  |
| --- | --- | --- |
| **Имя объекта** | **Свойство** | Значение **свойства** |
| **Form1** | Text | *Тема 4.5. Программирование алгоритмов регулярных циклических структур и цикл в цикле* |
| **Label1** | Name | *Label1* |
| Text | *Начальное значение* |
| ForeColor | *Синий* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **Label2** | Name | *Label2* |
| Text | *Конечное значение* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **Label3** | Name | *Label3* |
| Text | *Шаг* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **Label4** | Name | *Label4* |
| Text | *n=* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **Label5** | Name | *Label5* |
| Text | *Задание* |
| ForeColor | *Красный* |
| Font | *Arial, Жирный, 12 пунктов* |
| **TextBox1** | Name | *TextBox1* |
| Text | *Протабулировать функцию f(x)=sin(x+2)\*12+cos(x\*4)\*15* |
| ForeColor | *Красный* |
| Font | *Microsoft Sans Serif, Жирный, 10 пунктов* |
| **TextBox2** | Name | *TextBox2* |
| Text | *Аргумент х* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **TextBox3** | Name | *TextBox3* |
| Text | *Функция f(x)* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **TextBox8** | Name | *TextBox8* |
| Text | *Наибольшее значение функции* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **TextBox10** | Name | *TextBox10* |
| Text | *Наименьшее значение функции* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **TextBox12** | Name | *TextBox12* |
| Text | *Число положительных элементов* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **TextBox14** | Name | *TextBox7* |
| Text | *Число отрицательных элементов* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **TextBox4**  **TextBox5**  **TextBox6**  **TextBox7**  **TextBox9**  **TextBox11**  **TextBox13**  **TextBox15** | Name | *TextBox4, TextBox5, TextBox6, TextBox7, TextBox9, TextBox11, TextBox13, TextBox15* |
| Text |  |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **ListBox1** | Name | *ListBox1* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **ListBox2** | Name | *ListBox2* |
| ForeColor | *Черный* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| **Panel1** | Name | *Panel1* |
| Font | *Microsoft Sans Serif, Жирный, 8 пунктов* |
| AutoScroll | *True* |
| **Button1** | Name | *Button1* |
| Text | *Выполнить* |
| **Button2** | Name | *Button2* |
| Text | *Конец* |

1. **Схемы алгоритмов:**

Схемы алгоритмов представлены на рис. 4.5.4-2.

|  |  |
| --- | --- |
|  |  |

*Рис. 4.5.4-2. Схемы алгоритмов процедур* **Fun(x)**и **RegCikl** *проекта* 1-го задания ***Проект 5-1:***  Формирование таблицы значений функции **y=f(x)** на отрезке **[a; b]** с шагом **h**

1. **Программный код проекта:**

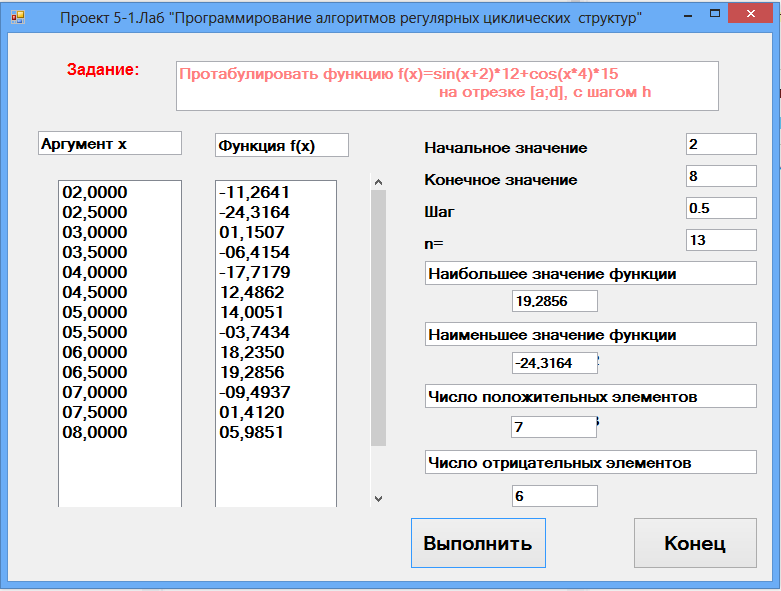
Программный код проекта приведен в на рис. 4.5.4-3.

|  |
| --- |
| **Imports System.Math**  **Public Class Form1**  *'Функция ввода исходных данных из TextBox*  **Function vvod(ByVal T As TextBox) As Double**  **Return Val(T.Text)**  **End Function**  *'Процедура вывода результата в TextBox*  **Sub vivod(ByVal Z As Double, ByVal T As TextBox)**  **T.Text = Format(Z, "00.0000")**  **End Sub**  *'Процедура вывода результата в ListBox*  **Sub vivodList(ByVal Z As Double, ByVal LB As ListBox)**  **LB.Items.Add(Format(Z, "00.0000"))**  **End Sub**  *'Процедура вывода целого результата в TextBox*  **Sub vivodint(ByVal Z As Integer, ByVal T As TextBox)**  **T.Text = CStr(Z)**  **End Sub**  *'Функция вычисления значения функции*  **Public Function Fun(ByVal x As Double) As Double**  **Return Sin(x + 2) \* 12 + Cos(x \* 4) \* 15**  **End Function**  *'Процедура решения задачи*  **Public Sub RegCikl(ByVal a As Double, ByVal b As Double,\_**  **ByVal h As Double, ByRef max As Double, \_**  **ByRef min As Double, ByRef n1 As Integer, \_**  **ByRef n2 As Integer)**  **Dim i As Integer, n As Integer, y As Double, x As Double**  **n = CInt((b - a) / h + 1)***'количество значений функции*  **vivodint(n, TextBox7)**  **x = a : n1 = 0 : n2 = 0**  **max = Double.MinValue : min = Double.MaxValue**  **For i = 1 To n**  **y = Fun(x)**  **If y > max Then max = y : If y < min Then min = y**  **If y > 0 Then n1 = n1 + 1 : If y < 0 Then n2 = n2 + 1**  **vivodList(x, ListBox1) : vivodList(y, ListBox2)**  **x = x + h**  **Next**  **End Sub**  **Private Sub Button1\_Click(sender As Object, \_**  **e As EventArgs) Handles Button1.Click**  **Dim nn1 As Integer, nn2 As Integer**  **Dim mm1 As Double, mm2 As Double**  **Dim a As Double, b As Double, h As Double**  **a = vvod(TextBox4) : b = vvod(TextBox5) : h = vvod(TextBox6)**  **RegCikl(a, b, h, mm1, mm2, nn1, nn2)**  *'вызов проц. реш.задачи*  **vivod(mm1, TextBox9) : vivod(mm2, TextBox11)**  **vivodint(nn1, TextBox13) : vivodint(nn2, TextBox15)**  **End Sub**  **Private Sub Button2\_Click(sender As Object, \_**  **e As EventArgs) Handles Button2.Click**  **End**  **End Sub**  **End Class** |

*Рис. 4.5.4-3. Программный код проекта* 1-го задания ***Проект 4.5-1:*** Формирование таблицы значений функции **y=f(x)** на отрезке **[a;b]** с шагом **h**

* 1. **Результаты выполнения проекта:**

Результат выполнения программы приведен на рис. 4.5.4-4.



*Рис 4.5.4-4. Результаты выполнения проекта* 1-го задания ***Проект 5-1:*** Формирование таблицы значений функции **y=f(x)** на отрезке **[a;b]** с шагом **h**

* 1. **Доказательства правильности результатов:**

Для доказательства правильности результата протабулируйте свой вариант функции с использованием программы Microsoft Excel, сделайте вывод о том, что полученные результаты подтверждают правильность выполнения программы.

1. **Реализация 2-го задания:**
2. **Тема и название лабораторной работы:**

Программирование алгоритмов циклических структур и цикл в

цикле.

Формирования таблицы значений функции **y1(x, y)** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**.

1. **Задание на разработку задачи:**

Создайте проект ***Проект 5-2*** для формирования таблицы значений

функции **y1(x, y)** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2** (**а=-1; в=2; с=-1; d=2; h1=1; y2=1**).

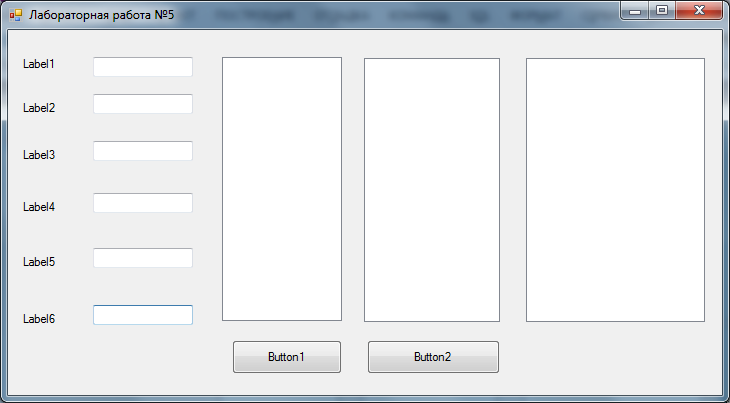
1. **Формализация и уточнение задания:**

Для решения задачи табуляции функции на заданных интервалах следует использовать алгоритмическую структуру регулярного типа, т.к. заданы начальные и конечные значения аргумента, а также шаги их изменения, то есть в этом случае можно заранее определить количество вычисленных значений.

1. **Элементы, разрабатываемого проекта:**
2. **Графический интерфейс пользователя:**

Форма приложения может иметь такой же вид,

как на рис. 4.5.4-5.



*Рис. 4.5.4-5. Форма проекта* 2-го задания ***Проект 5-2:*** Формирования таблицы значений функции **y1(x, y)** при изменении **x** на отрезке **[a; b]** с шагом **h1** и **y** на отрезке **[c; d]** с шагом **h2**

1. **Таблица свойств объектов:**

Таблицу свойств объектов разработайте

самостоятельно.

1. **Схемы алгоритмов:**

Схемы алгоритмов представлены на рис.4.5.4-6.

|  |  |
| --- | --- |
|  |  |

*Рис. 4.5.4-6. Схемы алгоритмов процедур* **fun1(), fun2()** *и* **Cikl()**

*проекта* 2-го задания **Проект 5-2*:***

Формирования таблицы значений функции **y1(x, y)** при изменении **x** на отрезке **[a;b]** с шагом **h1** и **y** на отрезке **[c;d]** с шагом **h2**

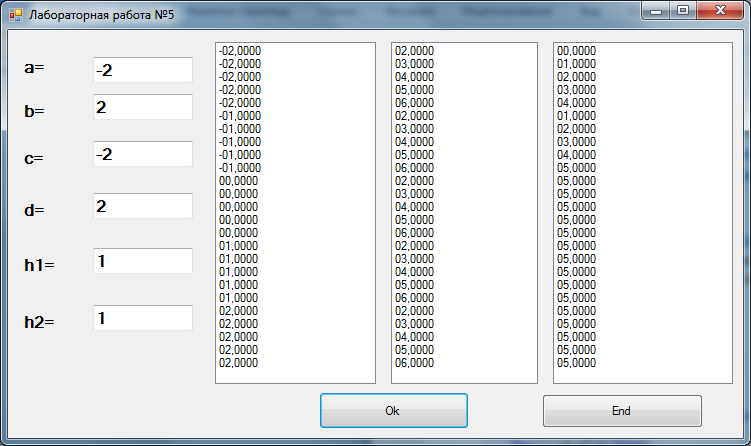
1. **Программный код проекта;**

Программный код проекта 2-го задания – на рис. *4.5.4*-7.

|  |
| --- |
| **Public Class Form1**  **Function vvod(ByVal t As TextBox) As Double**  **Return Val(t.Text)**  **End Function**  **Sub vivodlist(ByVal z As Double, ByVal lb As ListBox)**  **lb.Items.Add(Format(z, "00.0000"))**  **End Sub**  **Public Function fun1(ByVal x As Double, \_**  **ByVal y As Double) As Double**  **Return x ^ 2 + y ^ 2**  **End Function**  **Public Function fun2(ByVal x As Double, \_**  **ByVal y As Double) As Double**  **Return x + y**  **End Function**  **Public Sub Cikl(ByVal a As Double, ByVal b As Double,\_**  **ByVal c As Double, ByVal d As Double, \_**  **ByVal h1 As Double, ByVal h2 As Double)**  **Dim i, j, n1, n2 As Integer**  **Dim x, y, y1 As Double**  **n1 = CInt(Fix(b - a)/h1 + 1) : n2 = CInt(Fix(d - c)/h2 + 1)**  **x = a**  **For i = 1 To n1**  **y = b**  **For j = 1 To n2**  **If (x > 0 And x < 1) And (y > 0 And y < 1) Then**  **y1 = fun1(x, y)**  **Else**  **y1 = fun2(x, y)**  **End If**  **vivodlist(x, ListBox1) : vivodlist(y, ListBox2)**  **vivodlist(y1, ListBox3)**  **y = y + h2**  **Next**  **x = x + h1**  **Next**  **End Sub**  **Private Sub Button1\_Click(sender As Object, \_**  **e As EventArgs) Handles Button1.Click**  **Dim a, b, c, d, h1, h2 As Double**  **a = vvod(TextBox1) : b = vvod(TextBox2)**  **c = vvod(TextBox3) : d = vvod(TextBox4)**  **h1 = vvod(TextBox5)**  **h2 = vvod(TextBox6)**  **Cikl(a, b, c, d, h1, h2)**  **End Sub**  **Private Sub Button2\_Click(sender As Object,\_**  **e As EventArgs) Handles Button2.Click**  **End**  **End Sub**  **End Class** |

*Рис. 4.5.4* -7*. Программный код проекта* 2-го задания ***Проект 5-2:*** Формирования таблицы значений функции **y1(x, y)** при изменении **x** на отрезке **[a;b]** с шагом **h1** и **y** на отрезке **[c;d]** с шагом **h2**

1. **Результаты выполнения проекта приведены на рис. 4.5.4** **-8.**



*Рис. 4.5.4* *-8. Результаты выполнение проекта* 2-го задания ***Проект 5-2:*** Формирования таблицы значений функции **y1(x, y)** при изменении **x** на отрезке **[a;b]** с шагом **h1** и **y** на отрезке **[c;d]** с шагом **h2**

1. **Доказательства правильности результатов:**

Для доказательства правильности результата протабулируйте свой вариант функции в ручную, сделайте вывод о том, что полученные результаты подтверждают правильность выполнения программы.

### Контрольные вопросы

1. Что называется, циклическим алгоритмом?
2. Как записывается цикл с оператором For в блок-схемах?
3. Как записывается цикл с оператором For в программах?
4. Как работает цикл с оператором For?
5. Что называется, циклом?
6. Что называется, регулярным циклом?
7. Какой оператор языка VB служит для организации регулярных циклов и как он работает?
8. Что такое параметр цикла, и какой тип данных рекомендуется использовать для его описания?
9. Каким образом определить число повторений цикла, если переменная, меняющая свое значение от цикла к циклу, вещественного типа?
10. Что такое вложенные циклы, каким образом они изображаются в схеме алгоритма и в программе?
11. Для чего служит оператор Exit For?
12. В чем суть базовых алгоритмов: табулирование функции в заданном диапазоне изменения аргумента; определение суммы (произведения) значений функции в границах изменения аргумента;
13. поиск минимального (максимального) значения функции в заданном диапазоне изменения аргумента?
14. В чем суть алгоритма табулирования функции в заданном диапазоне изменения аргумента?
15. В чем суть алгоритма определения суммы (произведения) значений функции в пределах заданных границ изменения аргумента?
16. В чем суть алгоритма поиска минимального (максимального) значения функции в заданном диапазоне изменения аргумента?
17. Как называется переменная, изменяющая свое значение при каждом новом вхождении в цикл?
18. Как называется многократно повторяющаяся часть алгоритма?
19. Как определяется число повторений операторов тела цикла в регулярной циклической структуре?
20. Что является телом цикла в операторе Fоr … Next?
21. Что необходимо для досрочного прекращения регулярного цикла?
22. Для чего необходимо ключевое слово Step?
23. Что следует принять за начальное наименьшее значение для нахождения наименьшего значения числовой функции от аргумента, значение которого изменяется на заданном интервале?
24. Что следует принять за начальное значение произведения для нахождения значения произведения числовой функции от аргумента, значение которого изменяется на заданном интервале?