## Тема 4.8

## Программирование алгоритмов формирования и обработки двумерных массивов

### 4.8.1. Средства описания и работы с двумерными массивами данных

При решении математических, экономических, статистических задач информация располагается в матрицах и таблицах, которые являются прототипом двумерного массива в программировании. Если в одномерных массивах каждый элемент содержит только один индекс, то в двумерных массивах каждый элемент имеет два индекса – номера строки и номера столбца.

Из предыдущей темы известно, что ***массив*** представляет собой последовательность переменных одинакового типа, объединенных общим ***именем***.

***Количество индексов (измерений)*** указывает на ***размерность*** (***ранг***) массива. Например, ***двумерный массив* b(3,2)**:

|  |  |  |
| --- | --- | --- |
| **b(0,0)** | **b(0,1)** | **b(0,2)** |
| **b(1,0)** | **b(1,1)** | **b(1,2)** |
| **b(2,0)** | **b(2,1)** | **b(2,2)** |
| **b(3,0)** | **b(3,1)** | **b(3,2)** |

Так, в приведенном выше примере размерность массива **b(3,2)** *–* 2**,** т.е. массив двумерный (имеет два измерения). В нашем примере ***количество элементов*** массива **b(3,2)**равно **12** (4 строки и 3 столбца).

Перед использованием двумерного массива в программе, его необходимо объявить с помощью оператора **Dim**, который выделяет место в памяти для размещения элементов статического массива. Например:

|  |
| --- |
| **Dim b(3,2) As Single** |

Необходимо обратить внимание на то, что, как при описании – **A(m,n)**, так и при обращении – **A(i,j)** к элементам двумерного массива, в скобках ***сначала указывается номер строки, а затем номер столбца* (**рис. 4.8.1-1)*.*

Для определения параметров двумерного массива могут использоваться свойство класса **Array** – Rank и метод класса **Array** – **GetLength().** Свойство Rank используется для определения количества измерений (ранга) массива (для двумерного массива ранг равен 2). Причем каждое измерение в массиве может иметь свою длину. Метод **GetLength()** используется для определения количества элементов в заданном измерении массива (для первого измерения используется значение 0, для второго – 1 и т.д.). Необходимо обратить внимание, что метод **GetLength()** возвращает значение, которое на единицу больше, чем указанное в объявлении **Dim**, так как индексация элементов массива в VBвсегда начинается с нуля, а метод **GetLength()** определяет количество элементов, считая от единицы.

***Матрица прямоугольная* A(n,m)** - n≠m ***Матрица квадратная* A(n,m)** - n=m

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | 0 | j | m | | 0 |  |  |  | |  |  |  |  | | i |  | A(i,j) |  | |  |  |  |  | | **n** |  |  |  | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 0 |  | j |  | m | | 0 |  |  |  |  |  | |  |  |  |  |  |  | | i |  |  |  |  |  | |  |  |  |  |  |  | | n |  |  |  |  |  |   Побочная диагональ Главная диагональ  матрицы i=n-j матрицы i= j |

*Рис. 4.8.1-1. Прямоугольная и квадратные матрицы*

Так, для определения параметров массива **Mac1(,),** описанного как

|  |
| --- |
| **Dim Mac1(4,7) As Integer** |

можно воспользоваться следующими выражениями:

|  |
| --- |
| **ArrayRank = Mac1.Rank**  **RawCount = Mac1.GetLength(ArrayRank-2)**  **ColumnCount= Mac1.GetLength(ArrayRank-1)** |

Где **ArrayRank** – ранг матрицы (равен 2);

**RawCount** – количество строк (равно 5);

**ColumnCount** – количество столбцов (равно 8).

Таким образом, **ArrayRank, RawCount, ColumnCount** являются переменными целочисленного типа, которые принимают значения 2, 5 и 4.8.

Кроме того, для определения параметров двумерного массива можно использовать метод **GetUpperBound()**, который находит верхнюю границу (максимальный индекс) заданного измерения. В качестве параметра этот метод принимает значения аналогично методу **GetLength()**, т.е. для первого измерения (строк) используется значение 0, для второго (столбцов) – 1.

Например, для массива **Mac2(4,7)**:

|  |
| --- |
| **Dim Mac2(4,7) As Double**  **Dim Cтрока1, Столбец2 As Integer**  **Строка1 = Mac2.GetUpperBound(0)** *'Значение Строка1 'равно 4*  **Столбец2 = Mac2.GetUpperBound(1)** *'Значение Столбец2 'равно 7* |

Ввод, вывод и обработка двумерных массивов, как правило, основаны на ис­пользовании вложенных циклов, которые обеспечивают перебор всех элементов массива. В некоторых задачах может иметь значение ***порядок перебора*** элементов массива: «***по строкам***» или «***по столбцам***». Если внешний цикл будет организован по первому индексу (по строкам), а внутренний цикл по второму индексу (по столб­цам), то выполняется построчный перебор элементов двумерного массива. Если внешний цикл в качестве параметра использует второй индекс, а внутренний цикл – первый индекс, то эле­менты массива перебираются по столбцам.

**Пример 4.8.1-1. Написать процедуры ввода/вывода, которые могут использоваться в алгоритмах обработки двумерных массивов.**

Некоторые процедуры ввода и вывода приведены в **Темах 4.3, 4.4, 4.5, 4.6 и 4.7.** Остальные процедуры ввода и вывода, которые можно использовать при написании базовых алгоритмов формирования и обработки двумерных массивов, представлены на рис. 4.8.1-2, 4.8.1-3, 4.8.1-4 и 4.8.1-5.

|  |
| --- |
| *'Процедура ввода элементов двумерного массива Single с клавиатуры*  **Sub vvodSngMac18(ByRef a(,) As Single)**  **Dim i, j, m, n As Integer, y As Single**  **m = a.GetLength(0) – 1 : n = a.GetLength(1) - 1**  **Dim r1, r2 As String**  **For i = 0 To m**  **For j = 0 To n**  **r1 = СStr(i) : r2 = СStr(j)**  **y = InputBox("эл-т массива a(" + r1 + "," + r2 + ") = ", \_**  **"Ввод эначений эл-тов массива a()")**  **a(i,j) = CSng(Val(y))**  **Next j**  **Next i**  **End Sub** |

*Рис. 4.8.1-2. Процедура ввода элементов двумерного массива***vvodSngMac18() Примера 4.8.1-1**

|  |
| --- |
| *'Процедура формирования двумерного массива случайными числами*  **Sub vvod2(ByRef a(,) As Double, ByVal m As Integer, \_**  **ByVal n As Integer)**  **Dim i, j As Integer**  **Randomize()**  **For i = 0 To m**  **For j = 0 To n**  **a(i, j) = 10 \* Rnd() - 5**  **Next i**  **Next j**  **End Sub** |

*Рис. 4.8.1-3. Процедура формирования массива* **vvod2()**

*случайными числами* **Примера 4.8.1-1**

|  |
| --- |
| *'Процедура форматированного вывода двумер-го массива в ListBox*  **Sub vivodSngMac19(ByRef x(,) As Single, ByRef LB As ListBox)**  **Dim i, j, m, n As Integer**  **Dim z, z1 As String**  **m = x.GetLength(0) – 1 : n = x.GetLength(1) - 1**  **LB.Items.Clear()**  **For i = 0 To m**  **z= ""**  **For j = 0 To n**  **z1 = Format(x(i,j), "0.000")**  **If x(i,j) < 0 Then**  **z1 = Space(2) + z1**  **Else**  **z1 = Space(3) + z1**  **End If**  **z = z + z1**  **Next j**  **LB.Items.Add(z)**  **Next i**  **End Sub** |

*Рис. 4.8.1-4. Процедура форматированного вывода массива* **vivodSngMac19()**

**Примера 4.8.1-1**

На рис. 4.8.1-4 представлен пример процедуры вывода целочисленной матрицы в **TextBox*.*** Напомним, что для того, чтобы в элементе управления **TextBox** можно было записать несколько строк текста, необходимо присвоить его свойству MultiLine значение *True*.

|  |
| --- |
| *'Процедура форматированного вывода двум-го массива Integer в TextBox*  **Sub vivodIntMac20(ByRef x(,) As Integer, ByRef TB As TextBox)**  **Dim i, j, m, n As Integer**  **Dim z As String**  **m = x.GetLength(0) – 1 : n = x.GetLength(1) - 1**  **TB.Text = ""**  **For i = 0 To m**  **For j = 0 To n**  **If Abs(x(i,j)) >= 100 Then**  **z = CStr (x(i,j)) & Space(2)**  **ElseIf Abs(x(i,j)) >= 10 Then**  **z = CStr (x(i,j)) & Space(4)**  **Else**  **z = CStr (x(i,j)) & Space(6)**  **End If**  **If x(I,j) >= 0 Then z = Space(1) + z**  **TB.Text = TB.Text & z**  **Next j**  **TB.Text = TB.Text & vbCrLf**  **Next i**  **End Sub** |

*Рис. 4.8.1-5. Процедура форматированного вывода массива***vivodIntMac20** *в TextBox* **Примера 4.8.1-1**

### 4.8.2. Базовые алгоритмы обработки двумерных массивов

Для работы с массивами, как правило, используются алгоритмы регулярной циклической структуры.

К базовым алгоритмам обработки двумерных массивов можно отнести алгоритмы:

* Нахождение количества элементов матрицы при заданном условии   
  (Пример 4.8.2-1).
* Нахождение суммы значений элементов матрицы при заданном условии (Пример 4.8.2-2).
* Нахождение произведения значений элементов матрицы при заданном условии (Пример 4.8.2-3).
* Поиск экстремальных значений элементов матрицы   
  (Пример 4.8.2-4 – 4.8.2-5).
* Формирование матрицы в соответствии с определенными правилами (Пример 4.8.2-6 – 4.8.2-7).
* Формирование одномерного массива из двумерного в соответствии с некоторым условием ((Пример 4.8. 2-8)).
* Транспонирование матриц ((Пример 4.8.2-9)).
* Произведение двух матриц ((Пример 4.8.2-10)).

**Пример 4.8.2-1. Написать процедуру, которая определяет количество элементов матрицы a(,) при условии a(i,j) > 0.**

|  |
| --- |
| Function Pr821(ByRef a(,)) As Integer  Dim kol As Integer = 0  For i = 1 To a.GetLength(0) – 1  For j = 1 To a.GetLength(0) – 1  If a (i,j) > = 0 Then kol = kol + 1  **Next j**  **Next i**  **Return kol**  **END Function** |

Рис. 4.8.2-1. Программный код процедуры ***Pr821()***

**Примера 4.8.2-1**

**Пример 4.8.2-2. Написать процедуру, которая вычисляет сумму значений элементов матрицы b(,) для элементов b(i,j) > 0.**

|  |
| --- |
| Function Pr822(ByRef b(,) As Single  Dim i, j, sum As Single  sum = 0  For i = 1 To b.GetLength(0) – 1  For j = 1 To b.GetLength(1) – 1  If b(i,j) > 0 Then sum = sum + b(i,j)  **Next j**  **Next i**  **Return sum**  **END Function** |

Рис. 4.8.2-2. Программный код процедуры ***Pr822()***

**Примера 4.8.2-2**

**Пример 4.8.2-3. Написать процедуру, которая вычисляет произведение значений элементов матрицы для элементов x(i,j) < 0.**

|  |
| --- |
| Function Pr823(ByRef x( , ) As Single  Dim i, j As Integer  Dim proizv As Single  proizv = 1  For i = 1 To x.GetLength(0) – 1  For j = 1 To x.GetLength(1) – 1  If a(i, j) < 0 Then proizv = proizv \* a(i, j)  **Next**  **Next**  **END Function** |

Рис. 4.8.2-3. Программный код процедуры ***Pr823()***

**Примера 4.8.2-3**

**Пример 4.8.2-4. Написать процедуру, которая определяет максимальный элемент среди элементов, лежащих ниже главной диагонали массива.**

В переменной **xmax** в цикле фиксируется мак­симальный элемент массива, в переменной **imax, jmax** – номер строки и номер столбца.

|  |  |
| --- | --- |
|  | **Sub Pr824(ByRef x(,) As Single, \_**  **ByRef xmax As Single, \_**  **ByRef imax As Integer, \_**  **ByRef jmax As Integer)**  **Dim i, j, m As Integer**  **m = x.GetLength(0) – 1**  **imax = 1 : jmax = 0 : xmax = x(1,0)**  **For i = 0 To m**  **For j = 0 To i - 1**  **If x(i,j) > xmax Then**  **xmax = x(i,j) : imax = i : jmax = j**  **End If**  **Next j**  **Next i**  **End Sub** |

*Рис. 4.8.2-4. Программный код процедуры* **Pr824()**

**Примера 4.8.2-4**

**Пример 4.8.2-5. Написать процедуру, которая определяет минимальный элемент, лежащий выше главной диагонали матрицы.**

Алгоритм и программный код процедуры представлен на рис. 4.8.2-5.

|  |  |
| --- | --- |
|  | **Sub Pr825(ByRef x(,) As Single, \_**  **ByRef xmin As Single, \_**  **ByRef imin As Integer,**  **ByRef jmin As Integer)**  **Dim i, j, m As Integer**  **m = x.GetLength(0) – 1**  **imin = 0 : jmin = 1 : xmin = x(0,1)**  **For i = 0 To m**  **For j = i + 1 To m**  **If x(i,j) < xmin Then**  **xmin = x(i,j) : imin = i**  **jmin = j**  **End If**  **Next j**  **Next i**  **End Sub** |

*Рис. 4.8.2-5. Программный код процедуры* **Pr825()**

**Примера 4.8.2-5**

**Пример 4.8.2-6. Написать процедуру, которая заполняет элементы массива f(5,2) целыми числами, а затем переставляет элементы первого и последнего столбцов.**

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 10 | 11 | 12 |
| 13 | 14 | 15 |
| 16 | 17 | 18 |

Алгоритм и код программы приведены на   
рис. 4.8.2-6.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 10 | 11 | 12 |
| 13 | 14 | 15 |
| 16 | 17 | 18 |

Данный пример демонстрирует заполнение массива целых чисел **f(,)** и перестановку элементов первого и последнего столбцов. Исходные данные для инициализации массива **f(,)** записываются в фигурных скобках при описании массива. Вложенные циклы обеспечивают перебор элемен­тов двумерного массива по столбцам: внутренний цикл по пер­вому индексу, а внешний по второму. Перестановка элементов первого и последнего столбцов в программе вы­полняется в цикле, изменяющем номер строки **i**, в котором выполняется обмен значениями элементов **f(i,0)** и **f(i,2).**

После этого мас­сив выводится на экран в следующем виде:

|  |
| --- |
| **Sub Pr826()**  **Dim i, j, r, c1, c2 As Integer**  **Dim f(,) As Integer={{1,2,3}, {4,5,6}, {7,8,9}, {10,11,12}, \_**  **{13,14,15},{16,17,18}}**  **c1 = 0 : c2 = 2** *'номера столбцов*  **vivodIntMac20(f, TextBox1)'Вывод мас.**  **Pr826(f, c1, c2)** *'Перест.1-го и 2-го столбцов*  **vivodIntMac20(f, TextBox2)** *'Вывод преобразованно массива*  **End Sub**  *'Процедура перестановки столбцов*  **Sub ПерСтол(ByRef f(,) As Integer, ByVal cc1 As Integer, \_**  **ByVal cc2 As Integer)**  **Dim ArrayRank, RawCount, i, r As Integer**  **ArrayRank = f.Rank : RawCount = f.GetLength(ArrayRank - 2)**  **For i = 0 To RawCount - 1**  **r = f(i,cc1) : f(i,сс1) = f(i,cc2) : f(i,cc2) = r**  **Next i**  **End Sub** |

*Рис. 4.8.2-6. Программный код процедуры* **Pr826() Примера 4.8.2-6**

**Пример 4.8.2-7. Написать процедуру, которая формирует двумерный массив x по правилу: x(i ,j ) = 2\*(i - 2) 2 - (j - 2)2).**

Алгоритм и процедура данной задачи приведены на рис. 4.8.2-7.

|  |
| --- |
| *'Формирование двумерного массива x(,)*  **Sub Pr827(ByRef x(,) As Single)**  **Dim i, j, m As Integer**  **m = x.GetLength(0) - 1**  **For i = 0 To m**  **For j = 0 To m**  **x(i, j) = CSng(2 \* (i - 2) ^ 2 - (j - 2) ^ 2)**  **Next**  **Next**  **End Sub** |

*Рис. 4.8.2-7. Программный код процедуры* **Pr827()**

**Примера 4.8.2-7**

**Пример 4.8.2-4.8. Написать процедуру, которая формирует одномерный массив, каждый элемент которого представляет собой количество положительных элементов соответствующего столбца массива b(,).**

|  |  |
| --- | --- |
|  | **Sub Pr828(ByRef b(,) As Single, \_**  **ByRef a() As Integer)**  **Dim Rank,RC,CC,j,i,n As Integer**  **Rank = b.Rank**  **RC = b.GetLength(Rank - 2) - 1**  **CC = b.GetLength(Rank - 1) - 1**  **ReDim a(CC)**  **Dim m As String**  **For j = 0 To CC**  **n = 0**  **For i = 0 TO RC**  **If b(i,j) > 0 Then n = n + 1**  **Next i**  **a(j)= n**  **Next j**  **End Sub** |

*Рис. 4.8.2-8. Программный код процедуры* **Pr828()**

**Примера 4.8.2-8**

Предполагается, что выделение памяти и ввод массива **b(,)** осуществляются в процедуре, которая вызывает процедуру **Pr828().**

Алгоритм и код программы представлены на рис. 4.8.2-5.

Для перебора всех элементов исходной таблицы **b(,)** также использованы вложенные циклы, в данном случае во внешнем цикле меняется но­мер столбца, а во внутреннем – номер строки. Такой порядок перебора элементов двумерного массива диктуется условиями задачи. В каждом столбце про­изводится подсчет положительных элементов, и по окончании цикла ре­зультат записывается в соответствующий элемент одномерного массива **а()**, номер которого совпадает с номером столбца матрицы **b(,)**.

**Пример 4.8.2-9. Написать процедуру, которая из матрицы х(n,m), являющейся входным параметром, получает транспонированную к ней матрицу y(m, n).**

Алгоритм и процедура приведены на рис. 4.8.2-9.

Для решения поставленной задачи в алгоритме необходимо реализовать вложенные циклы перебора всех элементов исходной матрицы **х(,)**. Транспонирование матрицы осуществляется путем перестановки (обмена) индексов транспонированной матрицы по отношению к исходной.

|  |  |
| --- | --- |
|  | **Sub Pr829(ByRef x(,) As Single, \_**  **ByRef y(,) As Single )**  **Dim ArrayRank, n, m, j, i As Integer**  **ArrayRank = x.Rank**  **n = x.GetLength(ArrayRank - 2) - 1**  **m = x.GetLength(ArrayRank - 1) - 1**  **ReDim y(m,n)**  **For i = 0 To n**  **For j = 0 To m**  **y(j,i) = x(i,j)**  **Next j**  **Next i**  **End Sub** |

*Рис. 4.8.2-9. Программный код процедуры* **Pr829()**

**Примера 4.8.2-9**

**Пример 4.8.2-10.Написать процедуру, которая вычисляет алгебраическое произ­ведение матрицы х(n ,m ) на матрицу y(m,n).**

Алгоритм и процедура решения задачи представлены на рис. 4.8.2-10.

Произведением матрицы **x(n,m)** на матрицу **y(m,k)** называется матрица **p(m,k)**, в которой элемент, стоящий на пересечении **i**-й строки и **j**-го столбца, равен произведению **i**-го вектора-строки матрицы **x(n,m)** на **j**-й вектор-столбец матрицы **y(m,k).**

Произведением вектора-строки **a(n)** на вектор-столбец **b** с одинаковым числом элементов **n** называется сумма произведений соответствующих элементов этих векторов . Таким образом, произведение двух матриц можно записать следующим образом:



|  |  |
| --- | --- |
|  | **Sub Pr8210(ByRef x(,) As Single \_**  **ByRef y(,) As Single \_**  **ByRef p(,) As Single)**  **Dim nx, mx, ky, j, i, k As Integer**  **Dim XRank, YRank As Integer**  **Dim s As Single**  **XRank = x.Rank : YRank = y.Rank**  **mx = x.GetLength(0)- 1**  **nx = x.GetLength(1)- 1**  **ny = y.GetLength(0)- 1**  **ky = y.GetLength(1)- 1**  **If XRank=2 And YRank=2 And nx=ny Then**  **ReDim p(mx,ky)**  **For i = 0 To mx**  **For j = 0 To ky**  **s = 0**  **For k = 0 To nx**  **S = s + x(i,k) \* y(k,j)**  **Next k**  **p(i,j)= s**  **Next j**  **Next i**  **vivodSngMac19(p, ListBox1)**  **Else**  **TextBox1.Text="Проверь размерности"**  **End If**  **End Sub** |

*Рис. 4.8.2-10. Программный код процедуры* **Pr8210()**

**Примера 4.8.2-10**

Для формирования такой матрицы необходимо организовать вложенные циклы: внешний цикл, изменяющий номер строки от **0** до **m**, и внутренний цикл, изменяющий номер столбца от **0** до **k**. Для вычисления произведения текущей строки матрицы **x(i,t)** на текущий столбец матрицы **y(t,j)** в теле внутреннего цикла требуется еще один цикл, параметр которого меняется от **0** до **n**.

На рис. 4.8.2-10 представлен общий алгоритм вычисления алгебраического произведения двух матриц размерами (mx\*nx) и   
(nx\*ky). В программу добавлена проверка на соответствие количества столбцов первой матрицы количеству строк второй матрицы, причем в схеме алгоритма она отсутствует.

Пример 4.8.2-11. Создать проект, в котором формируется двумерный массив x, состоящий из четырех строк и четырех столбцов, по правилу: x(i,j)=2\*(i-2)2-(j-2)2, а затем определяется максимальный элемент среди элементов, лежащих ниже главной диагонали, а также минимальный элемент, расположенный выше главной диагонали. Найденные максимальный и минимальный элементы необходимо поменять местами.

Формирование значений элементов двумерного массива **x(,)** по заданной формуле и их вывод в виде матрицы производятся с помощью вложенных циклов. В переменных **xmin** и **xmax** в цикле фиксируются минимальный и мак­симальный элементы массива, в переменных **imin**, **jmin** и **imax**, **jmax** – их координаты (номер строки и номер столбца).

Программный код проекта представлен на рис. 4.8.2-11.

|  |
| --- |
| **Option Strict On**  **Option Explicit On**  **Public Class Form1**  *'Формирование двумерного массива x(,)*  **Sub Form(ByRef x(,) As Single)**  **Dim i, j, m As Integer**  **m = x.GetLength(0) - 1**  **For i = 0 To m**  **For j = 0 To m**  **x(i, j) = CSng(2 \* (i - 2) ^ 2 - (j - 2) ^ 2)**  **Next**  **Next**  **End Sub**      *'Процедура форматного вывода двумерного массива в ListBox*  **Sub vivod1(ByRef x(,) As Single, ByRef LB As ListBox)**  **Dim i, j, m, n As Integer**  **Dim z, z1 As String**  **m = x.GetLength(0) – 1 : n = x.GetLength(1) - 1**  **LB.Items.Clear()**  **For i = 0 To m**  **z = ""**  **For j = 0 To n**  **z1 = Format(x(i, j),** "0.000")  **If x(i, j) < 0 Then**  **z1 = Space(2) + z1**  **Else**  **z1 = Space(3) + z1**  **End If**  **z = z + z1**  **Next**  **LB.Items.Add(z)**  **Next**  **End Sub**  *'Процедура вывода результата в TextBox*  **Sub vivod3(ByVal Z As Double, ByRef T As TextBox)**  **T.Text = CStr(Z)**  **End Sub**  *'Процедура вывода данных типа Integer в TextBox*  **Sub vivod4(ByVal Z As Integer, ByVal T As TextBox)**  **T.Text = CStr(Z)**  **End Sub**  *'Нахождение max элемента и его индексов,*  *'лежащих ниже главной диагонали*  **Sub MaxN(ByRef x(,) As Single, ByRef xmax As Single, \_**  **ByRef imax As Integer, ByRef jmax As Integer)**  **Dim i, j, m As Integer**  **m = x.GetLength(0) - 1**  **imax = 1 : jmax = 0 : xmax = x(1, 0)**  **For i = 0 To m**  **For j = 0 To i - 1**  **If x(i, j) > xmax Then**  **xmax = x(i, j) : imax = i : jmax = j**  **End If**  **Next**  **Next**  **End Sub**  *'Нахождение min элемента и его индексов,*  *'лежащих выше главной диагонали*  **Sub MinV(ByRef x(,) As Single, ByRef xmin As Single, \_**  **ByRef imin As Integer, ByRef jmin As Integer)**  **Dim i, j, m As Integer**  **m = x.GetLength(0) - 1**  **imin = 0 : jmin = 1 : xmin = x(0, 1)**  **For i = 0 To m**  **For j = i + 1 To m**  **If x(i, j) < xmin Then**  **xmin = x(i, j) : imin = i : jmin = j**  **End If**  **Next**  **Next**  **End Sub**  *'Обмен max и min элементов*  **Sub Ob(ByRef x(,) As Single, \_**  **ByVal imax As Integer, ByVal jmax As Integer, \_**  **ByVal imin As Integer, ByVal jmin As Integer)**  **Dim r As Single**  **r = x(imax, jmax) : x(imax, jmax) = x(imin, jmin)**  **x(imin, jmin) = r**  **End Sub**  *'Основная процедура*  **Sub Pr8211(ByRef x(,) As Single, ByRef xmin As Single, \_**  **ByRef xmax As Single)**  **Dim imin, jmin, imax, jmax As Integer**  **Form(x)** *'Формирование двумерного массива x(,)*  **vivod1(x, ListBox1)**  **MaxN(x, xmax, imax, jmax)**  **vivod3(xmax, TextBox1)**  **vivod4(imax, TextBox2)**  **vivod4(jmax, TextBox3)**  **MinV(x, xmin, imin, jmin)**  **vivod3(xmin, TextBox4)**  **vivod4(imin, TextBox5)**  **vivod4(jmin, TextBox6)**  **Ob(x, imax, jmax, imin, jmin)**  **vivod1(x, ListBox2)** *'Вывод преобразованного массива*  **End Sub**    **Private Sub Button1\_Click(sender As Object, \_**  **e As EventArgs) Handles Buton1.Click**  **Dim xx(3, 3) As Single**  **Dim xm1, xm2 As Single**  **Pr8211(xx, xm1, xm2)**  **End Sub**  **End Class** |

*Рис. 4.8.2-11. Программный код проекта* **Пример 4.8.2-11**

### 4.8.3. Тестовые задания

### 

1. **Двумерный массив – это массив, у которого количество индексов равно**
2. 2
3. 4
4. 3
5. любому целому числу
6. **Индексами двумерного массива могут быть:**
7. любые целочисленные выражения;
8. выражения любого типа
9. только целочисленные переменные
10. переменные любого типа
11. **Индексы в двумерном массиве указывают**
12. первый - номер строки, второй – номер столбца
13. первый - номер столбца, второй – номер строки
14. не имеет значения
15. все зависит от программиста
16. **Укажите размер данного двумерного массива A(2,3)**
17. 6
18. 12
19. 2
20. **Ввод и обработка двумерных массивов производится при использовании**
21. вложенных разветвлений
22. вложенных процедур
23. вложенных циклов
24. любых конструкций языка программирования
25. **Значениями параметров цикла в операторе For при вводе, обработке и выводе массивов являются**
26. значения элементов массива
27. максимальные значения индексов массива
28. номера строк и столбцов
29. минимальные значения индексов массива
30. **Если данные представлены в табличной форме, то для их обработки удобнее использовать**
31. одномерный массив
32. двумерный массив
33. трехмерный массив
34. любой массив
35. **Первый индекс в двумерном массиве указывает**
36. номер столбца
37. номер строки
38. не имеет значения
39. все зависит от программиста
40. **При вводе и обработке массива по столбцам внешний цикл должен быть организован**
41. по любому из индексов
42. по второму индексу
43. по первому индексу
44. все зависит от программиста
45. **Определите, как будет отображен двумерный массив после выполнения следующего фрагмента программы (2)**

|  |
| --- |
| **For i = 0 To 7**  **For j = 0 To 5**  **TextBox1.Text= TextBox1.Text &a(i) & ” ”**  **Next j**  **TextBox1.Text= TextBox1.Text & vbCrLf**  **Next i** |

1. в строку
2. в виде таблицы
3. в столбец
4. будет выдано сообщение об ошибке
5. **Dim A (3,5) As Single**
6. статический
7. динамический
8. фактический
9. вещественный
10. **После выполнения программы на форму будет выведено значение**

|  |
| --- |
| **Dim a (0 To 3,0 To 4), i, j As Single**  **For i = 0 To 3**  **For j = 0 To 4**  **a(i,j) = i\*j**  **Next j**  **Next i**  **TextBox1.Text = CStr(a(i,j))** |

1. сообщение об ошибке
2. 12
3. 20
4. 0
5. **Фрагмент программы выполняет алгоритм (2)**

|  |
| --- |
| **For i =0 To 4**  **r = a(i,2) : a(i,2) = a(i,4) : a(i,4) = r**  **Next i** |

1. обмена элементов главной и побочной диагоналей
2. присваивания 2-ому и 4-ому столбцам одинаковых значений
3. обмена 2-го и 4-го столбцов матрицы
4. обмен 2-й и 4-й строк матрицы
5. **Фрагмент программы выполняет операцию (2)**

|  |
| --- |
| **s = 0**  **For i = 0 To 5**  **For j = 0 To i -1**  **s = s + a(i,j)**  **Next j**  **Next i**  **TextBox1.Text =”s=” & CStr(s)** |

1. вывода суммы элементов ниже главной диагонали
2. вывода суммы элементов выше главной диагонали
3. вывода суммы элементов главной диагонали
4. вывода суммы элементов побочной диагонали
5. **Фрагмент программы выполняет операцию (2)**

|  |
| --- |
| **For i = 0 To 5**  **For j = 0 To 5**  **If j Mod 2 = 0 Then**  **a(i,j) = 0**  **End If**  **Next j**  **Next i** |

1. заполнения четных строк нулями
2. заполнения нулями четных элементов матрицы
3. заполнения нулями элементов четных столбцов
4. обнуления элементов побочной диагонали

## 4.8.4. Лабораторная работа по теме «Программирование алгоритмов формирования и обработки двумерных массивов»

**Цель работы:** овладение практическими навыками разработки и программирования вычислительного процесса, связанного с программированием алгоритмов ввода, заполнения и вывода двумерных массивов, классических приемов обработки двумерных массивов.

### Вопросы, подлежащие изучению

1. Способы описания двумерных массивов.
2. Способы задания значений элементам двумерного массива: присваиванием; инициализацией; вводом по запросу с клавиатуры; заполнением массива данных случайными числами в заданном диапазоне.
3. Использование вложенных циклов для ввода и вывода элементов двумерного массива.
4. Методы обработки элементов двумерного массива по строкам и по столбцам.

### Общее задание на разработку проекта

1. ***Изучите вопросы программирование алгоритмов формирования и обработки двумерных массивов*** *(Тема 8).*
2. ***Создайте приложение*** *с именем* ***Проект-8.***
3. ***Выберите вариант задания*** *из таблицы табл. 4.8.4-1.*
4. ***Разработайте графический интерфейс*** *пользователя.*
5. ***Разработайте схемы алгоритмов процедур*** *пользователя в соответствии с индивидуальным заданием, предварительно проведя формализацию.*
6. ***Напишите программный код процедур*** *в соответствии с разработанными схемами алгоритмов.*
7. ***Разработайте проект приложения****, решающий поставленную задачу, который состоит из интерфейса пользователя и соответствующего программного кода, а также написанных ранее процедур обработки и процедур ввода и вывода элементов массива. Все пользовательские процедуры должны находиться в модуле формы. Обмен данными между пользовательскими процедурами должен осуществляться через параметры, без использования глобальных переменных.*
8. ***Подготовьте массивы чисел для решения задачи****, если исходные данные не заданы.*
9. ***Выполните приложение и получите результат****.*
10. ***Докажите правильность результата****.*

### Варианты индивидуальных заданий

Таблица 4.8.4-1

|  |  |
| --- | --- |
| **№** | **Задача** |
| **1)** | Сформируйте произвольный двумерный массив N(10,10). Из положительных элементов массива N сформировать массив M(10,max), где max – максимальное число положительных элементов строки массива N, располагая их в строках массива M подряд. Запишите нули на место отсутствующих элементов массива M. Выведите полученный массив. |
| **2)** | Сформируйте двумерный массив L(10,8) по правилу:  где r – случайное число из отрезка [0;1]. Поменяйте местами левую и правую половины этого массива. Выведите полученный массив. |
| **3)** | Сформируйте произвольный двумерный массив L(9,9). Найдите в каждой строке наибольший элемент и поменяйте его местами с элементом главной диагонали. Выведите полученный массив. |
| **4)** | Сформируйте произвольный двумерный массив N(12,12) по правилу: Вычислите и выведите сумму и количество положительных элементов массива, находящихся под главной диагональю. Преобразуйте матрицу, уменьшив каждый ее элемент на найденную сумму. |
| **5)** | Сформируйте двумерный массив N(15,10) из случайных чисел, принадлежащих отрезку [-5;10]. Найдите строки с наибольшей и наименьшей суммой элементов. Выведите найденные строки и суммы их элементов. |
| **6)** | Сформируйте двумерный массив K(8,8) по правилу:  Kij=8∙(i - j)+j. Транспонируйте массив (поменяйте местами строки и столбцы), и выведите элементы главной диагонали и диагонали, расположенной под главной, разместив их в двух строках экрана |
| **7)** | Сформируйте двумерный массив K(6,3**)** по правилу  , где r – случайное число из отрезка [0;1]. Получите и выведите массив L(6,3), получаемый из массива K перестановкой строк: первой с последней, второй с предпоследней и т.д. |
| **8)** | Сформируйте произвольный двумерный массив N(12,12)**.** Вычислите и выведите сумму и кол. положительных элементов массива, находящихся под побочной диагональю. Преобразуйте матрицу, увеличив каждый ее элемент на найденную сумму. |
| **9)** | Сформируйте произвольный двумерный массив M(8,8). Найдите в каждой строке массива максимальный и минимальный элементы и поменять их местами соответственно с первым и последним элементами строки. Выведите полученный массив. |
| **10)** | Сформируйте двумерный массив K(10,12**)** из случайных чисел, принадлежащих отрезку [-5;10]. Определите в каждом столбце количество простых чисел и запишите их в соответствующий элемент одномерного массива L**.** Выведите массив L. |
| **11)** | Сформируйте произвольный двумерный массив L(7,9). Поменяйте местами левую верхнюю и правую нижнюю четверти матрицы. Выведите полученный массив. |
| **12)** | Сформируйте двумерный массив М(4,5) по правилу:  , где r – случайное число из отрезка [0;1]**.** Вставьте после столбца с заданным номером k (0<= k <=5) столбец из единиц (значение k вводится с клавиатуры). Выведите полученный массив M**.** |
| **13)** | Сформируйте двумерный массив t(4,4**)** по правилу:  Удалите из него столбцы, содержащие элементы, меньше, чем число 10**.** |
| **14)** | Сформируйте произвольный двумерный массив Z(10,4). Вычислите максимальное значение суммы модулей элементов в столбцах массива и выведите этот столбец. |
| **15)** | Сформируйте двумерный массив A(9,3) по правилу: . Определите наименьший элемент в каждой строке массива и запишите его в соответствующий элемент одномерного массива B. Выведите массив B. |
| **16)** | Сформируйте двумерный массив M(6,5) из случайных чисел, принадлежащих отрезку [-15;5]. Преобразуйте массив перестановкой столбцов: первого с последним, второго с предпоследним и т.д. |
| **17)** | Сформируйте произвольный двумерный массив B(4,4). Преобразуйте матрицу, перестановкой в каждой строке наибольшего по абсолютной величине элемента с диагональным. |
| **18)** | Сформируйте произвольный двумерный массив N(10,10). Получите и выведите одномерный массив K(10), где Ki **–** наименьшее из значений элементов, находящихся в начале i-ой строки массива N до элемента, принадлежащего главной диагонали включительно. |
| **19)** | Сформируйте произвольный двумерный массив X(4,4). Получите и выведите одномерный массив Y(4), где Yi – значение первого по порядку положительного элемента i-ой строки; если такого элемента нет, то примите Yi=1. |
| **20)** | Сформируйте двумерный массив Z(10,3) из случайных чисел, принадлежащих отрезку [-2;15]. Получите и выведите одномерный массив P, где Pi – сумма элементов, расположенных за первым отрицательным элементом в i**-**й строке; если все элементы строки неотрицательны, то примите Pi=100**.** |
| **21)** | Сформируйте произвольный двумерный массив F(10,3). Получите и выведите одномерный массив R**,** где Rj – сумма элементов, предшествующих последнему отрицательному элементу j-го столбца; если все элементы столбца неотрицательны, то примите Rj= -1. |
| **22)** | Сформируйте произвольный двумерный массив R(8,3**)** Найдите и выведите значение и индексы элемента, являющегося одновременно наименьшим в своей строке и наибольшим в своем столбце. При отсутствии такого элемента выведите сообщение. |
| **23)** | Сформируйте двумерный массив N(5,7) по правилу:    Вставьте после строки с заданным номером k (0<= k <=5) строку из нулей (значение k вводится с клавиатуры). |
| **24)** | Сформируйте произвольный двумерный массив M(5,7) Поменяйте местами строки, содержащие максимальный и минимальный элементы. Если минимум и максимум принадлежат одной строке, то поменяйте местами соответствующие столбцы. |
| **25)** | Сформируйте произвольный двумерный массив K(8,4)**.** Преобразуйте массив путем удаления из массива K строки и столбца, на пересечении которых находится наибольший элемент. |
| **26)** | Сформируйте двумерный массив M(7,7) по правилу:  , где r – случайное число из отрезка [0;1].  Введите одномерный массив K = {-5, -3, -1, 1, 3}. Замените нулями в массиве M те элементы, для которых имеются равные среди элементов массива K. Выведите полученный массив M. |
| **27)** | Сформируйте произвольные двумерные массивы K(3,3) и L(3,3). Получите массив M путем прибавления к элементам каждого столбца массива K произведения элементов соответствующей строки массива L. Выведите массив M. |
| **28)** | Сформируйте двумерный массив X(4,4) по правилу:  . Вставьте столбец из единиц перед столбцом, содержащим минимальный элемент матрицы. |
| **29)** | Сформируйте двумерный массив K(7,8) по правилу:  где r – случайное число из отрезка [0;1]. Вставьте строку из нулей перед строкой, содержащей максимальный элемент матрицы. |
| **30)** | Сформируйте произвольные двумерные массивы A(4,4) и B(4,4**)**. Получите массив R(4,4) путем умножения элементов каждой строки массива A на наибольший из элементов соответствующей строки массива B**.** Выведите массив R. |

### Содержание отчёта

1. Тема и название лабораторной работы.
2. Фамилия, имя студента, номер группы, номер варианта.
3. Задание на разработку проекта.
4. Формализация и уточнение задания.
5. Элементы, разрабатываемого проекта:
6. графический интерфейс пользователя;
7. таблица свойств объектов;
8. схема алгоритма решаемой задачи;
9. программный код проекта.
10. Результаты выполнения проектов.
11. Доказательство правильности работы программы.

### Пример выполнения задания

1. **Тема и название лабораторной работы:**

Программирование алгоритмов формирования и обработки двумерных массивов.

Заполнение и преобразование двумерного массива различными   
 способами.

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

Иванов И., БИН1405, вариант 13.

1. **Задание на разработку проекта:**

Введите двумерный массив натуральных чисел **a(m,n)** и, используя различные алгоритмы заполнения массива, сформируйте его. Обработайте сформированный массив, используя различные методы, и вычислите одну из характеристик сформированного массива.

1. **Формализация и уточнение задания:**

В разрабатываемом приложении ввод исходного массива может осуществляться одним из следующих способов:

* вводом с клавиатуры;
* с помощью датчика случайных чисел;
* формированием по формуле.

Обработка исходного массива осуществляться одним из вариантов:

* нахождение максимального элемента в двумерном массиве;
* нахождение суммы всех элементов двумерного массива;
* нахождение суммы элементов двумерного массива по строкам;
* нахождение суммы элементов двумерного массива по столбцам;
* нахождение суммы элементов главной диагонали двумерного массива;
* нахождение суммы элементов побочной диагонали двумерного массива.

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

Форма приложения может иметь такой же вид, как на рис. 4.8.4-1.



*Рис. П.1.8-1. Форма проекта* ***Проект 8:***

Заполнение и преобразование двумерного массива различными способами

1. Таблица свойств объектов:

Установите и сведите в табл. 4.8.4-2 свойства всех объектов.

Таблица. 4.8.4-2

|  |  |  |
| --- | --- | --- |
| **Имя**  **объекта** | **Свойство** | **Значение свойства** |
| **Form1** | Text | *Form1* |
| **Label3** | Font | *Times New Roman, Bold (Жирный), 14-point (14 пунктов)* |
| ForeColor | *Синий* |
| Name | *Label3* |
| Text | *Изучение двумерных массивов* |
| **Label4** | Font | *Times New Roman, Bold (Жирный), 10-point (10 пунктов)* |
| Name | *Label4* |
| Text | *Задание* |
| **Label5** | Font | *Times New Roman, Bold (Жирный), 10-point (10 пунктов)* |
| Name | *Label5* |
| Text | *Ввести двумерный массив A(m, n) и преобразовать массив A(m,n), используя меню* |
| **Label6** | Font | *Times New Roman, Bold (Жирный), 14-point (14 пунктов)* |
| ForeColor | *Синий* |
| Name | *Label6* |
| Text | *Исходный массив a(m,n)* |
| **Label7** | Font | *Times New Roman, Bold (Жирный), 14-point (14 пунктов)* |
| ForeColor | *Синий* |
| Name | *Label7* |
| Text | *Результат:* |
| **Label8** | ForeColor | *Red(Красный)* |
| Name | *Label8* |
| Text | *Задайте кол-во строк (m) и столбцов(n) массива* |
| **Label10** | Font | *10 пунктов* |
| Name | *Label10* |
| Text | *Заполнение массива* |
| **Label11** | AutoSize | *False* |
| ForeColor | *Синий* |
| Name | *Label11* |
| Text | *Преобразование массива* |
| **Label12** | Font | *Times New Roman, Bold (Жирный), 10-point (10 пунктов)* |
| Name | *Label12* |
| Text | *m=* |
| **Label13** | Font | *Times New Roman, Bold (Жирный), 10-point (10 пунктов)* |
| Name | *Label13* |
| Text | *n=* |
| **ComboBox1** | Items  (коллекция) | *Ввод массива с клавиатуры*  *Ввод массива по генератору случайных чисел*  *Ввод массива по формуле* |
| Name | *ComboBox1* |
| **ComboBox2** | Items  (коллекция) | *Найти max элемент в массиве*  *Найти сумму всех элементов*  *Найти сумму элементов по строкам*  *Найти сумму элементов по столбцам*  *Найти сумму элементов главной диагонали*  *Найти сумму элементов побочной диагонали* |
|  | Name | *ComboBox2* |
| **ListBox1** | Name | *ListBox1* |
| Text | *ListBox1* |
| **ListBox2** | Name | *ListBox2* |
| Text | *ListBox2* |
| **TextBox1** | Name | *TextBox1* |
| Text |  |
| **TextBox2** | Name | *TextBox2* |
| Text |  |
| **Button2** | Name | *Button2* |
| Text | *Выход* |
| **Button3** | Name | *Button3* |
| Text | *Повторить* |

1. Схема алгоритма решаемой задачи:

Схемы отдельных алгоритмов представлены в Теме 4.8.

1. Программный код проекта:

Программный код проекта приведен на рис. 4.8.4-2

|  |
| --- |
| **Public Class Form1**  **Dim a(,) As Integer**  *'Функция ввода данных (кол. строк и столбцов) из TextBox*  **Function vvod(ByVal T As TextBox) As Integer**  **Return CInt(T.Text)**  **End Function**  *'Процедура вывода одномерного массива с заголовком z*  **Sub PrintL(ByRef x() As Integer, ByVal z As String, \_**  **ByRef LB As ListBox)**  **Dim i As Integer**  **LB.Items.Add(" ")**  **LB.Items.Add(z)**  **Dim y As String = ""**  **For i = 0 To UBound(x)**  **y = y + CStr(x(i)) + Space(4)**  **Next i**  **LB.Items.Add(y)**  **End Sub**    *' Процедура вывода результата с заголовком z в ListBox*  **Sub vivodlist(ByVal c As Integer, ByVal z As String, \_**  **ByRef Lb As ListBox)**  **Lb.Items.Add(" ")**  **z = z + CStr(c)**  **Lb.Items.Add(z)**  **End Sub**  *'Процедура форматированного вывода матрицы в ListBox*  **Sub PrintMatr(ByRef x(,) As Integer, ByRef LB As ListBox)**  **Dim i, j, m, n As Integer**  **Dim z, z1 As String**  **m = x.GetLength(0) - 1**  **n = x.GetLength(1) - 1**  **LB.Items.Clear()**  **For i = 0 To m**  **z = ""**  **For j = 0 To n**  **z1 = Format(x(i, j), "000")**  **If x(i, j) < 0 Then**  **z1 = Space(2) + z1**  **Else**  **z1 = Space(3) + z1**  **End If**  **z = z + z1**  **Next j**  **LB.Items.Add(z)**  **Next i**  **End Sub**    *'Процедура ввода матрицы (по индексу выбранной*  *'строки из ComboBox1)*  **Sub vvodMatr(ByRef a(,) As Integer, \_**  **ByVal index As Integer)**  **Dim i, j, m, n As Integer**  **Dim r1, r2 As String**  **Dim y As String**  **m = a.GetLength(0) - 1**  **n = a.GetLength(1) - 1**  **Select Case index**  **Case 0**  *'Ввод элементов массива с клавиатуры*  **For i = 0 To m**  **For j = 0 To n**  **r1 = Str(i) : r2 = Str(j)**  **y = InputBox("элемент массива a(" + r1 + \_**  **"," + r2 + "=", \_**  **"Ввод эначений элементов массива A()")**  **a(i, j) = CInt(y)**  **Next j**  **Next i**  **Case 1**  *'Формирование массива с помощью случайных чисел*  **For i = 0 To m**  **For j = 0 To n**  **a(i, j) = CInt(Rnd(1) \* 100 - 50)**  **Next j**  **Next**  **Case 2**  *'Формирование массива по заданной формуле*  **For i = 0 To m**  **For j = 0 To n**  **a(i, j) = (i + j) \* 25 - 150**  **Next j**  **Next i**  **End Select**  **End Sub**    *' Вычисление суммы эл-тов побочной диагонали матрицы*  **Sub m5(ByRef a(,) As Integer, ByRef pob As Integer, \_**  **ByRef flag As Integer)**  **Dim RawCount, ColumnCount As Integer**  **Dim i, mm, nn As Integer**  **RawCount = a.GetLength(0)**  **ColumnCount = a.GetLength(1)**  **pob = 0**  **flag = 0**  **mm = RawCount - 1**  **nn = ColumnCount - 1**  **If nn = mm Then**  **For i = 0 To mm**  **pob = pob + a(i, mm - i)**  **Next i**  **Else**  **flag = 1 'количество строк и столбцов не равны**  **End If**  **End Sub**  *' Вычисление суммы эл-тов главной диагонали*  **Sub m4(ByRef a(,) As Integer, ByRef Glav As Integer, \_**  **ByRef flag As Integer)**  **Dim RawCount, ColumnCount As Integer**  **Dim i, nn, mm As Integer**  **RawCount = a.GetLength(0)**  **ColumnCount = a.GetLength(1)**  **Glav = 0**  **flag = 0**  **mm = RawCount - 1**  **nn = ColumnCount - 1**  **If nn = mm Then**  **For i = 0 To mm**  **Glav = Glav + a(i, i)**  **Next i**  **Else**  **flag = 1** *'количество строк и столбцов не равны*  **End If**  **End Sub**    'Вычисление сумм элементов матрицы по столбцам  ' (создание одномерного массива)  **Public Sub m3(ByRef a(,) As Integer, \_**  **ByRef stolb() As Integer)**  **Dim RawCount, ColumnCount As Integer**  **Dim i, j, nn, mm As Integer**  **RawCount = a.GetLength(0)**  **ColumnCount = a.GetLength(1)**  **mm = RawCount - 1**  **nn = ColumnCount - 1**  **ReDim stolb(0 To nn)**  **For j = 0 To nn**  **stolb(j) = 0**  **For i = 0 To mm**  **stolb(j) = stolb(j) + a(i, j)**  **Next i**  **Next j**  **End Sub**  ***' Вычисление сумм элементов матрицы по строкам***  ***' (создание одномерного массива)***  **Public Sub m2(ByRef a(,) As Integer, \_**  **ByRef strok() As Integer)**  **Dim RawCount, ColumnCount As Integer**  **Dim i, j, mm, nn As Integer**  **RawCount = a.GetLength(0)**  **ColumnCount = a.GetLength(1)**  **mm = RawCount - 1**  **nn = ColumnCount - 1**  **ReDim strok(0 To mm)**  **For i = 0 To mm**  **strok(i) = 0**  **For j = 0 To nn**  **strok(i) = strok(i) + a(i, j)**  **Next j**  **Next i**  **End Sub**    *' Вычисление суммы всех элементов матрицы*  **Public Sub m1(ByRef a(,) As Integer, ByRef sum As Integer)**  **Dim RawCount, ColumnCount As Integer**  **Dim i, j, mm, nn As Integer**  **RawCount = a.GetLength(0)**  **ColumnCount = a.GetLength(1)**  **sum = 0**  **mm = RawCount - 1**  **nn = ColumnCount - 1**  **For i = 0 To mm**  **For j = 0 To nn**  **sum = sum + a(i, j)**  **Next j**  **Next i**  **End Sub**  *' Поиск максимального элемента матрицы и его индексов*  **Public Sub m0(ByRef a(,) As Integer, ByRef max As Integer,\_**  **ByRef imax As Integer, ByRef jmax As Integer)**  **Dim RawCount, ColumnCount As Integer**  **Dim i, j, nn, mm As Integer**  **RawCount = a.GetLength(0)**  **ColumnCount = a.GetLength(1)**  **mm = RawCount - 1**  **nn = ColumnCount - 1**  **max = a(0, 0)**  **imax = 0**  **jmax = 0**  **For i = 0 To mm**  **For j = 0 To nn**  **If a(i, j) > max Then**  **max = a(i, j)**  **imax = i**  **jmax = j**  **End If**  **Next j**  **Next i**  **End Sub**    **Private Sub Button1\_Click(sender As Object, \_**  **e As EventArgs) Handles Button1.Click**  **ListBox1.Items.Clear()**  **ListBox2.Items.Clear()**  **TextBox1.Text = ""**  **TextBox2.Text = ""**  **End Sub**  **Private Sub ComboBox1\_SelectedIndexChanged \_**  **(sender As Object, e As EventArgs) \_**  **Handles ComboBox1.SelectedIndexChanged**  **Dim index, m, n As Integer**  **m = vvod(TextBox1)** *'кол-во строк массива*  **n = vvod(TextBox2)** *'кол-во столбцов массива*  **ReDim a(0 To m, 0 To n)**  **ListBox1.Items.Clear()**  **ListBox2.Items.Clear()**  **index = ComboBox1.SelectedIndex**  *'индекс выбранного элемента из ComboBox1*  *' (выбор способа ввода матрицы)*  **vvodMatr(a, index)**  **PrintMatr(a, ListBox1)**  **End Sub**  **Private Sub ComboBox2\_SelectedIndexChanged \_**  **(sender As Object, e As EventArgs) \_**  **Handles ComboBox2.SelectedIndexChanged**  **Dim s, f As Integer**  **Dim imax, jmax As Integer**  **Dim z As String = ""**  **Dim mas() As Integer**  **Select Case ComboBox2.SelectedIndex**  **Case 0 ' Поиск максимального элемента в матрице**  **m0(a, s, imax, jmax)**  **z = " Max элемент массива а()="**  **vivodlist(s, z, ListBox2)**  **z = " Номер строки элемента ="**  **vivodlist(imax, z, ListBox2)**  **z = " Номер столбца элемента ="**  **vivodlist(jmax, z, ListBox2)**  **Case 1** *' Вычисление суммы всех элементов массива*  **m1(a, s)**  **z = " Сумма элементов массива а()="**  **vivodlist(s, z, ListBox2)**  **Case 2** *' Вычисление сумм элем. массива по строкам*  **m2(a, mas)**  **z = " Сумма по строкам массива а()="**  **PrintL(mas, z, ListBox2)**  **Case 3** *' Вычисление сумм элем. массива по столбцам*  **m3(a, mas)**  **z = " Сумма элементов массива a() по столбцам "**  **PrintL(mas, z, ListBox2)**  **Case 4** *' Вычисление суммы элем.в главной диагонали*  **m4(a, s, f)**  **If f = 0 Then**  **z = " Сумма элементов главной диагонали ="**  **Else**  **z = "Кол. строк не равно кол. столбцов"**  **End If**  **vivodlist(s, z, ListBox2)**  **Case 5** *' Вычисление суммы элем. побочной диагонали*  **m5(a, s, f)**  **If f = 0 Then**  **z = " Сумма элем. побочной диагонали ="**  **Else**  **z = "Кол. строк не равно кол. столбцов"**  **End If**  **vivodlist(s, z, ListBox2)**  **End Select**  **End Sub**  **End Class** |

*Рис. 4.8.4-2. Программный код проекта* ***Проект 4.8:***

Заполнение и преобразование двумерного массива различными способами

1. **Результаты выполнения проектов.**

Результат выполнения проекта приведен на рис. 4.8.4-3.



*Рис. 4.8.4-3. Результат выполнения проекта* ***Проект 4.8:***

Заполнение и преобразование двумерного массива различными способами

1. **Доказательство правильности результата:**

Дана исходная матрица a(3,3):

21 3 8 -21

-20 27 -49 26

31 21 -45 -9

36 29 -13 46

Получим суммы элементов матрицы по столбцам:

21-20+31+36=68

3+27+21+29=80

8-49-45-13= -99

-21+26-9+46 = 42

Ручной расчет подтверждает правильность работы проекта для заданных исходных данных

### Контрольные вопросы

1. Что понимают в программировании под двумерным массивом данных?
2. В чем отличие ввода элементов двумерного массива по строкам и по столбцам?
3. Какие средства языка программирования применяются для вывода на экран двумерного массива по строкам?
4. Что может служить параметрами циклов в операторах For при вводе, обработке и выводе элементов массива?
5. Что такое динамический массив, и какие средства языка программирования используются для объявления динамического массива?
6. В чем отличие объявления одномерных и двумерных массивов?
7. С какого значения ведется нумерация индексов?
8. Как называется двумерный массив, у которого число индексов равно?
9. Что может использоваться в качестве индексов двумерного массива?
10. На что указывают индексы в двумерном массиве?
11. Какой размер двумерного массива A(2,3)?
12. Что необходимо сделать перед использованием двумерного массива?
13. На что указывает при объявлении двумерного массива диапазон изменения индексов?
14. Что такое размерность (ранг) массива?
15. На что указывает первый индекс в двумерном массиве?
16. Какой размер имеет двумерный массив A(2,3)?
17. На что указывает второй индекс в двумерном массиве?
18. С помощью какого оператора перед использованием двумерного массива в программе его необходимо объявить?
19. Как должен быть организован при вводе и обработке массива по строкам внешний цикл?
20. Как должен быть организован при вводе и обработке массива по столбцам внутренний цикл?
21. Какими методами можно определить параметры массива?
22. Чем отличаются функции **GetLength( )** и **GetUpperBound( )**?
23. Каким образом осуществляется доступ к каждому элементу двумерного массива при его обработке?
24. Поясните, как выполнить инициализацию двумерного массива.
25. Каким образом осуществить вывод двумерного массива на экран?
26. *Поясните, каким образом осуществляется обмен значений элементов двумер*ного массива.