Лабораторная работа №11
Исследование возможностей интегрированной среды разработки Visual C# для создания приложений по обработке многомерных массивов данных.

Цель работы – исследовать возможности интегрированной среды разработки Visual Studio 2010 и получить практические навыки по созданию приложений, обрабатывающие многомерные массивы данных.

1. Теоретические сведения

Массив (array) — это коллекция переменных одинакового типа (int, char, double), обращение к которым происходит с использованием общего для всех имени. В C# массивы могут быть одномерными или многомерными. Многомерным называется такой массив, который характеризуется двумя или более измерениями, а доступ к отдельному элементу осуществляется посредством двух или более индексов.

1.1. Двумерные массивы

Простейший многомерный массив — двумерный. В двумерном массиве позиция любого элемента определяется двумя индексами. Если представить двумерный массив в виде таблицы данных, то один индекс означает строку, а второй — столбец.

Чтобы объявить двумерный массив целочисленных значений размером 10x20 с именем **table**, достаточно записать следующие операторы:

```
int [ , ] table;
table = new int[10, 20];
```

или

```
int [ , ] table = new int[10, 20];
```

В C# значения размерностей отделяются запятыми. Первый оператор этого объявления означает, что создается ссылочная переменная двумерного массива. Второй оператор используется для выделения памяти для этого массива. Тем самым обеспечивается создание массива размером 10x20, причем значения размерностей также отделяются запятой.

Чтобы получить доступ к элементу двумерного массива, необходимо указать оба индекса, разделив их запятой. Например, чтобы присвоить число 10 элементу массива **table** позиция, которого определяется координатами 3 и 5, можно использовать следующий оператор:

```
table [3, 5] = 10;
```

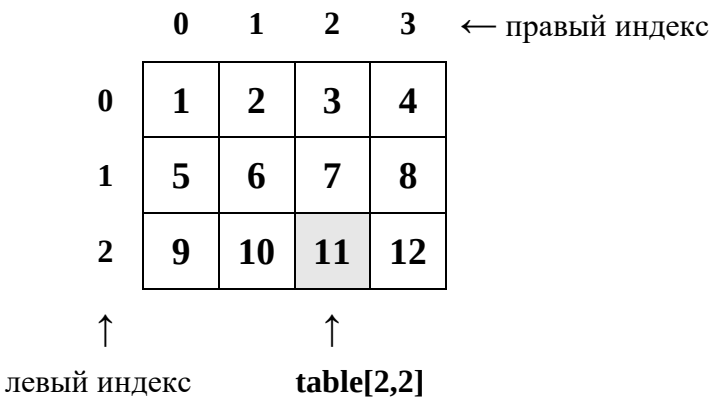
Пример 1. Нижеприведенный пример демонстрирует заполнение двумерного массива **table** числами от 1 до 12, а затем отображение содержимого этого массива.

```
public static void Main()
  {
    int t, i;
    int[,] table = new int[3, 4];
    for (t=0; t < 3; t++)
      {
        for (i=0; i < 4; i++)
          {
            table[t,i] = (t*4) + i + 1;
            Console.Write(table[t, i] + " " );
```

```
            }
          Console.WriteLine();
        }
    }
```

В этом примере элемент массива **table [0,0]** получит число **1**, элемент **table [0,1]** — число **2**, элемент table **[0,2]** — число **3** и т.д. Значение элемента table **[2,3]** будет равно **12**.

Схематично массив **table** можно представить в виде:



1.2. Массивы трех и более измерений

В C# можно определять массивы трех и более измерений. Вот как объявляется многомерный массив:

тип [, .. ,] имя = new тип[размер1, ..., размерN] ;

Например, с помощью следующего объявления создается трехмерный целочисленный массив размером 4x10x3:

int [, ,] multimas = new int [4 , 10, 3];

Чтобы присвоить число 100 элементу массива **multimas**, занимающему позицию с координатами 2,4,1, используйте такую инструкцию:

multidim[2, 4, 1] = 100;

Пример 2. Нижеприведенный пример демонстрирует использование трехмерного массива, содержащего 3*3*3 - матрицу значений.

```
public static void Main()
  {
   int[ , , ] m = new int[3, 3, 3];
   int sum = 0;
   int n = 1;
   for (int x=0; x < 3; x++)
      for (int y=0; y < 3; y++)
         for (int z=0; z < 3; z++)
            m[x, y, z] = n++;
   sum = m[0,0,0] + m [l, l, l] + m[2, 2, 2] ;
   Console.WriteLine("Сумма первой диагонали: " + sum);
  }
```

1.3. Инициализация многомерных массивов

Многомерный массив можно инициализировать, заключив список инициализаторов каждой размерности в собственный набор фигурных скобок.

Например, формат инициализации двумерного массива N*M имеет вид:

тип[,] имя_массива = {
{val_{11}, val_{12}, val_{13}, ..., val_{1m}},
{val_{21}, val_{22}, val_{23}, ..., val_{2m}},
. . .
{val_{n1}, val_{n2}, val_{n3}, ..., val_{nm}}
};

где элемент **val_{ij}** — значение инициализации. Каждый внутренний блок означает строку. В каждой строке первое значение будет сохранено в первой позиции массива, второе значение — во второй и т.д. Обратите внимание на то, что блоки инициализаторов отделяются запятыми, а точка с запятой становится только после закрывающей фигурной скобки.

Пример 3. Нижеприведенный пример демонстрирует инициализацию массива **kvadrat** числами от 1 до 5 и квадратами этих чисел.

```
public static void Main()
  {
   int[ , ] kvadrat = {
                        {1, 1},
                        {2, 4},
                        {3, 9},
                        {4, 16},
                        {5, 25},
                       };
   int i, j;
   for (i=0; i < 5; i++)
      {
       for (j=0; j < 2; j++) Console.Write(kvadrat [i,j] + " " ) ;
       Console.WriteLine();
      }
  }
```

2. Рабочее задание

Задание 1. Руководствуясь теоретическим материалом раздела 1 изучить возможности языка C# по созданию приложений, обрабатывающие многомерные массивы данных, и выполнить практически все примеры, описанные в этом разделе.

Задание 2. Разработать приложение с заголовком «Работа с матрицами», с помощью которого выполняется операция сложения матрицы **A** и транспонированной матрицей **B** с последующей записью результата в матрицу **A ($A = A + B^T$)**. Размерность матриц **A, B** и значения их элементов вводятся с клавиатуры. Исходные матрицы **A, B**, транспонированную матрицу **B** и результирующую матрицу вывести на экран монитора.

Задание 3. Разработать приложение с заголовком «Номер строки», которое определяет номер строки матрицы, сумма элементов которой максимальна. Размерность матрицы и значения её элементов вводятся с клавиатуры. Поиск суммы элементов строки оформить в виде отдельного метода с использованием модификатора **params**. По окончании выполнения приложения на экран монитора должны быть выведены исходная

матрица, суммы строк матрицы и номер строки, сумма элементов которой максимальна (см. рис.1).

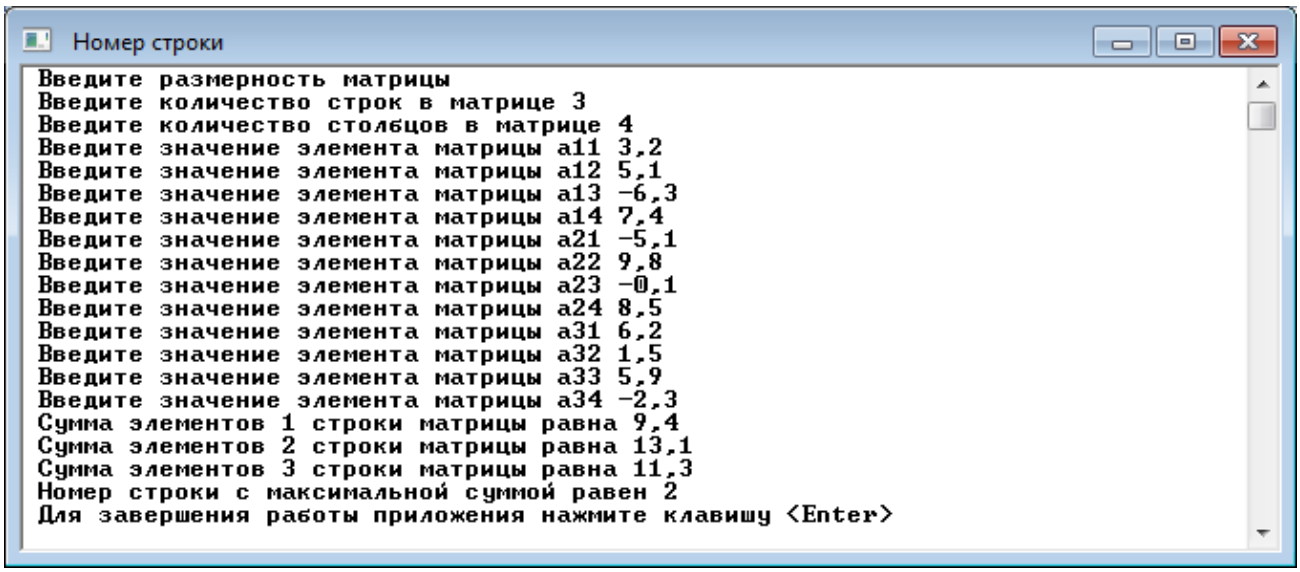


Рис. 1. Результат выполнения приложения «Номер строки»

3. Контрольные вопросы

1. Что такое многомерный массив?
2. Как объявляются двумерные массивы?
3. Как объявляются многомерные массивы?
4. Как осуществляется доступ к элементам многомерного массива?
5. Как инициализируются многомерные массивы?

Литература

1. Голощапов А.Л. Microsoft Visual Studio 2010. – СПб.:БХВ-Петербург, 2011. – 544 с.: ил.
2. Петцольд Ч. Программирование для Microsoft Windows на C#. В 2-х томах. Том 1. Пер. с англ. - М.: «Русская Редакция», 2002.- 576 с.: ил.
3. Петцольд Ч. Программирование для Microsoft Windows на C#. В 2-х томах. Том 2. Пер. с англ. - М.: «Русская Редакция», 2002.- 624 с.: ил.
4. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4.0. Пер. с англ. - М.: Издательский дом "Вильямс", 2011. — 1392 с.: ил.
5. Фленов М.Е. Библия C#. - СПб.: БХВ-Петербург, 2011. – 560с.: ил.
6. Шилдт Г. C# Учебный курс. – СПб.: Питер, Издательская группа BHV, 2003. – 512 с.: ил.