

Частное учреждение образования  
«Колледж бизнеса и права»

УТВЕРЖДАЮ  
Заместитель директора  
по учебной работе  
\_\_\_\_\_Голубков В.К.  
«\_\_»\_\_\_\_\_2012

Специальность 2-40 01 01: «Программное обеспечение ин- формационных технологий»	Дисциплина: «Базы данных и системы управления базами данных»
Составлена на основании учебной программы, утвержденной директором Колледжа бизнеса и права 25.11.2011	

## Лабораторная работа № 26

### Инструкционно-технологическая карта

Тема: массивы переменных

Цель работы: научиться разрабатывать программы с использованием массивов

Время выполнения: 2 часа

### Краткие теоретические сведения

Массив представляет собой переменную, для хранения значения которой отводится не одна (как для обычной переменной), а несколько ячеек памяти. Каждая ячейка отводится под отдельный элемент массива. Таким образом, массив является структурированной переменной в отличие от скалярной простой переменной. Все элементы массива имеют один и тот же тип. При необходимости выполнить некоторое действие над всеми элементами массива можно сослаться на массив как целое по его имени. Возможны также ссылки на его отдельные элементы. Каждый отдельный элемент массива определяется именем массива и значениями индексов. Например, ссылки  $a(7)$  или  $a1(2, 9)$  означают, что:

$a$  – это имя одномерного массива (вектора) с одним индексом, элемент массива имеет значение индекса равное 7;

$a1$  – это имя двумерного массива (матрицы), элемент массива принадлежит 2 строке и 9 столбцу.

Как и простые переменные, массивы описываются с помощью инструкций **Dim**, **Static**, **Private** или **Public**. Разница в объявлении между скалярными переменными (т.е. не массивами) и массивами состоит в том, что для массива надо указывать его размер (количество ячеек, отводимых под массив). Массив с заданным размером называется **фиксированным**. Массив, объявление размера которого отложено, называется **динамическим**.

### Объявление массива фиксированного размера

При объявлении фиксированного массива, кроме его имени и типа элементов, необходимо для каждого индекса указать его верхнюю границу, нижняя граница всегда равна 0.

В следующей строке программы массив фиксированного размера описывается как массив типа Integer, имеющий 11 строк и 11 столбцов:

```
Dim a1(10, 10) As Integer
```

Первый аргумент внутри скобок обозначает наибольший номер строки, а второй – наибольший номер столбца.

Как и при описании переменных, если тип при описании массива не задается, то массив будет иметь тип Object.

### Описание динамического массива

Если массив описан как динамический, можно изменять его размер во время работы программы. Для описания динамического массива используются инструкции **Static**, **Dim**, **Private** или **Public** с парой скобок, внутри которых помещаются запятые для обозначения размерности, если она больше 1:

```
Dim a2(), b(.) As Single
```

При выполнении программы (но только внутри подпрограмм и функций) можно **переобъявлять динамический массив с помощью инструкции ReDim**. Пусть, например, на уровне проекта или на уровне формы был объявлен динамический массив

```
a3: Dim a3(.) As Single
```

Тогда среди инструкций некоторой подпрограммы или функции можно записать:

```
ReDim a3(5, 10)
```

Далее этот массив может быть переопределен: `ReDim a3(7,15)`

С помощью инструкции **ReDim** можно изменять размер массива (число элементов), верхние границы индексов. Инструкцию **ReDim** можно применять для изменения динамического массива столько раз, сколько потребуется. Однако при каждом её применении данные, содержащиеся в массиве, теряются. Инструкция **ReDim Preserve** может увеличить размер массива, сохраняя при этом его содержимое. В следующем примере показывается, как можно увеличить размер массива `a4` на 10 элементов без уничтожения текущих значений элементов массива. Пусть был объявлен динамический массив:

```
Dim a4() As Integer
```

Затем в программе установлен размер этого массива: `ReDim a4(n)`

Далее в программе может быть, например, записано: `ReDim Preserve a4(n+10)`

Использование зарезервированных слов **ReDim Preserve** вместе с динамическим массивом позволяет изменить только верхнюю границу и только последней размерности массива.

### Некоторые полезные свойства и методы массивов

Пусть даны следующие массивы:

```
Dim a() As Integer = {80, 60, 50, 90, 40, 20, 50, 70}
```

```
Dim t(.) As Integer = {{99, 99, 99, 99, 99}, {99, -8, -14, -19, -18}, {99, 25, 28, 26, 20}, {99, 11, 18, 20, 25}}
```

Свойств и методы	Описание	Результат
<b>a.Length</b>	Длина массива <code>a</code> (число элементов)	8

Свойства и методы	Описание	Результат
<b>t.Length</b>	Длина массива t (число элементов)	20
<b>t.GetUpperBound (0)</b>	Число строк (макс. индекс первого измерения) - 1	3
<b>t.GetUpperBound(1)</b>	Число столбцов (макс. индекс второго измерения) - 1	4
<b>Array.IndexOf (a, 50)</b>	Ищется первое вхождение числа 50 в одномерный массив a и находится его индекс	2
<b>Array.LastIndexOf (a, 50)</b>	Ищется последнее вхождение числа 50 в одномерный массив a и находится его индекс	6
<b>Array.IndexOf(a, 55)</b>	Ищется число 55 в массиве a и не находится	-1
<b>Array.Reverse</b> Ошибка! Закладка не определена.(a)	Все элементы массива a меняют порядок на обратный	{70, 50, 20, 40, 90, 50, 60, 80}
<b>Array.Sort (a)</b>	Все элементы массива a сортируются по возрастанию	{20, 40, 50, 50, 60, 70, 80, 90}
<b>Array.Clear (a, 4, 3)</b>	Обнуляется 3 элемента массива a, начиная с индекса 4	{20, 40, 50, 50, 0, 0, 0, 90}

### Порядок выполнения работы

**1. Условие задачи.** При заданных значениях элементов векторов  $a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_m$  и некоторого порога  $h$  получить значения  $c_1, c_2, \dots, c_m$ , последовательно рассматривая тройки  $(h, a_i, b_i)$ . Элементам вектора  $c_i$  следует задать значение по правилу:

- $a_i$ , если  $a_i$  наибольшее из этих трех значений;
- $b_i$ , если  $b_i$  наибольшее из этих трех значений;
- $h$  при любых других соотношениях значений элементов в тройке  $(h, a_i, b_i)$

Также требуется определить по отдельности, сколько элементов вектора  $a$  и вектора  $b$  стали значениями элементов вектора  $c$ .

### 2. Выбор данных.

Исходные данные:

- $m$  – количество элементов векторов, переменная целого типа;
- $h$  – порог, переменная ординарной точности с дробной частью;
- $a, b$  – одномерные векторы, переменные ординарной точности с дробной частью.


Результаты:

- $c$  – искомый одномерный вектор, ординарной точности с дробной частью;
- $ka$  – количество элементов вектора  $a$ , вошедших в вектор  $c$ , переменная целого типа;
- $kb$  – количество элементов вектора  $b$ , вошедших в вектор  $c$ , переменная целого типа.

Промежуточные:

- $i$  – параметр цикла, переменная целого типа.

### 3. Выполнение примера на компьютере.

1. Создайте новый проект с именем Цикл\_с\_разветвлением.
2. Если окно конструктора форм не открыто, то откройте его щелчком на кнопке  **View Designer** (просмотреть конструктор), расположенной на панели инструментов окна обозревателя решений. Если же эта кнопка на панели инструментов отсутствует, то предварительно щелкните в окне обозревателя решений на компоненте проекта Form1.vb.
3. Для разработки интерфейса следуйте рисунку 1.

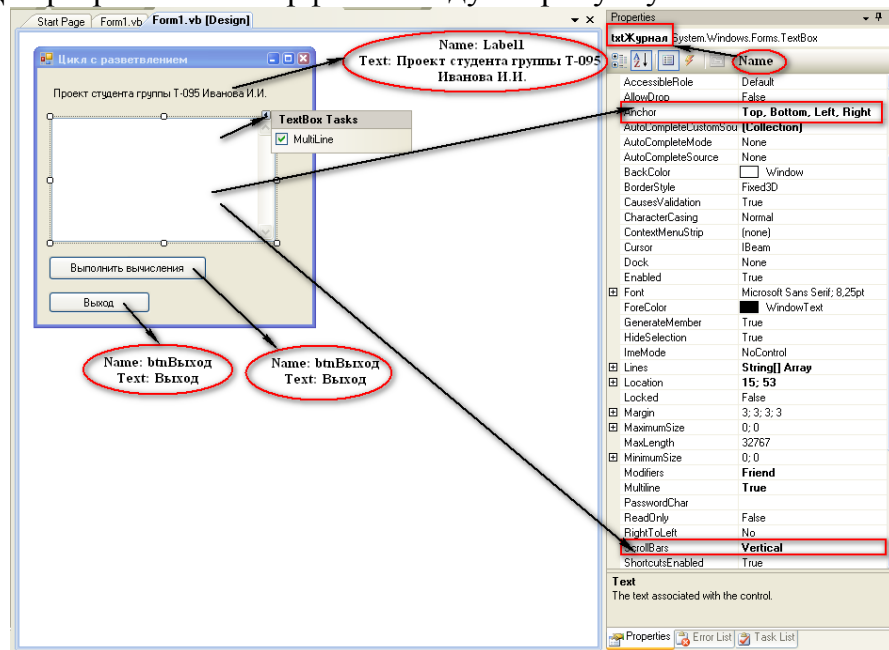


Рисунок 1. Разработка интерфейса

4. Двойным щелчком на кнопке **btnПуск** вставьте в программный код заготовку подпрограммы **btnПуск\_Click** – обработчика события, заключающегося в щелчке на кнопке **btnПуск**. Это событие будет для проекта командой выполнить требуемые вычисления.
5. Двойным щелчком на надписи **Label1** создайте заготовку подпрограммы **Label1\_Click**.
6. Введите код подпрограмм **btnПуск\_Click** и **Label1\_Click**.
7. Копируйте в проект код подпрограмм **InputVector** и **OutputVector**. В итоге код проекта должен соответствовать листингу 1.

#### Листинг 1. Код проекта

**Public Class Form1**

'Эта подпрограмма выполняется при каждом нажатии на кнопку, на которой написано **Вычислить**

**Private Sub** btnПуск\_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** btnПуск.Click

'Объявление простых переменных

**Dim** m, i, ka, kb **As** Integer

'Объявление трех динамических массива

**Dim** h, a(), b(), c() **As** Single

'Очистка текстового поля txtЖурнал

txtЖурнал.Clear()

'Ввод значений и контрольный вывод переменных m и h

```

m = InputBox("m = ?")
txtЖурнал.AppendText("m = " & m & vbCrLf)
h = InputBox("h = ?")
txtЖурнал.AppendText("h = " & h & vbCrLf)

```

'Завершение объявления трех одномерных динамических массивов и обеспечение их размещения в памяти

```
ReDim a(m), b(m), c(m)
```

'Обращение к подпрограмме InputVector, которая обеспечивает ввод всех элементов вектора a, начиная со значения a(1). Первый аргумент этой программы - имя вектора, который нужно ввести. Второй необязательный аргумент этой подпрограммы - строка символов, являющаяся именем вводимого массива. Действие этого обращения к подпрограмме следует понимать как выполнение тела подпрограммы InputVector, в котором имя x везде заменено именем a

```
InputVector(a, "a")
```

'Вывод в текстовое поле txtЖурнал строки Вектор a

```
txtЖурнал.AppendText("Вектор a" & vbCrLf)
```

'Обращение к подпрограмме OutputVector, которая обеспечивает вывод в текстовом поле txtЖурнал значений всех элементов вектора a, начиная со значения a(1)

```
OutputVector(a, txtЖурнал)
```

'Ввод и контрольный вывод значений всех элементов вектора b, начиная b(1)

```
InputVector(b, "b")
```

```
txtЖурнал.AppendText("Вектор b" & vbCrLf)
```

```
OutputVector(b, txtЖурнал)
```

'Цикл, в котором обеспечивается повторное разветвление

```
For i = 1 To m
```

'Повторное разветвление

```
If a(i) > b(i) And a(i) > h Then
```

```
    c(i) = a(i)
```

```
    ka = ka + 1
```

```
ElseIf b(i) > a(i) And b(i) > h Then
```

```
    c(i) = b(i)
```

```
    kb = kb + 1
```

```
Else : c(i) = h
```

```
End If
```

```
Next
```

'Вывод в текстовое поле txtЖурнал результатов вычислений

```
txtЖурнал.AppendText("ka = " & ka & vbTab)
```

```
txtЖурнал.AppendText("kb = " & kb & vbCrLf)
```

```
txtЖурнал.AppendText("Вектор c" & vbCrLf)
```

```
OutputVector(c, txtЖурнал)
```

```
End Sub
```

```
Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label1.Click
```

```
End
```

```
End Sub
```

'Подпрограмма InputVector обеспечивает ввод значений всех элементов вектора x, начиная с x(1) по последний. Вторым аргументом этой функции типа строки символов не является обязательным. Он передает функции InputBox имя вводимого

вектора, что позволяет сделать операцию ввода нагляднее. Номер последнего элемента массива  $x$  возвращает метод массива `GetUpperBound(0)`. Его аргумент, равный 0, это индекс первой размерности массива. Размерность массива нумеруется, начиная с нуля (0, 1, 2, ...)

```
Private Sub InputVector(ByRef x() As Single, _
Optional ByVal ArrayName As String = "элемент")
    Dim i As Integer
    For i = 1 To x.GetUpperBound(0)
        x(i) = InputBox(ArrayName & "(" & i & ") = ?")
    Next
End Sub
```

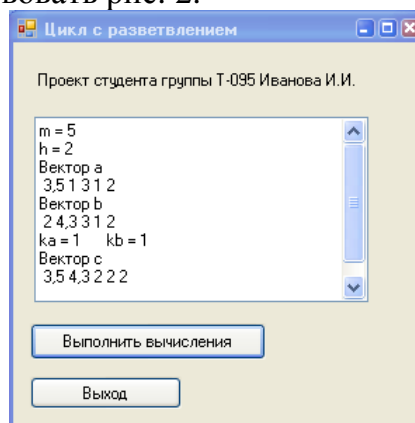
Подпрограмма `OutputVector` обеспечивает вывод всех элементов вектора  $x$ , начиная с  $x(1)$  по последний, в заданное значение аргумента `txtBx` текстовое поле. Номер последнего элемента массива возвращает метод массива `GetUpperBound(0)`.

```
Private Sub OutputVector(ByRef x() As Single, ByVal txtBx As TextBox)
    Dim i As Integer
    For i = 1 To x.GetUpperBound(0)
        txtBx.AppendText(" " & x(i))
    Next
    txtBx.AppendText(vbCrLf)
End Sub
End Class
```

8. Запустите проект. Проверьте его работу при значениях исходных данных:

$$\begin{aligned} m &= 5, h = 2, \\ a &= (3,5; 1; 3; 1; 2), \\ b &= (2; 4,3; 3; 1; 2). \end{aligned}$$

Если при разработке проекта не были допущены ошибки, то результат выполнения программы должен соответствовать рис. 2.



**Рис. 2 Результат выполнения проекта**

9. Замените код программы `btnПуск` кодом, составленным Вами для решения заданного индивидуального варианта задания.

10. Сохраните проект.

11. Выполните отладку и тестирование проекта. Устраните обнаруженные ошибки.

12. Покажите полученный результат преподавателю.

13. Попробуйте ответить на вопросы для контроля.

### Контрольные вопросы

1. В чем отличие массива от переменной?
2. Могут ли элементы массива иметь разный тип?
3. Что означает инструкция Dim g(5) As String?
4. Что означает инструкция Dim g() As String?
5. Что означает инструкция Dim g(.) As String?
6. Чем отличается фиксированный массив от динамического массива?
7. В какой последовательности выполняются операции в выражении  

$$a+b/c*d-e^2?$$
8. Имеется квадрат со стороной  $h$  и круг диаметра  $d$  ( $d < h$ ). Центры обеих фигур расположены в начале координат. Запишите логическое выражение, которое принимает значение True только в том случае, когда точка с заданными координатами  $(x, y)$  принадлежит квадрату, но не принадлежит кругу.
9. Что означает инструкция ReDim g(n)?
10. Для чего применяются обращение к методу GetUpperBound(0)?
11. Что определяют значения свойств надписи: Text, TextAlign?
12. Что определяют значения свойств кнопки: Name, Text, Font?
13. Что определяют значения свойств текстового поля: Name, Text, Font, Multiline, ReadOnly, Anchor, ScrollBars?

Преподаватель

С.В. Бурко

Рассмотрено на заседании цикловой  
 комиссии программного обеспечения  
 информационных технологий №10  
 Протокол № от « » \_\_\_\_\_ 201\_  
 Председатель ЦК \_\_\_\_\_ С.В. Бурко