

## ЛЕКЦИЯ 9

### ТЕМА: МАССИВЫ В VB

1. **Массивы в VB**
2. **Объявление массива в VB**
3. **Объявление динамического массива**

**1. Массивы в VB.** Для хранения величин кроме простых переменных можно использовать массивы. *Массив* представляет собой набор переменных с одним именем и разными индексами. Каждая такая переменная называется *элементом массива*. Количество хранящихся в массиве элементов называется *размером массива*. Размер массива ограничен объемом оперативной памяти и типом данных элементов массива.

Все элементы массива имеют одинаковый тип. Однако если массив имеет тип `variant`, то отдельные элементы могут содержать данные разных типов. Например, одни элементы могут быть числами, другие — строками или объектами.

Индекс элемента указывается в круглых скобках после имени массива. Например, `strNames(1)`, `strNames(2)`, `strNames(10)` ЯВЛЯЮТСЯ элементами массива с именем `strNames`. Вы можете использовать каждый из элементов массива точно так же, как и простую переменную.

#### 1. **Объявление массива в VB**

В Visual Basic существуют массивы фиксированного размера и динамические массивы. Массив фиксированного размера имеет неизменный размер, заданный при его объявлении. Динамические массивы могут изменять размер в процессе выполнения.

#### 2. **Объявление массива фиксированного размера**

Объявление массива фиксированного размера зависит от области его видимости и осуществляется следующим образом:

- глобальный массив объявляется с помощью оператора `public` в секции **Declaration** модуля:

- массив уровня модуля — с помощью оператора `private` в секции **Declaration** модуля;
- локальный массив — с помощью оператора `private` процедуры.

При объявлении массива после его имени в круглых скобках указывается верхняя граница массива. По умолчанию нижней границей массива является 0. Например, в приведенном ниже коде, который вводится в секцию **Declaration** модуля, задается массив из 21 элемента. Индекс элементов массива изменяется от 0 до 20:

```
Dim intCountPar (20) As Integer
```

Для создания глобального массива такого же размера необходимо использовать следующий код:

```
Public intCountPar (20) As Integer
```

Можно явно задать нижнюю границу массива, используя ключевое слово `to`:

```
Dim intCountPar (1 To 20) As Integer
```

В этом случае задается массив из 20 элементов. Индекс элементов массива изменяется от 1 до 20.

Visual Basic позволяет использовать многомерные массивы. Например, в следующем коде объявляется двумерный массив размерностью 21x21:

```
Dim intCountPar (20, 20) As Integer
```

При использовании многомерных массивов, как и в случае одномерных, можно явно задавать нижнюю границу:

```
Dim intCountPar (1 To 20, 1 To 20) As Integer Dim intCountPar (20, 1 To 20) As Integer
```

В верхней строке кода явно заданы верхняя и нижняя граница обеих размерностей. В нижней строке задана верхняя и нижняя граница только для второй размерности.

### 3. Объявление динамического массива

В случае, когда размер массива заранее неизвестен. Visual Basic позволяет использовать динамические массивы, размеры которых можно изменять во время выполнения. Применение динамических массивов

позволяет эффективно управлять памятью, выделяя память под большой массив лишь на то время, когда этот массив используется, а затем освобождая ее.

Создание динамического массива осуществляется следующим образом:

1. Объявляется массив с помощью ключевых слов, используемых при создании массива фиксированного размера. Список размерностей массива остается пустым. При объявлении глобального массива необходимо выбрать ключевое слово `Public`, при объявлении массива на уровне модуля — `Dim`, при объявлении массива в процедуре — `Dim` или `static`. Например,

```
Dim intCountPar O As Integer
```

2. С помощью выполняемого оператора `ReDim` указывается размерность массива в виде числа или выражения. Синтаксис оператора `ReDim` аналогичен синтаксису оператора объявления массива фиксированного размера. Например, размерность массива может быть задана любым из следующих способов:

```
ReDim intCountPar (x)
```

```
ReDim intCountPar (20)
```

```
ReDim intCountPar (1 To 20)
```

При выполнении оператора `ReDim` данные, размещенные в массиве ранее, теряются. Это удобно в том случае, если данные вам больше не нужны и вы хотите переопределить размерность массива и подготовить его для размещения новых данных. Если вы хотите изменить размер массива, не потеряв при этом данных, то необходимо воспользоваться оператором `ReDim` с ключевым словом `Preserve`. Например, приведенный ниже программный код увеличивает размер массива на единицу без потери хранящихся в массиве данных:

```
ReDim Preserve intCountPar (X + 1)
```

### **Замечание**

Использование оператора `ReDim` с ключевым словом `Preserve` позволяет изменять только верхнюю границу последней размерности многомерных размеров.