



Переменные
Visual Basic



Переменные

Переменная – зарезервированное место в оперативной памяти компьютера для временного хранения данных.

- Каждая переменная имеет собственное имя.
- После того, как переменной присвоено значение, она может использоваться в программе вместо самого значения.

Правила формирования имен переменных:

- содержит не более 255 символов,
- используются любые буквы, цифры и знаки подчеркивания,
- первый символ в имени должен быть буквой,
- нельзя использовать пробелы,
- имя должно быть уникальным.
- не может быть ключевым словом

Примеры: CurrentNum, Data_of_birth





Простые типы данных



- Byte
- Integer
- Long
- Single
- Double
- Currency

- Boolean

- String
- 
- 

Типы данных

Тип данных	Описание	Объем памяти	Диапазон значений
Integer	Целые числа	2 байта	От -32768 до 32767
Long	Целые числа	4 байта	От -2147483648 до 2147483647
Single	Десятичные числа с плавающей точкой	4 байта	От $-3,4^{38}$ до $-1,4^{-45}$ для отрицательных чисел и от $1,4^{-45}$ до $1,4^{38}$ для положительных чисел
Double	Десятичные числа с плавающей точкой	8 байта	От $-1,7^{308}$ до $-4,9^{-324}$ для отрицательных чисел и от $4,9^{-324}$ до $1,7^{308}$ для положительных
Currency	Числа с фиксированной точкой	8 байта	Целая часть числа до 15 цифр, дробная до 4

Типы данных

Тип данных	Описание	Объем памяти	Диапазон значений
String	Строки фиксированной (до 2^{16} символов) и переменной длины (до 2^{31} символов)	ASCII - 1 байт на символ UNICODE - 2 байта на символ + 10 байт для строки в целом	Строка дополняется пробелами или обрезается (при использовании фиксированной длины)
Date	Хранение даты и времени	8 байт	
Byte	Хранение двоичных данных	1 байт	
Boolean	Логический тип	2 байта	True False по умолчанию присваивается значение False
Variant	Универсальный тип (для любого типа данных)	Переменная длина не менее 16 байт	
Object	Хранение ссылок на объекты	4 байта	



Объявление переменных

Явное объявление переменных

Оператор:

`Dim имяПеременной [As типДанных]`

Примеры:

`Dim Name As String*15`

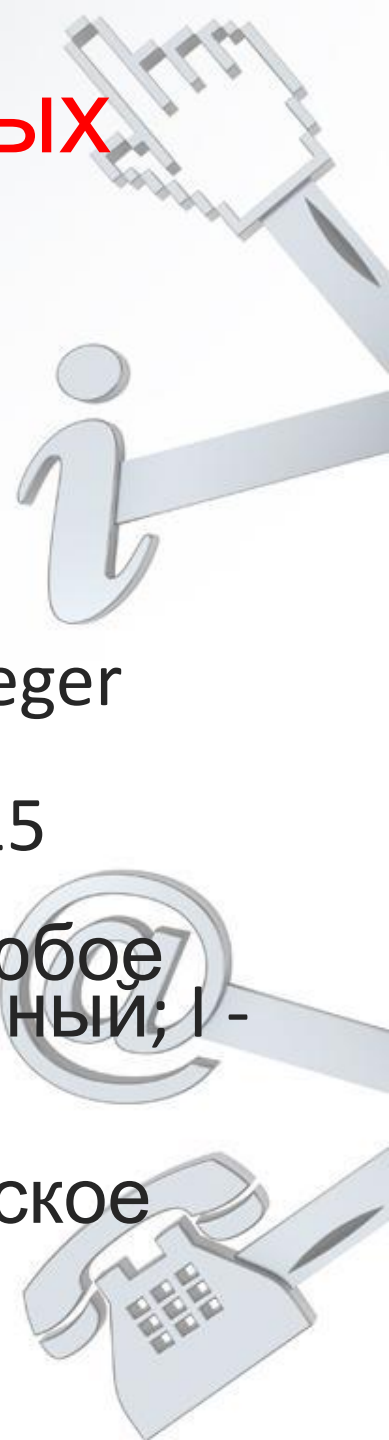
`Dim SurName As String, A As Single, I As Integer`

`Dim Money As Currency, Alfa As Boolean`

Name - строка фиксированной длины (15
СИМВ.);

SurName - строка переменной длины (любое
количество символов); A - вещественный; I -
целый;

Money - арифметическое данное для
коммерческих расчетов, Alfa - логическое



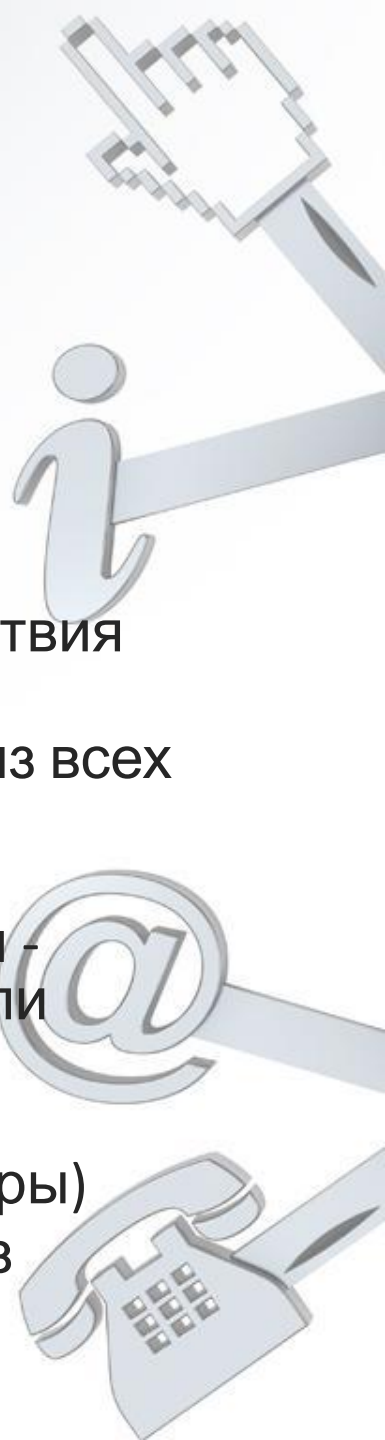


Область действия переменных

Private имяПеременной [As типДанных]
Static имяПеременной [As типДанных]
Public имяПеременной [As типДанных]

Dim, Private, Static, Public – определяют область действия переменной.

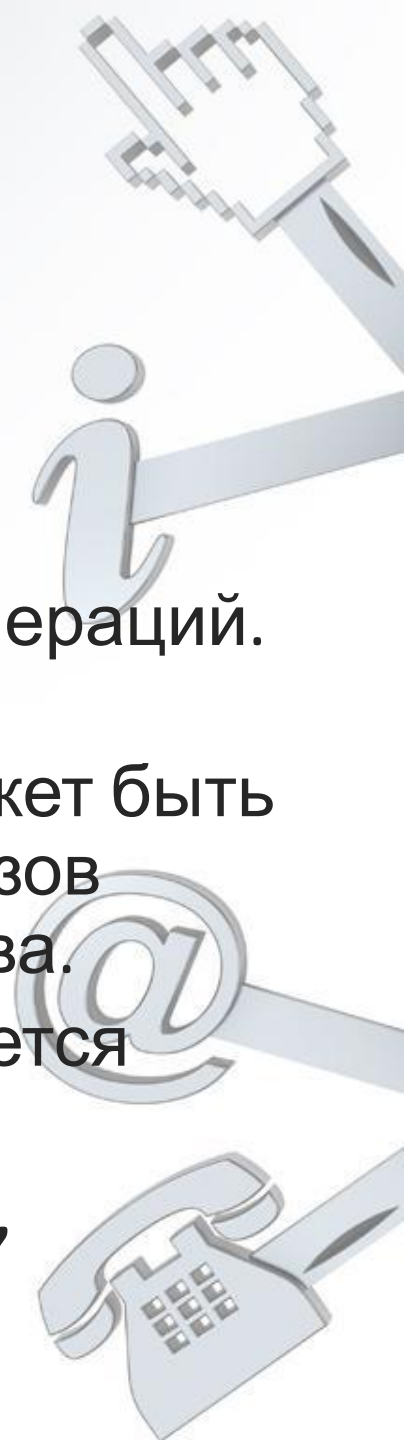
- **Public** – глобальная (уровень проекта, доступна из всех модулей и процедур, описание в разделе General Declarations главного модуля приложения).
- **Dim** – локальная (уровень модуля или процедуры - описание в разделе General Declarations модуля или формы; уровень процедуры - описание внутри процедуры).
- **Private** - локальная (уровень модуля или процедуры)
- **Static** – статическая (не обнуляется при выходе из процедуры, локальная)





Выражения

- **Выражение** - это набор операндов, объединенных между собой знаками операций.
- **Операнд** - это объект, над которым выполняется операция. Операндом может быть константа, переменная, выражение, вызов функции, обращение к элементу массива.
- Набор допустимых операций определяется типом операндов.
- Выражения делят на **арифметические, логические, выражения отношения и символные.**





Арифметические выражения

+ - сложение ($5+2$);

- - вычитание ($34-2$);

***** - умножение ($5*5$);

/ - деление ($3/5 = 0.6$);

**** - деление нацело ($5\backslash 2 = 2$);

Mod - остаток от деления ($7 \text{ Mod } 2 = 1$);

^ - возведение в степень ($2^2 = 4$, $25 ^ (1/2) = 5$);

Приоритет операций в порядке убывания:

1. **^**
2. *** /**
3. **\ Mod**
4. **+ -**

Операнды арифметического выражения могут быть следующих типов:
Integer, Long, Currency, Double, Single, Variant.

Примеры выражений:

$A^2+2*A*B+3.5$ $I+1$ $(1+R\backslash T)*0.5$ $(P21*SIN(X)-B)/(D+C)$

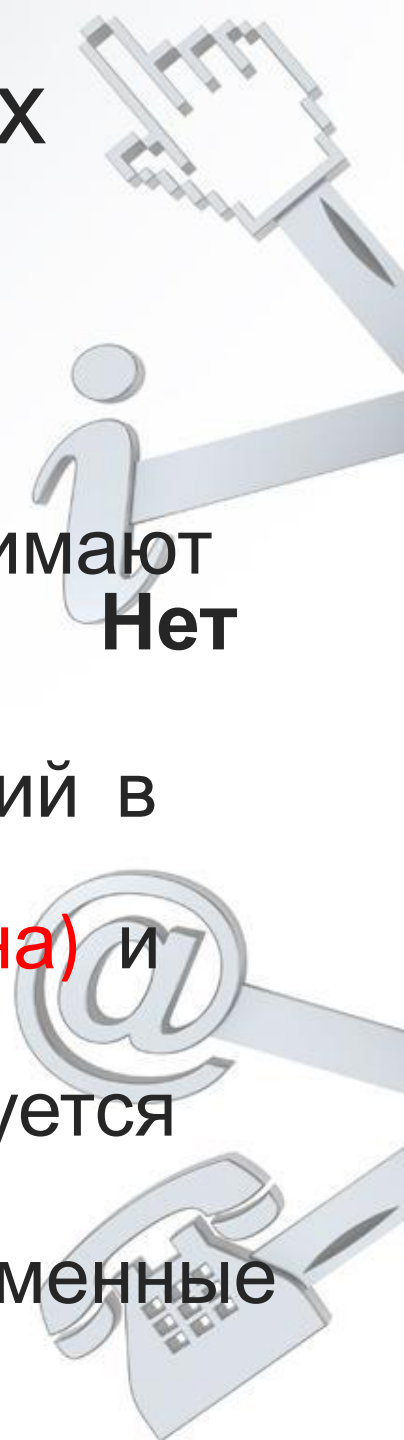




Логический тип данных

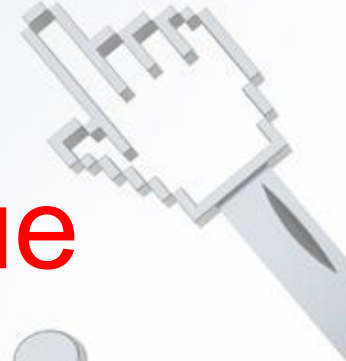
Boolean

- Переменные логического типа принимают только два значения : **Да (Истина)** **Нет (Ложь)**
- Для обозначения логических значений в языке программирования Vbasic используются константы: **True (Истина)** и **False (Ложь)**
- В операторе описания типа используется ключевое слово **Boolean**
- `Dim F1 As Boolean, F2 As Boolean` - переменные F1, F2 логического типа





Логическое выражение



- Для логических данных определены логические операции;
- Для логических операций операнды должны принимать значения или «истина» True или «Ложь» False.
- Логическим выражением называется выражение составленное с использованием логических операций и результатом вычисления которого является или TRUE или FALSE.



Логические операции

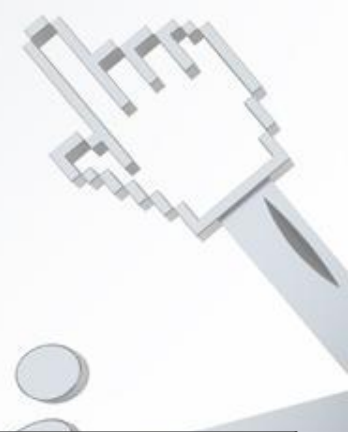
- **Not** - логическое отрицание (НЕ);
- **And** - логическое умножение (И);
- **Or** - логическое сложение (ИЛИ);
- **Xor** - логическое исключающее ИЛИ;

Также как для чисел существует таблица умножения, так же для любой логической операции существует таблица значений (**«таблица ИСТИННОСТИ»**), показывающая результат выполнения логической операции в зависимости от значений операндов.





Таблица истинности



Значения операндов		Результат операций			
A	B	Not A	A AND B	A OR B	A XOR B
True	True	False	True	True	False
False	True	True	False	True	True
True	False		False	True	True
False	False		False	False	False



Основные логические операции в порядке убывания **приоритета**

- Not - логическое отрицание (НЕ);
- And - логическое умножение (И);
- Or - логическое сложение (ИЛИ);
- Xor - логическое исключающее ИЛИ;

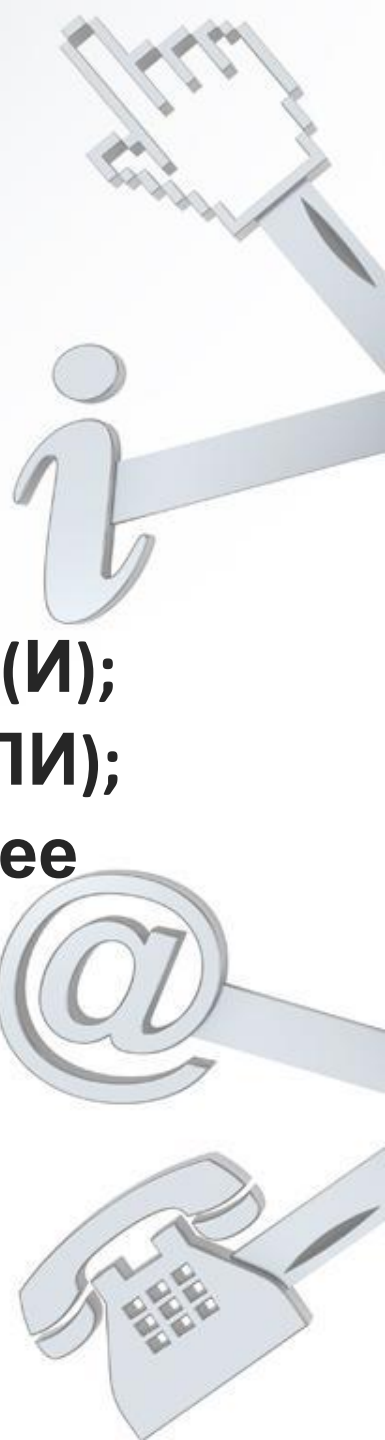


2

1

3

True And Not False Or False = True

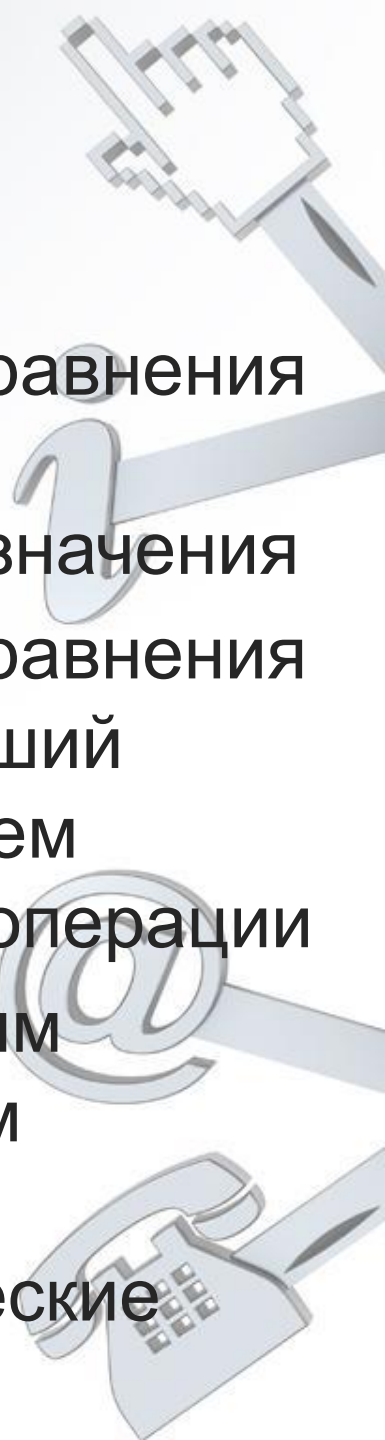




Операции сравнения

- $>$ больше
- $>=$ больше или равно
- $<$ меньше
- $<=$ меньше или равно
- $=$ равно
- $<>$ не равно

- Операции сравнения генерируют логические значения
- Операции сравнения имеют больший приоритет чем логические операции
- Ещё большим приоритетом обладают арифметические операции





Операции сравнения

- $"A" < "B" = \text{True}$ (сравнение строк)
чем ближе буква к началу алфавита(кодовой таблицы), тем она меньше
- $2 > 4 = \text{False}$

Двойные неравенства записываются с помощью логических операций:

- $A > 5 \text{ And } A < 10$ соответствуют математической записи $(5 < A < 10)$.





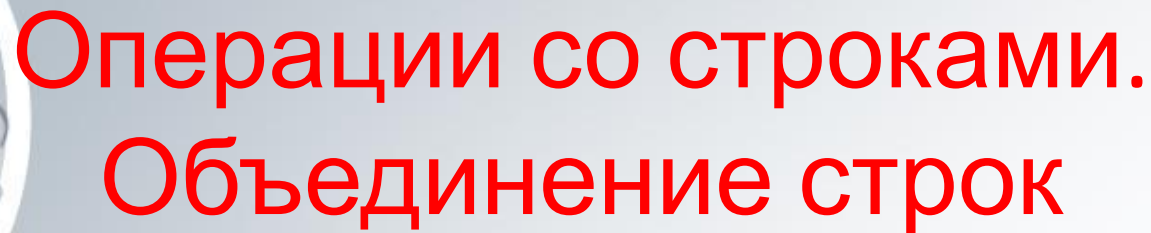
Символьные данные

- Символьные данные или строка символов - любая последовательность СИМВОЛОВ.
- В Visual Basic строка символов заключается в кавычки.
“12345678” ; “Иванов” ; “Фамилия И.О.”;
В операторе описания типа используется ключевое слово String.
- В Visual Basic символьные данные (строки символов) могут быть постоянной или переменной длины. Это определяется в операторе описания типа.

Примеры:

```
Dim Name As String*15, SurName As String
```

- Для работы с символьными данными используется **оператор конкатенации** (объединения) и **встроенные функции.**



Строка - "Иванов_" & "Иван_" & "Иванович"

Dim SurName As String*15

Dim Name As String

Dim Author As String*30

SurName = "ИВАНОВ" 'В SurName - "ИВАНОВ_____ " -15

Name = "Иван" 'длина равна 4 символам'

Author = SurName & Name

В переменной Author будет храниться значение

"Иванов_____Иван_____" - длина равна 30 символам.

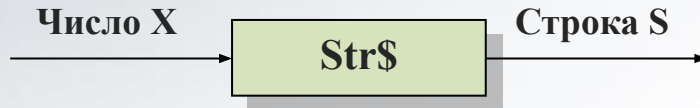
До заданной длины строка дополняется пробелами.



Встроенные функции для работы со строками. Преобразование данных



$X = \text{Val}(\text{"12.5"})$ 'Результат: число 12.5'



$S = \text{Str}\$(12.5)$ 'Результат: строка "12.5"'

Чаще всего такие преобразования необходимо использовать при работе с элементами управления (Label, TextBox, ...) и с функциями MsgBox или с InputBox.

```
Dim x As Integer  
x = Val(InputBox("Введите x"))
```

```
Text1.Text = str$(x)
```

Константы

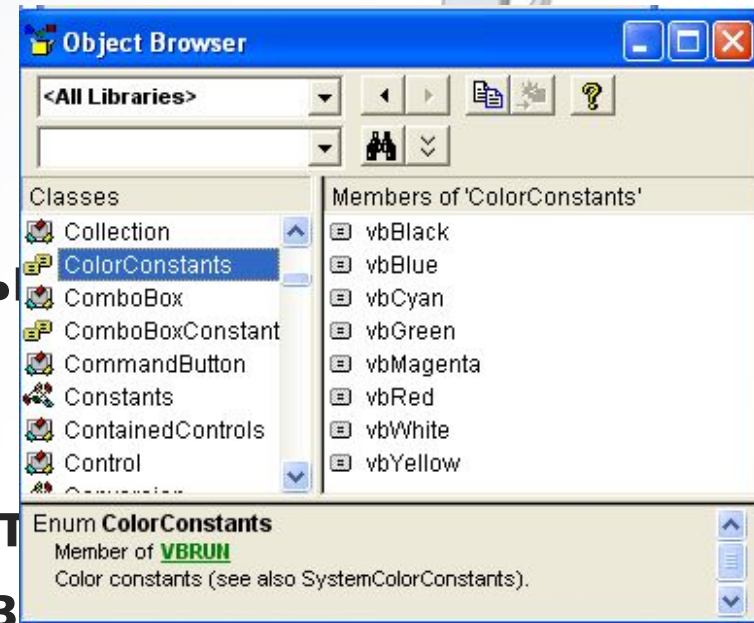
Константы – элементы выражения, значения которых не меняется в процессе выполнения программы.

Примеры

76.33 – числовая константа,
“Ошибка доступа к БД” – символ
константа

False –логическая константа

VB содержит огромное количество
встроенных констант (цвета, клавиш,
сообщения и т.д.). Имеют префикс vb.





Массивы

10	12	44	77	46	78	3
----	----	----	----	----	----	---

Различные данные часто представляются таблицами.

В языках программирования несколько однотипных и одномерных данных, обозначенных одним именем, называется **массивом**.

- Каждый элемент массива имеет порядковый номер (индекс), по которому его можно выбрать.
- Каждый массив содержит данные только одного типа.
- Все используемые в программе массивы должны быть определены с помощью оператора Dim.

Синтаксис:

Dim имя (нижняя_граница To верхняя_гр) As тип



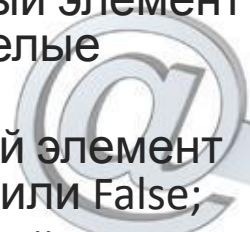


Массивы

Примеры:

- **Dim Tab(1 To 10) As Single**
- **Dim AA2(1 To 5) As Integer**
- **Dim C(1 To 25) As Boolean**
- **Dim Crab(1 To 5) As String * 10** 'длина строки - 10 символов

В этом примере определены:

- массив с именем Tab, состоящий из 10 элементов и каждый элемент арифметического типа Single;
 - массив с именем AA2, состоящий из 5 элементов и каждый элемент арифметического типа Integer, т.е. может иметь только целые значения;
 - массив с именем C, состоящий из 25 элементов и каждый элемент логического типа, т.е. может иметь только значения True или False;
 - массив с именем Crab, состоящий из 5 элементов и каждый элемент типа строки символов длиной 10 символов.
- 
- 



Массивы

При использовании в программе некоторого элемента определенного массива указывается имя массива и в скобках индекс элемента этого массива.

Примеры:

- $\text{Tab}(7)$ - используется 7 ой элемент массива Tab ;
- $\text{AA2}(3) + \text{AA2}(4)$ - сложение 3-го и 4-го элемента массива AA2 ;
- $\text{C}(I)$ - используется I -ый элемент массива C (значение I должно быть определено и не может быть более 25);
- $\text{Crab}(J+1)$ - в качестве значения индекса используется арифметическое выражение (его значение не может быть более 5).



Массивы

В VB используются массивы фиксированного размера и **динамические массивы** (могут менять размер в процессе выполнения программы).

Позволяют эффективно управлять памятью.

Порядок использования динамического массива:

1. Объявляется массив с помощью ключевых слов, используемых при создании массива фиксированного размера. Список размерностей остается пустым.

Dim intCount () As Integer

2. С помощью выполняемого оператора ReDim указывается размерность массива в виде числа или выражения.

ReDim intCount (x)

ReDim intCount (20)

ReDim intCount (1 To 20)

При выполнении оператора ReDim данные, размещенные в массиве теряются.

Для изменения размера массива без потери данных ReDim используется с ключевым словом Preserve

ReDim Preserve intCount (x+1)

