

Университет машиностроения
Кафедра «Автоматика и процессы управления»

Дисциплина
Информационные технологии
2 семестр

Тема 05

Сложные типы данных
(Язык VBA for Ms Excel)

Простые (Simple) типы данных

Byte 'Короткое целое без знака

Boolean 'Логический

Integer 'Целое со знаком

Long 'Длинное целое со знаком

Single 'С плавающей точкой одинарной точности

Double 'С плавающей точкой двойной точности

Date 'Дата и время

Decimal 'Действительное значение, хранимое как целое

Currency 'Число строго с четырьмя знаками после запятой

«Почти-простые» типы данных

Variant 'Нетипизированный указатель

Object 'Указатель на объект

String 'Строка символов

Пользовательские типы данных

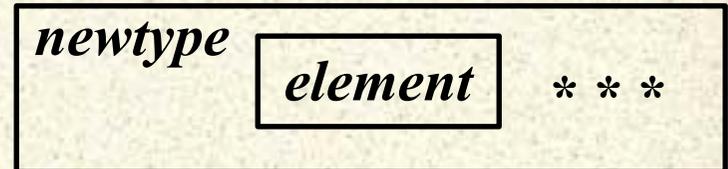
Директива **Type** задает (декларирует) описание *пользовательского типа*. В VBA под пользовательским типом понимается *структура* (**struct** в с#) или *запись* (**record** в Delphi).

Type *newtype* 'Определили имя нового типа данных

element As SimpleType 'Описали элемент структуры

* * *

End Type



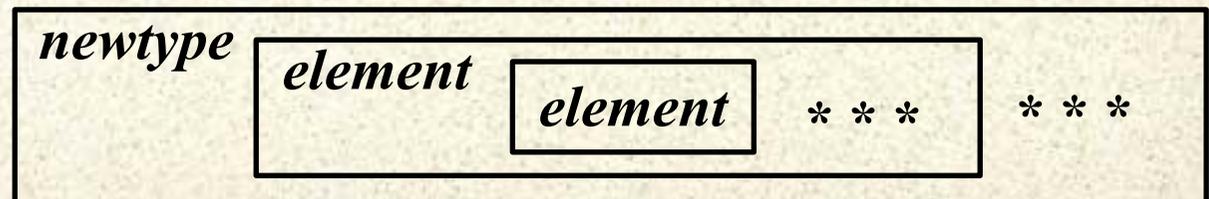
Пользовательские типы могут «вкладываться» друг в друга

Type *extranewtype*

element As newtype '«Вложили» описанный ранее тип

* * *

End Type



Использование

Главное преимущество использования пользовательских типов – поддержание целостности данных при копировании

```
Dim Ivanov_fio As String
Dim Ivanov_course As Integer
Dim Ivanov_score As Double
Dim students_fio() As String
Dim students_course() As Integer
Dim students_score() As Double
* * *
Ivanov_fio = "ИВАНОВ"
students_score(1) = 5
* * *
students_fio(5) = Ivanov_fio
students_course(5) = Ivanov_course
students_score(5) = Ivanov_score
```

```
Type student
    fio As String
    course As Integer
    score As Double
End Type
* * *
Dim Ivanov As student
Dim students() As student
* * *
Ivanov.fio = "ИВАНОВ"
students(1).score = 5
* * *
students(5) = Ivanov
```

А если нужно добавить студенту, например, возраст?

Объявление и видимость

В отличие от «стандартных» типов, объявление пользовательского типа должно быть доступно для любого кода, использующего этот тип.

Лучший вариант – **вынесение объявлений в отдельный модуль**

DefModule

```
Type student
  fio As String
  course As Integer
  score As Double
End Type
```

Module1

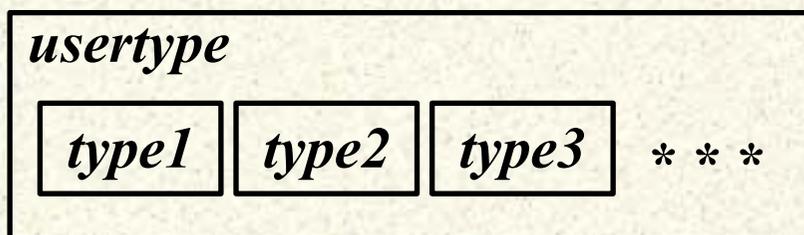
```
Dim Ivanov As student
Dim Ivanov As DefModule.student
```

Если тип объявлен в нескольких модулях, то используем полное имя типа: ***ModuleName.TypeName***

Если тип с совпадающим именем объявлен также в своём модуле, то его определение экранирует остальные. При необходимости, можно использовать полное имя типа: ***ModuleName.TypeName***

Статические и динамические массивы

Иногда нужно сохранять набор однотипных данных



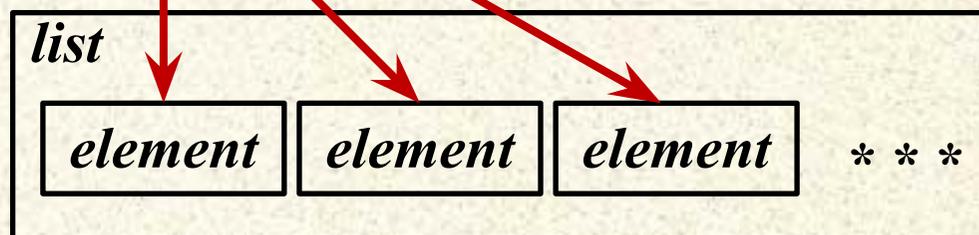
Type *usertype*

type1 As String

type2 As Integer

type3 As Double

End Type



Если данные **одного типа** (даже пользовательского) и их **можно упорядочить** (пусть и формально) для хранения набора можно использовать **массив**

Dim *ArrayName*(*Length*) As *smthtype*

Dim *ArrayName*() As *smthtype*

Важно:

Для статического массива заранее известно количество элементов

Для динамического массива - количество элементов может меняться

Работа с индексами

Для статического массива (массива фиксированного размера):

`Dim Array1(5) As Integer` 'Определена верхняя граница

`Dim Array2(2 To 10) As Integer` 'Определены обе границы

Возможно создать многомерный «прямоугольный» массив

`Dim Array3(5, 2 To 10) As Integer` 'Определен двумерный массив

Не путать размер и размерность!

Для динамического массива (массива переменного размера):

`Dim Array4() As Integer` 'Определен динамический массив

Перед использованием обязательно указать размер и размерность

`ReDim Array4(2 To 6, 3) As Integer` 'Настройка массива

При использовании ReDim массив «очищается», но

`ReDim Preserve Array4(2 To 6, 5) As Integer` 'Данные сохраняются

Preserve используется **только** при изменении последнего индекса

На границе...

Менеджер памяти VBA всегда выполняет контроль диапазонов индексов для массивов при обращении к элементам

Ошибка '9': Subscript out of range

В VBA индекс элемента массива не равен его порядковому номеру

Option Base 2 'Установка нижнего индекса «по-умолчанию»

Dim Array5 (4) As Integer 'Объявлен массив с 3 элементами

Dim Array5 (1 To 4) As Integer 'Объявлен массив с 4 элементами

Индексы массива всегда образуют непрерывный ряд целых чисел

В VBA определены функции для получения граничных индексов

MinIndex = LBound (ArrayName) 'Нижняя граница

MaxIndex = UBound (ArrayName) 'Верхняя граница

Пример: Добавление элемента **X** «в конец» массива **A()**:

N = UBound (A); **ReDim Preserve A(N + 1)**; **A(N + 1) = X** ' «на вершину»

Работа со строками

Строка – это последовательность символов.

Последовательность символов образующая строку хранится в памяти как неизменная величина в динамической памяти. Указатель на структуру, образующую строку, обрабатывается средой VBA «прозрачно» для пользователя.

Объявление переменной строкового типа

`Dim str1 As String` ‘Строка переменной длины

`Dim str2 As String*10` ‘Строка фиксированной длины

Для строк определены два оператора

`str1 = "Hello world!"` ‘Присвоение значения

`str1 = "Здравствуй, " & "Мир!"` ‘Конкатенация – слияние строк

Функции преобразования типов

`str1 = Str (100500)` ‘Получение строкового представления

`N = Val ("100500")` ‘Преобразование строки в число (тип Double)

VBA использует для строк кодировку Unicode (UTF-16LE)

Функции для обработки строковых значений

Строка – это последовательность двухбайтных кодовых единиц. Непосредственная работа с кодовыми единицами выполняется функциями `Asc()` и `Chr()`:

`1055 ← AscW (“Привет”)` ‘ Unicode-код первого символа строки

`31 ← AscB (“Привет”)` ‘ Значение первого байта строки

`207 ← Asc (“Привет”)` ‘ ASCII-код первого символа строки

`П ← ChrW (1055)` ‘ Символ по Unicode-коду

`Z ← ChrB (90)` ‘ **!!! Неполная кодовая единица !!!**

`П ← Chr (207)` ‘ Символ по ASCII-коду

Можно использовать комбинации

`П̄ ← ChrB(31) & ChrB(4) & ChrW(774)` ‘ «Надчёркнутое» П (cyrilic)

Подсчет «длины» строки – функция `Len()`

`L = Len (“Привет”)` ‘ Число «полных» кодовых единиц в строке

Число кодовых единиц не равно числу кодовых точек (символов)

`L = LenB (“Привет”)` ‘ «Устаревшая» функция «длина в байтах»

Полезные функции

Выборка фрагмента строки

- Mid (<строка>, <старт> [, <длина>])** ‘ Скопировать подстроку
- Mid (<строка>, <старт>)** ‘ Скопировать от <старт> до конца
- Left (<строка>, <длина>)** ‘ Скопировать от левого края
- Right (<строка>, <длина>)** ‘ Скопировать от правого края

«Чистка» от служебных символов и пробелов

- Trim (<строка>)** ‘ Удаление краевых пробелов (всех)
- LTrim (<строка>)** ‘ Удаление пробелов слева
- RTrim (<строка>)** ‘ То же, но справа

Анализ строки (поиск подстрок и замена вхождений)

- InStr([<старт>,) < строка1>, <строка2> [,<сравнение>])** ‘ Поиск подстроки
- InStrRev(...)** ‘ Поиск подстроки, но с конца
- Replace (<строка>, <строкаПоиск>, <строкаЗамена>)** ‘ Замена

Задание: Напишите программу обработки CSV:

- S = “1; 2; 3; 4”** ‘ Загрузить в Dim A() AS String по отдельности

Полезные функции 2

Работа с символами в строках

StrReverse (<строка>) ‘ «Переворот» строки

UCase (<строка>) ‘ Всё в ВЕРХНИЙ регистр

LCase (<строка>) ‘ Всё в нижний регистр

StrConv (<строка> [, <метод>]) ‘ Преобразование строки

‘ метод = vbProperCase (Слова

‘ начинаются с заглавных)

Проверка и сравнение строк

<строка> **Like** <шаблон> ‘ Сравнить с шаблоном

StrComp (<строка1>, <строка2> [, <метод>]) ‘ Сравнить строки:

‘ Результат = { -1, 0, 1, Null } для { <, =, >, Null }

‘ <метод> = { vbBinaryCompare, vbTextCompare }

Заполнение

String (<длина>, <символ>) ‘ Создать строку из символов

Space (<длина>) ‘ То же, но из пробелов

Операции со строковыми массивами

Разбор строки в массив по символу-разделителю

Buf = "" ' Это буфер

For I = 1 To Len(Str1) ' «Пробежимся» по строке

If (Mid(Str1, I, 1) = ";") Then ' Если i-тый символ – разделитель

***** Buf → S() ***** ' Здесь код добавления в массив

Else ' Если i-тый символ – не разделитель

Buf = Buf & Mid(Str1, I, 1) ' Добавляем в буфер

Next I

Ничего не упустили?

Очистка буфера, последний элемент , а если нет разделителя?

Что предлагает VBA?

Split (<строка> [, <разделитель>]) ' Разбить строку на подстроки

Join (<массив строк> [, <разделитель>]) ' Объединить строки

Filter(<массив строк>, <фрагмент>[, <включение>] [, <сравнение>])

' Просмотр массива строк с целью поиска искомой

' Возвращает массив подходящих строк

Полезная «экзотика»

Непрямоугольные массивы

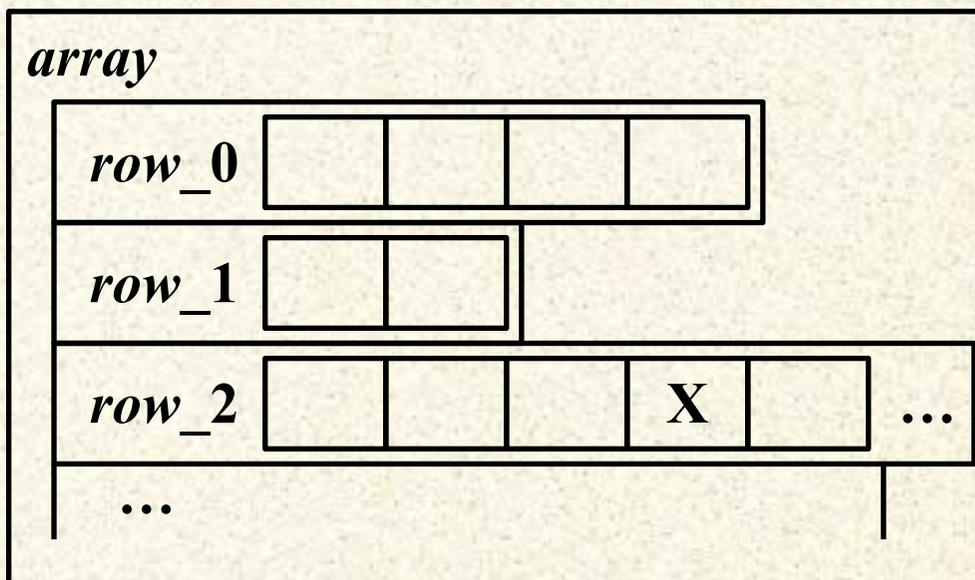
Type rowtype

el() As String

End Type

Dim myarr() As rowtype

myarr(2).el(3) = "X"



«Разреженные» массивы

Type celltype

id As Integer

value As String

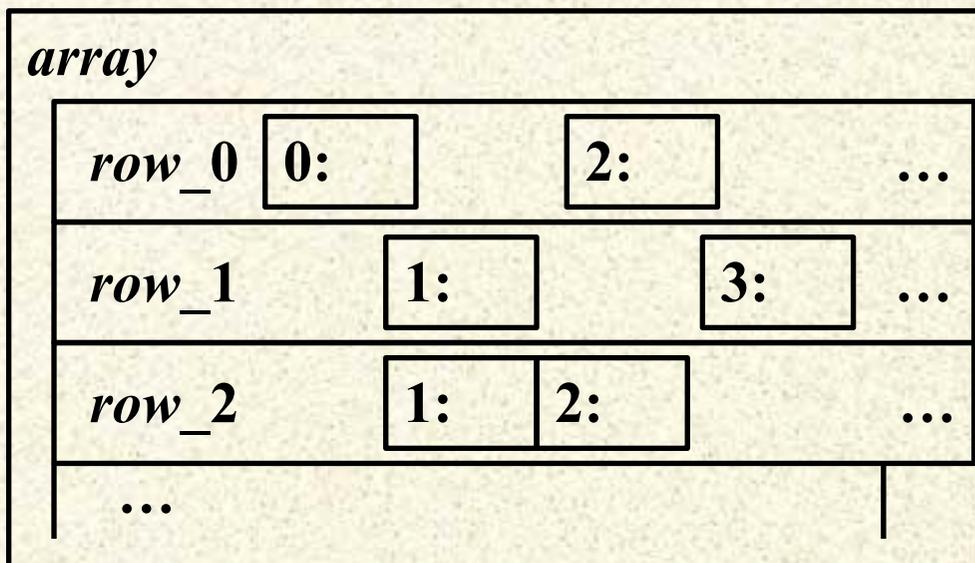
End Type

Type rowtype

el() As celltype

End Type

* * *



А есть ещё «облака» и коллекции...

Спасибо за внимание !!!

Далее:

- **Обработка данных**
- **Основные алгоритмические конструкции**
- **Сложные типы данных. Работа со строками**
- **Использование функций. Передача параметров**

...

Контакты:

mami.testolog.ru

timid@mami.ru

inform437@gmail.com