

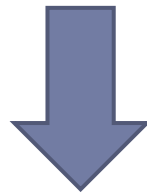
Офісне програмування

Базові засоби мови програмування VBA

Слайди до лекцій
(1 частина)

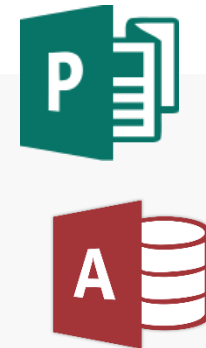
Програмування в Microsoft Office – усунення повторюваних дій

- ▶ однакова обробка документів
- ▶ перевірка введених користувачем даних
- ▶ аналіз інформації таблиці за різні періоди
- ▶ додавання елементів до слайдів за умовами
- ▶ робота з інформацією з бази даних
- ▶ ...



VBA (Visual Basic for Applications)

- ▶ VBA (Visual Basic for Applications) - це діалект мови Visual Basic, що розширює його можливості і призначений для роботи з додатками Microsoft Office і іншими додатками від Microsoft та інших компаній і фірм
- ▶ Мова VBA вбудована в додатки Microsoft Office , її код можна зберігати всередині файлів додатків та запускати звідти.



VBA (Visual Basic for Applications)

- ▶ Мова VBA вбудована в інші додатки Microsoft – MS Visio, MS Project



- ▶ Мова VBA вбудована в додатки інших виробників - CorelDraw и CorelWordPerfect Office 2000, AutoCAD



WordPerfect®



A AUTODESK®
AUTOCAD®

- ▶ Програми на VBA за замовчуванням не компілюються

Використання макрорекордера

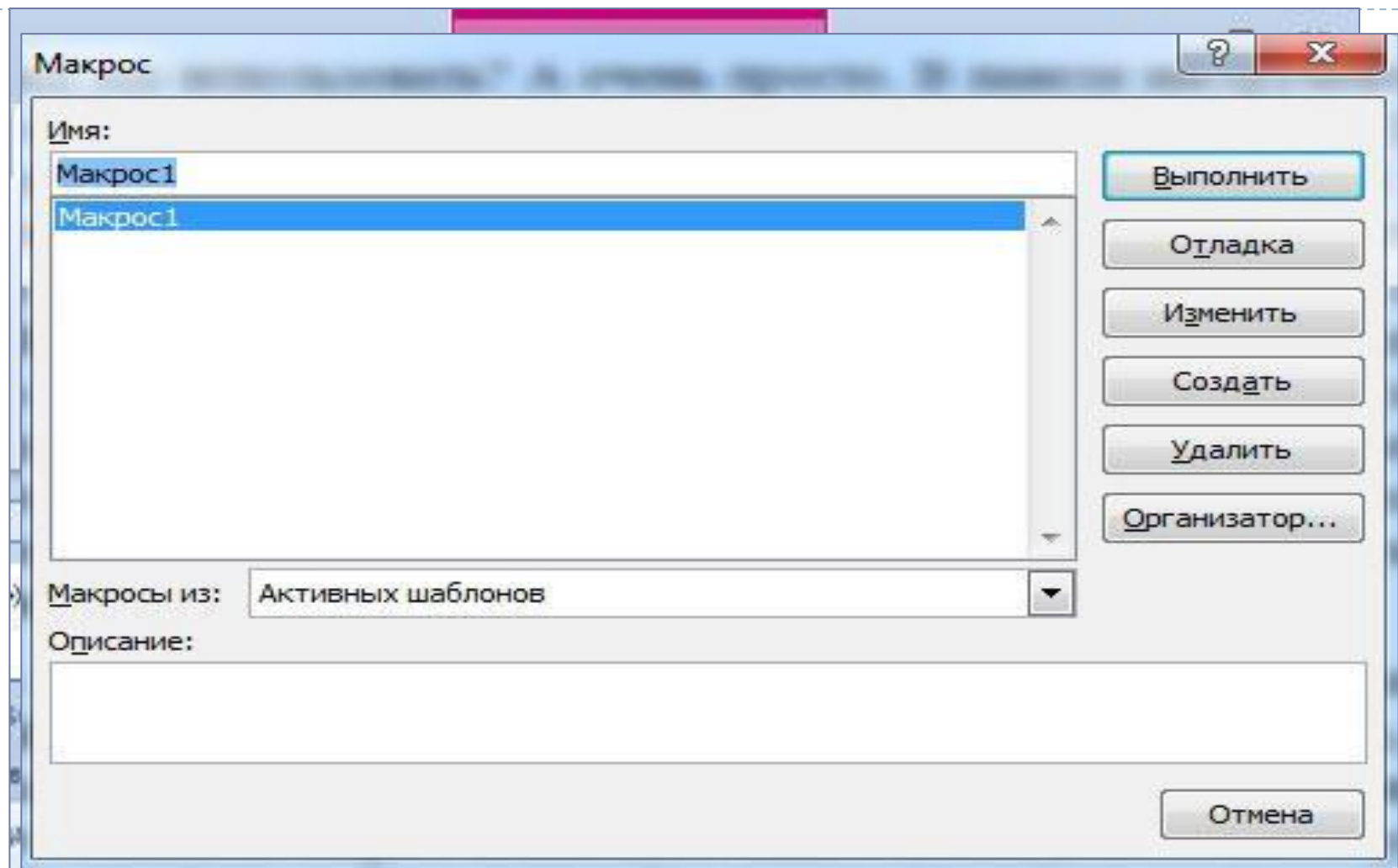
- ▶ Макрорекордер - засіб для запису макросів.
- ▶ Макрос - VBA-програма, що створюється автоматично макрорекордером.

Алгоритм дій при створенні макроса*:

- ▶ визначити порядок дій (бажано використовувати гарячі клавіші);
- ▶ виконати;
- ▶ записати дії в макрос.

* В MS Office 2016 – 2018 необхідно відобразити команду меню Разработчик

Використання макрорекордера



Демонстрація створення та використання макросу

ПРОЕКТ VBA

- ▶ MS Word зберігає проекти в документах і шаблонах з бінарним форматом (DOC і DOT) і в документах і шаблонах з форматом Open XML з підтримкою макросів (DOCM і DOTM).
- ▶ MS Excel зберігає проекти в робочих книгах і шаблонах з бінарним форматом (XLS і XLT) і в робочих книгах і шаблонах з форматом Open XML з підтримкою макросів (XLSM і XLTM).
- ▶ MS Access зберігає проекти в файлах бази даних (MDB і ACCDB).
- ▶ MS PowerPoint зберігає проекти в презентаціях і шаблонах з бінарним форматом (PPT, PPS і POT) і в презентаціях з підтримкою макросів (PPTM, PPSM і POTM).



Елементи проекту VBA

- ▶ **Процедури** - поійменована послідовність сумісно виконуваних інструкцій (операторів). Існують процедури типу Sub, Function і Property. Ім'я процедури завжди визначається на рівні модуля.
- ▶ **Модулі** - служать для розміщення в них процедур. Всередину одного модуля має сенс поміщати процедури, що забезпечують вирішення однієї прикладної задачі.
- ▶ **Макроси** - підпрограми (процедури типу Sub), що не мають параметрів і розміщуються в модулі. Макроси, створювані шляхом протоколювання дій користувача, розміщуються в модулі *New Macros*.
- ▶ **Модулі класу** служать для розміщення в них опису об'єктів, що створюються в VBA. Кожен модуль класу містить опис одного типу об'єкта.
- ▶ **Форми** дозволяють вирішити задачу організації введення і виведення даних. У середовищі VBA можна розробляти форми, а також процедури обробки подій, що виникають у цих формах.



Включения меню Разработчик

Параметры Excel

?

✕

Общие

Формулы

Правописание

Сохранение

Язык

Дополнительно

Настроить ленту

Панель быстрого доступа

Надстройки

Центр управления безопасностью



Настройка ленты

Выбрать команды: ⓘ

Часто используемые команды

- Быстрая печать
- Вернуть
- Все типы диаграмм...
- Вставить
- Вставить
- Вставить рисунок
- Вставить столбцы на лист
- Вставить строки на лист
- Вставить таблицу
- Вставить функцию...
- Вставить ячейки...
- Вырезать
- Границы
- Диспетчер имен
- Добавить или удалить филь...
- Задать
- Закрепить области
- Копировать
- Макросы
- Настраиваемая сортировка...
- Обновить все

Добавить >>

<< Удалить

Настроить ленту: ⓘ

Основные вкладки

- Файлы
- Стили
- Ячейки
- Редактирование
- Вставка
- Разметка страницы
- Формулы
- Данные
- Рецензирование
- Вид
- Разработчик**
- Надстройки
- Acrobat
- Удаление фона

Создать вкладку

Создать группу

Переименование...

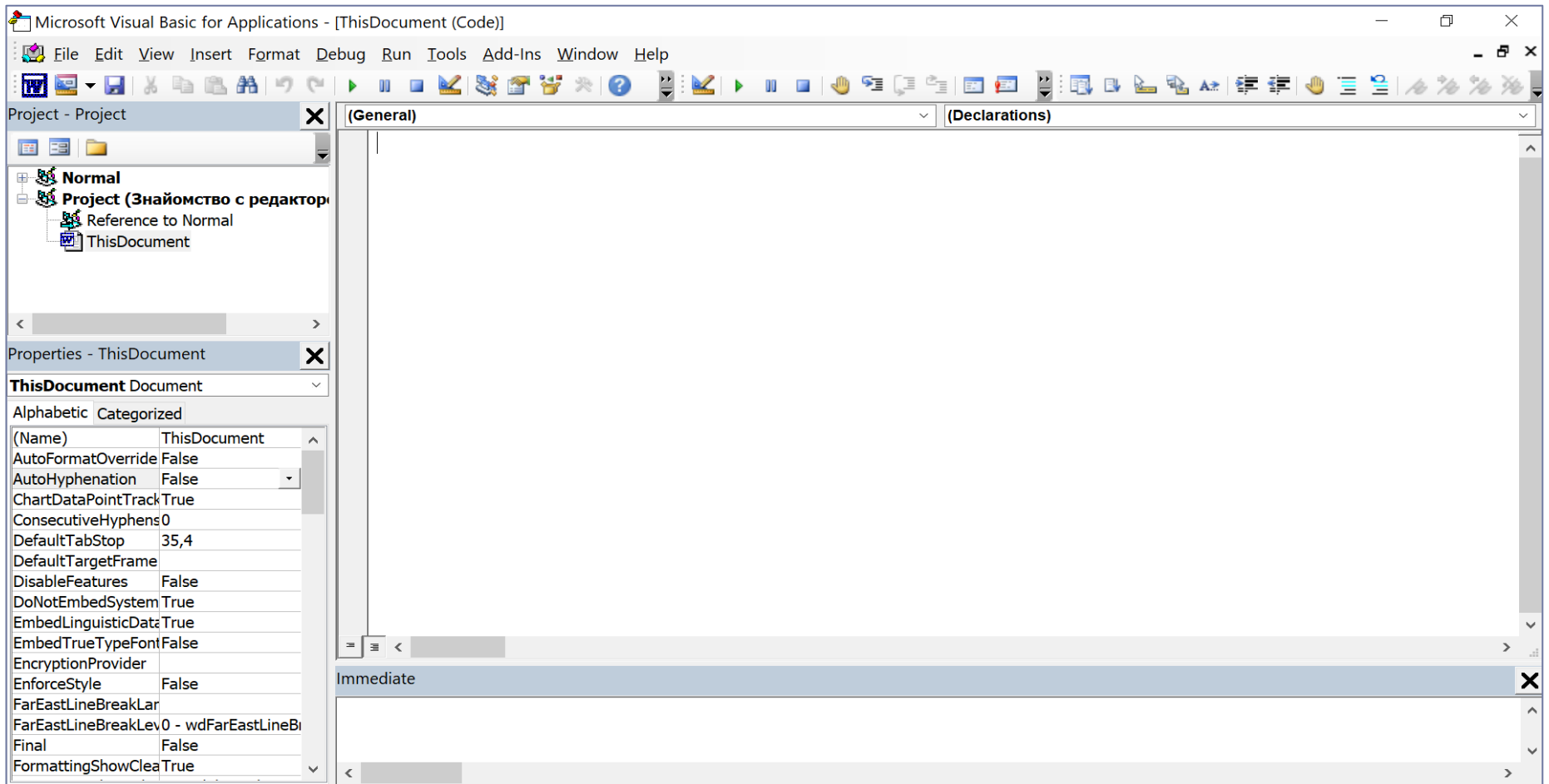
Настройки: Сброс ⓘ

Импорт и экспорт ⓘ

ОК

Отмена

Редактор Visual Basic



Вікна редактору Visual Basic

- ▶ **Project Explorer**- вікно провідника проекту. У ньому можна переглянути компоненти проекту і виконати з ними безліч операцій;
- ▶ **Code**- вікно програмного коду. У цьому вікні виконується основна робота з написання коду макросу. При відкритті програмного модуля відкривається автоматично.
- ▶ **Properties**- вікно властивостей. Через нього можна переглянути властивості елемента управління або компонента проекту і змінити їх.
- ▶ **Immediate**- вікно, у якому при налагодженні можна негайно отримати результат. У цьому ж вікні також можна виконати обчислення довільних виразів, задаючи їх, починаючи з символу ? та завершуючи натисканням клавіші Enter
- ▶ **Watch**- вікно контрольованих виразів. Використовується під час налагодження для відстеження значень обраних змінних програми і виразів.
- ▶ **Locals**- вікно локальних змінних. Потрібно для відстеження під час налагодження значень змінних поточної процедури.
- ▶ **UserForm**- вікно форми. З'являється тоді, коли Ви редагуєте користувацьку форму з вікна дизайнера форми.
- ▶ **Toolbox**- панель інструментів управління. З неї можна додати елементи управління в форму або в документ, з'являється разом з вікном дизайнера форм.
- ▶ **Object Browser**- оглядач об'єктів. Необхідний для отримання інформації про класи, доступні програмі.

Вікна редактору Visual Basic

Microsoft Visual Basic for Applications - Перевірка суми кредитів, середнього балу та наявності номерів-v.1.5 [break] - [ThisDocument (Code)]

File Edit View Insert Format Debug Run Tools Add-Ins Window Help

Project - Project

Normal
Project (ОП-лекції-модуль-1)
Reference to Normal
ThisDocument
Project (Перевірка суми кредитів, середнього балу)
Reference to Normal
ReportForm
ThisDocument
Project (Програми на VBA до конспекту лекцій)

Properties - CommandButtonCreditSum

CommandButtonCreditSum CommandButton

Alphabetic Categorized

(Name)	CommandButtonCreditSum
Accelerator	
AutoSize	False
BackColor	&H8000000F&
BackStyle	1 - fmBackStyleOpaque
Caption	Перевірити суму кредитів додатків
Enabled	True
Font	Calibri
ForeColor	&H80000012&
Height	26,05
Locked	False
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PicturePosition	7 - fmPicturePositionAboveCenter
TakeFocusOnClick	True
Width	228,85
WordWrap	False

(General)

CommandButtonCreditSum
CommandButtonGPA
CommandButtonNumbersCheck
Document

список об'єктів

OpenDocs

(Declarations)

CheckCreditSum
CheckEmptyNumbers
CheckGPA
CheckParagraph
CreateReport
OpenDocs
SetCaptionString
SetDiplomaSeriaYear
ShowReport

список процедур-подій

```
Sub OpenDocs (commandButtonNumber As Integer)

    Dim vDir, vFile As String
    Dim oDoc As Document
    Dim wrongDocs As String
    Dim isDiplomaSupplement As Boolean
    Dim isBachelorDiplomaSupplement, isMasterDiplomaSupplement As Boolean

    vDir = ActiveDocument.Path & "\"
    vFile = Dir(vDir & "*.doc?")

    Do While vFile <> ""
        Set oDoc = Documents.Open (FileName:=vDir & vFile)
        'Word 2007 does not work correctly with wildcard chars so check SECOND paragraph string
        'that is same for honors degree 'Paragraph text ends with CR char so add it to comparing
        isDiplomaSupplement = CheckParagraph(oDoc, 2, "Diploma Supplement")
    Loop
```

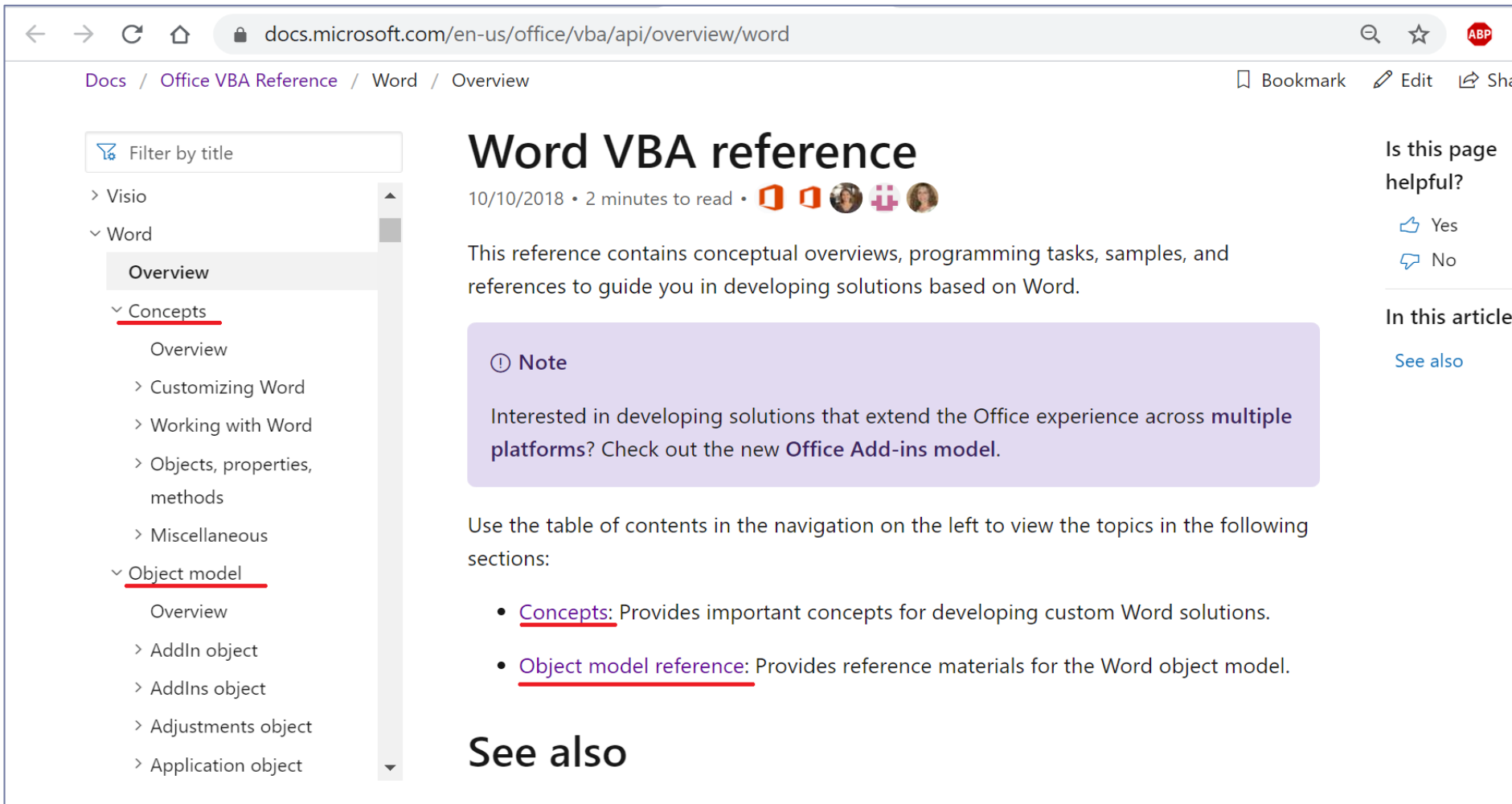
Immediate

Locals

Watches

Expression	Value	Type	Context
Project.ThisDocument.OpenDocs			
Me	ThisDocument/C		
commandBut 0	Integer		
vDir	"C:\Users\kpg\G Variant/String		
vFile	"014_Середня	String	
oDoc	Nothing	Document	
wrongDocs	""	String	
isDiplomaSu	False	Boolean	
isBachelorDi	Empty	Variant/Empty	
isMasterDipl	False	Boolean	

Довідка Visual Basic



← → ↺ 🏠 🔒 docs.microsoft.com/en-us/office/vba/api/overview/word 🔍 ☆ ABP

Docs / Office VBA Reference / Word / Overview Bookmark Edit Share

Filter by title

- > Visio
- ▼ Word
 - Overview**
 - ▼ Concepts
 - Overview
 - > Customizing Word
 - > Working with Word
 - > Objects, properties, methods
 - > Miscellaneous
 - ▼ Object model
 - Overview
 - > AddIn object
 - > AddIns object
 - > Adjustments object
 - > Application object

Word VBA reference

10/10/2018 • 2 minutes to read •  

This reference contains conceptual overviews, programming tasks, samples, and references to guide you in developing solutions based on Word.

Note

Interested in developing solutions that extend the Office experience across **multiple platforms**? Check out the new **Office Add-ins model**.

Use the table of contents in the navigation on the left to view the topics in the following sections:

- Concepts: Provides important concepts for developing custom Word solutions.
- Object model reference: Provides reference materials for the Word object model.

See also

Is this page helpful?

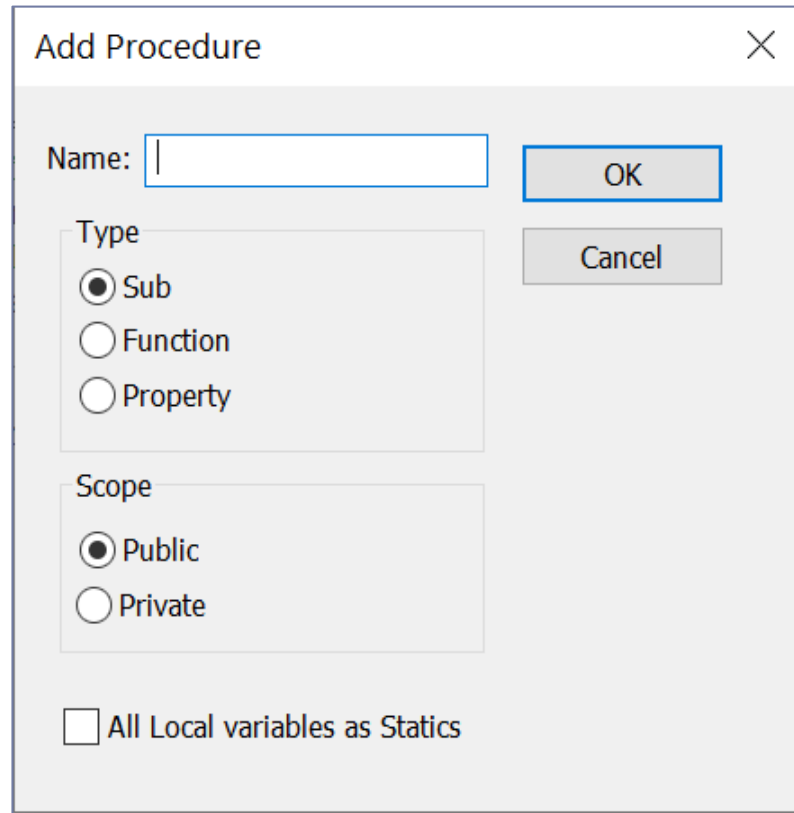
 Yes

 No

In this article

[See also](#)

Редактор Visual Basic



The image shows a standard Windows-style dialog box titled "Add Procedure". It has a close button (X) in the top right corner. The dialog is divided into several sections. At the top, there is a "Name:" label followed by a text input field. To the right of the input field are two buttons: "OK" and "Cancel". Below the name field is a section titled "Type" which contains three radio button options: "Sub" (which is selected), "Function", and "Property". Below the "Type" section is another section titled "Scope" which contains two radio button options: "Public" (which is selected) and "Private". At the bottom of the dialog is a checkbox labeled "All Local variables as Statics", which is currently unchecked.

Add Procedure

Name:

OK Cancel

Type

☒ Sub
☐ Function
☐ Property

Scope

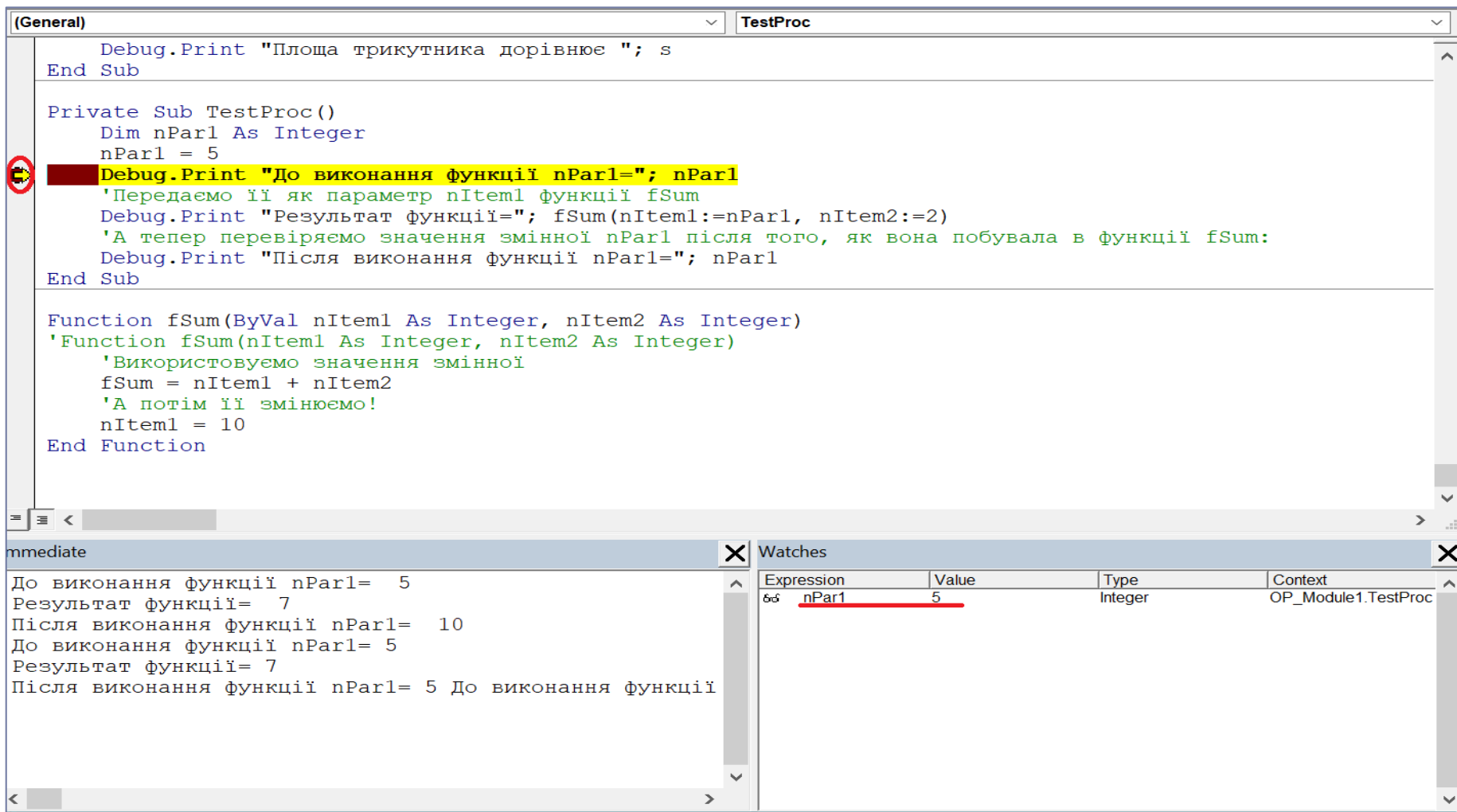
☒ Public
☐ Private

☐ All Local variables as Statics

Insert - Procedure...



Редактор Visual Basic



The screenshot displays the Visual Basic IDE interface. The main window is the code editor, showing a VBA module named `TestProc`. The code includes a `TestProc` subroutine and a `fSum` function. A red circle with a white 'X' marks a breakpoint on the line `Debug.Print "До виконання функції nPar1="; nPar1`. The code is as follows:

```
End Sub

Private Sub TestProc()
    Dim nPar1 As Integer
    nPar1 = 5
    Debug.Print "До виконання функції nPar1="; nPar1
    'Передаємо її як параметр nItem1 функції fSum
    Debug.Print "Результат функції="; fSum(nItem1:=nPar1, nItem2:=2)
    'А тепер перевіряємо значення змінної nPar1 після того, як вона побувала в функції fSum:
    Debug.Print "Після виконання функції nPar1="; nPar1
End Sub

Function fSum(ByVal nItem1 As Integer, nItem2 As Integer)
    'Function fSum(nItem1 As Integer, nItem2 As Integer)
    'Використовуємо значення змінної
    fSum = nItem1 + nItem2
    'А потім її змінюємо!
    nItem1 = 10
End Function
```

Below the code editor, there are two windows:

- Immediate Window:** Shows the execution flow with the following text:
До виконання функції nPar1= 5
Результат функції= 7
Після виконання функції nPar1= 10
До виконання функції nPar1= 5
Результат функції= 7
Після виконання функції nPar1= 5 До виконання функції
- Watches Window:** Contains a table with the following data:

Expression	Value	Type	Context
nPar1	5	Integer	OP_Module1.TestProc



Debug

Синтаксис VBA

- ▶ VBA нечутливий до регістру;
- ▶ максимальна довжина будь-якого імені в VBA (змінної, константи, процедури) - 255 символів;
- ▶ початок нового оператора - це символ переводу на новий рядок (крапка з комою, як в C, Java, JavaScript для цього не використовується);
- ▶ щоб закоментувати код до кінця рядка, використовується одинарна лапка (') або команда REM;
- ▶ символні значення мають вкладатися в подвійні лапки (" ");
- ▶ декілька операторів в одному рядку розділяються двокрапкою:

```
MsgBox "Перевірка 1": MsgBox "Перевірка 2"
```
- ▶ для зручності читання можна об'єднати кілька фізичних рядків в один логічний за допомогою пробілу:

```
MsgBox "Повідомлення користувачеві" _  
      & vUserName
```
- ▶ обмежень на максимальну довжину рядка немає

Правила іменування

- ▶ імена повинні починатися з літери;
- ▶ ім'я не повинно містити пробіл, крапку, знак оклику і символи @, &, \$, #;
- ▶ ім'я може містити символ підкреслення _
- ▶ ім'я повинне включати не більше 255 символів;
- ▶ не рекомендується обирати імена, що збігаються з ключовими словами VBA;
- ▶ ім'я кожної змінної повинно бути унікальним у поточній області видимості.



Класифікація типів даних

- ▶ **Прості і складені** типи даних. Прості типи даних можуть приймати єдине значення. Складені типи даних можуть приймати множини значень даних.
- ▶ **Вбудовані і визначені користувачем** типи даних. Вбудовані типи даних належать мові програмування і складають її базис. На основі вбудованих типів програміст може будувати власні, їм визначені типи даних.
- ▶ **Статичні і динамічні** типи даних. Для даних статичного типу пам'ять відводиться в момент їх оголошення - необхідний розмір даних відомий в момент оголошення. Для даних динамічного типу розмір даних в момент оголошення не відомий, і пам'ять їм виділяється в процесі виконання програми.

Прості типи даних VBA

Ім'я типу	Можливі значення	Необхідна пам'ять
Boolean	True, False	2 байти
Byte	0..255	1 байт
Integer	-32768 ... + 32767	2 байти
Long	Приблизно -2 000 000 000 ... + 2 000 000 000	4 байти
Decimal	Приблизно 30 десяткових цифр; можна вказати число цифр після десяткової коми	12 байтів
Single	-3,4E38 ... -1,4E-45 для негативних значень; 1,4E-45 ... 3,4E38 для позитивних значень	4 байти
Double	-1,7E308 ... -4,9E-324 для негативних значень; 4,9E-324 ... 1,7E308 для позитивних значень	8 байтів
Currency	Десяткові числа з фіксованою позицією коми. можливі 15 цифр до коми і 4 після	8 байтів

Прості типи даних VBA - продовження

Ім'я типу	Можливі значення	Необхідна пам'ять
String	Є два види рядків: - рядки фіксованої довжини мають до 2^{16} символів; - рядки змінної довжини мають до 2^{31} символів	10 байтів + 1 байт на символ при звичайному кодуванні і 2 байти на символ при Unicode кодуванні
Date	Зберігається, як Double, ціла частина зберігає число днів від початкової дати, а дробова частина зберігає час від півночі. Початкова дата – 30.12.1899.	8 байтів
Object	Посилання на об'єкт (покажчик)	4 байти
Variant	Універсальний тип, значенням якого можуть бути дані любого з перерахованих вище типів, об'єкти, значення NULL і значення помилок ERROR	4 байти залежить від контексту але не менше 16 байтів

Оголошення змінних

```
[Dim | Public | Private | Static] ім'я_змінної  
    [As ім'я_типу] [ім'я_змінної [As ім'я_типу]]
```

```
Dim B As Byte, i As Integer, L As Long
```

```
Dim Sng As Single, D As Double, c As Currency
```

```
Dim Sf As String, Sv As String, Dat As Date
```

```
Dim O As Object, V
```

Рівень процедури (локальні змінні)	Рівень модуля (глобальні змінні)
Dim Static	Dim Public Private

Option Explicit – забороняє використання змінних
без явного їх оголошення



Демонстрація VarAutoInitialization()

Оголошения констант

docs.microsoft.com/en-us/office/vba/language/reference/constants-visual-basic-for-applications

Microsoft

Docs

Documentation

Learn

Q&A

Code Samples

Search

Sign in

Office Add-ins Guides Office applications Office Dev Center Resources Developer Program Script Lab

Docs / Office VBA Reference / Language reference / Reference / Constants / Overview

Bookmark Edit Share

Filter by title

Language reference

Overview

Concepts

How-to topics

Reference

Overview

Character sets

Constants

Overview

Calendar

CallType

Color

Constants (Visual Basic for Applications)

12/11/2018 • 2 minutes to read •    

VBA constants

- Calendar
- CallType
- Color
- Comparison
- Date
- Date Format
- Dir, GetAttr, and SetAttr
- DriveType

Is this page helpful?

Yes

No

In this article

VBA constants

See also

Оператори VBA – арифметичні оператори

- ▶ додавання (+);
- ▶ віднімання (-);
- ▶ множення (*);
- ▶ ділення (/);
- ▶ піднесення до ступеня (^), наприклад, $2^3 = 8$ (результат має тип Double);
- ▶ цілочисельне ділення (\) - ділить перший операнд на другий, відкидаючи дробову частину, наприклад, $5 \setminus 2 = 2$;
- ▶ ділення по модулю (Mod) - ділить перше число на друге, повертаючи тільки залишок від ділення, наприклад, $5 \text{ Mod } 2 = 1$ (операція визначена над даними цілого типу та повертає результат цілого типу);
- ▶ унарного мінусу (-), унарного плюсу, як і інкременту, декременту та сумісних операцій +=, -=, *=, /= тощо в VBA не існує.
- ▶ *Оператор присвоювання в VBA - знак рівності (=).*



nVar = 10 VS If (nVar = 10)

Оператори VBA – оператори порівняння

- ▶ рівність (=), наприклад, `If (nVar = 10);`
- ▶ більше, ніж (>), наприклад, `If (nVar > 10);`
- ▶ менше, ніж (<), наприклад, `If (nVar < 10);`
- ▶ більше або дорівнює (>=), наприклад, `If (nVar >= 10);`
- ▶ менше або дорівнює (<=), наприклад, `If (nVar <= 10);`
- ▶ не дорівнює (<>), наприклад, `If (nVar <> 10);`
- ▶ оператор порівняння об'єктів (Is) -визначає, чи посилаються об'єктні змінні на той самий об'єкт або на різні об'єкти, наприклад, `If (obj1 is obj2);`
- ▶ оператор подібності (Like). порівнює і визначає, чи збігається строковий об'єкт з шаблоном.



Оператори VBA – логічні оператори

- ▶ And - логічне І, повинні бути істинними обидва умови, щоб оператор повернув True;
- ▶ OR - логічне АБО, має бути істинною. хоча б одна з умов, щоб оператор повернув True;
- ▶ Not - логічне заперечення, повертає True, якщо умова помилкова;
- ▶ Xor - логічний виняток. Операція $E1 \text{ Xor } E2$ повертає True, якщо тільки $E1 = \text{True}$ або тільки $E2 = \text{True}$, інакше - False;
- ▶ Eqv - еквівалентність двох виразів, повертає True, якщо вони мають однакове значення;
- ▶ Imp - імплікація, повертає False, якщо $E1 = \text{True}$ і $E2 = \text{False}$, інакше - True.

Пріоритети операторів VBA

- ▶ Оператори конкатенації. В VBA їх два + або &. Рекомендується завжди використовувати &

Пріоритет	Арифметичні	Порівняння	Логічні
1	^	= <> < > <= >= Like Is - всі операції мають однаковий пріоритет	Not
2	унарний мінус (-)		And
3	*, /		Or
4	\		Xor
5	Mod		Eqv
6	+, -		Imp
7	& (не є арифметичною)		

Оператори VBA – оператори присвоювання

- ▶ [Let] змінна = вираз

Демонстрація Assign1()

- ▶ Lset StringVar = вираз-рядок
Lset MyVar1 = MyVar2

Демонстрація Assign2(), Assign3()

- ▶ Rset StringVar = вираз-рядок

Демонстрація Assign4()



Оператори управління – умовний та вибору

► If умова Then
 [оператори1]
Else
 оператори2]
End If

Демонстрація If1()

► Select Case Вираз-тест
 [Case СписокВиразів-n
 [Оператори-n]]
 [Case Else
 [ІнакшеОператори]]
End Select

If умова Then
 [Оператори]
ElseIf умова-n Then
 [Оператори-n]
...
Else
 [ІнакшеОператори]]
End If

Вираз-тест може бути довільним виразом з числовим значенням або значенням-рядком

Демонстрація Case1()

Оператори управління - цикли

- ▶ For лічильник_циклу = початок To кінець [Step крок]
 тіло циклу
Next [лічильник_циклу]

Демонстрація for1()

- ▶ Do [{While | Until} умова]
 тіло циклу
Loop
- ▶ Do
 тіло циклу
Loop [{While | Until} умова]

Демонстрація Loop1()

- ▶ While умова
 тіло циклу
Wend
- ▶ For Each елемент In група
 тіло циклу

Next [елемент]

Демонстрація ForEach1()

Масиви – статичні та динамічні

► `[Dim | Public | Private | Static]`
`ім'я_змінної-масиву (список_розмірностей) [As`
`ім'я_типу]`

`Dim MyArray (0 To 2) As Integer`

`Dim MyArray (2) As Integer`

`MyArray (0) = 100`

однаковий масив
(если Option Base 0)

`Dim MyArray (4, 9)`

Демонстрація MyArray ()

► `Dim MyArray ()`
`MyArray = Array (100, 200, 300)` тип елементів `Variant/Integer`
`ReDim Preserve MyArray (5)`
`MyArray(3) = 400`

`Option Base 0 or 1` – декларація рівня модуля, що вказує значення за замовчуванням індексу нижньої межі масиву, якщо не використовується – вважається 0.

► **Демонстрація DinArray ()**

Функції для роботи з масивами

- ▶ **LBound (array_Name [, dimension])**
UBound (array_Name [, dimension])
array_Name - ім'я масиву,
dimension - ціле число (необов'язковий аргумент). Визначає вимір масиву, для якого треба отримати верхню чи нижню межу. При відсутності dimension повертається межа для першого виміру масиву.
- ▶ **Erase array_Name**
- ▶ **Filter(sourcearray, match, [include, [compare]])**
sourcearray - вихідний масив,
match - підрядок, який шукається у вихідному масиві,
include - Boolean-параметр, який вказує, чи будуть повернені тільки рядки, що включають match (True) або тільки ті, що не включають match (False).
compare - тип порівняння (0 - vbBinaryCompare - чутливий до регістру, 1 - vbTextCompare).



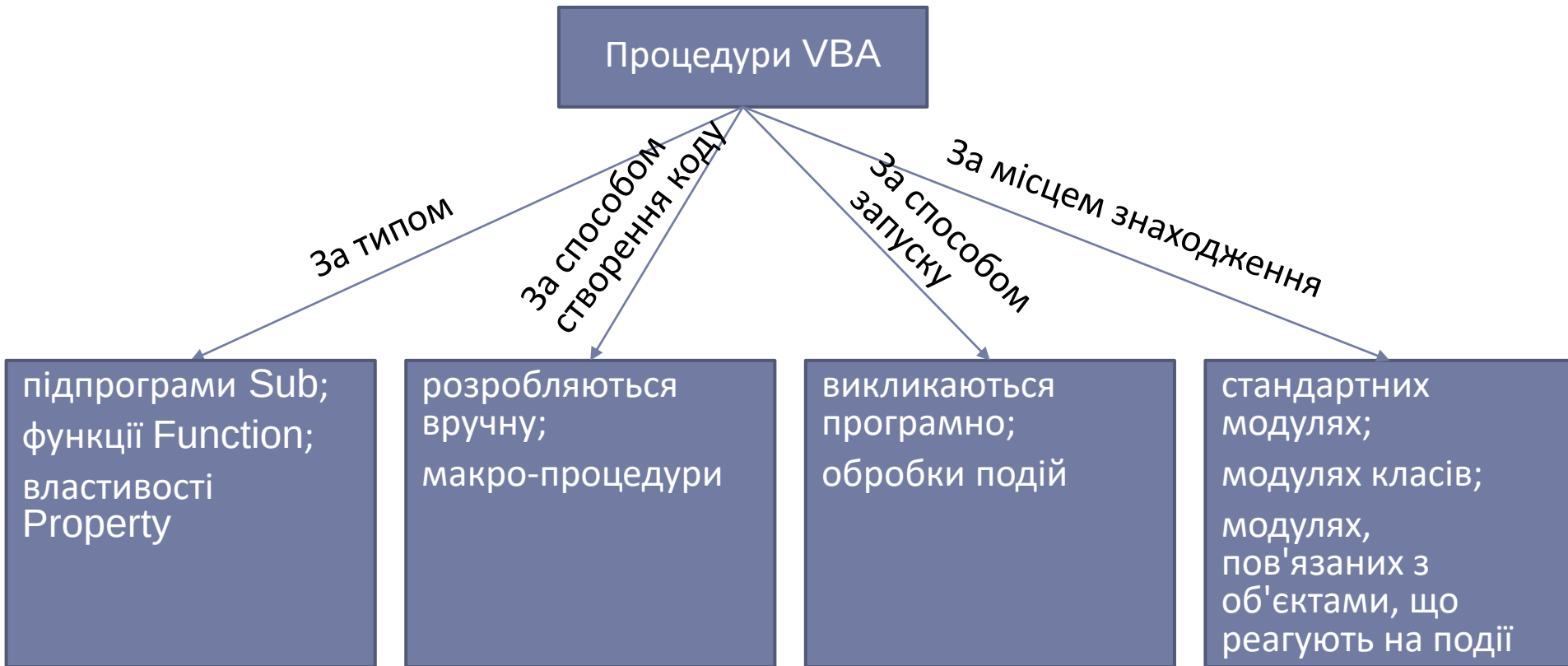
Функції для роботи з масивами

- ▶ **Split(expression, [delimiter, [limit, [compare]]])**
expression – вихідний рядок, який буде розбиватись.
delimiter – підрядок, який використовується як роздільник, сам роздільник не міститься в елементах масиву (часто використовується рядок з одного символу), за замовчуванням дорівнює пробілу " ".
limit – кількість підрядків, що будуть передані у результуючий масив, за замовчуванням дорівнює -1, це означає, що будуть передані всі підрядки.
compare – константа, що вказує на тип порівняння аналогічно функції Filter.
- ▶ **Join(sourcearray, [delimiter])**
sourcearray – масив рядків, які будуть об'єднуватись,
delimiter – підрядок, який використовується як роздільник, що буде додаватись у результуючий рядок між елементами масиву, за замовчуванням дорівнює пробілу " " .



Процедури VBA

- *Процедура* - це програмна одиниця VBA, що включає оператори опису її локальних даних і виконувані оператори.



Процедури VBA – підпрограми та функції

- **Процедура Sub (підпрограма)** являє собою набір команд, за допомогою яких можна вирішити певне завдання.

```
[Private | Public | Static] Sub ім'я ([список-  
параметрів])
```

тіло-процедури

```
End Sub
```

```
Option Private Module
```

- **Процедура Function (функція)** відрізняється від підпрограм тим, що обов'язково повертає одне результуюче значення.

```
[Private | Public | Static] Function ім'я ([список-  
аргументів]) As типЗначення
```

тіло-функції

```
End Function
```

локальна змінна, що повертає результат повинна збігатись з ім'ям функції

Існує можливість вийти з підпрограми або функції за допомогою команд

Exit Sub або Exit Function.

$$\prod_{k=1}^4 f_k + \prod \sin(f_k), \text{ де } f=\{5, -3, 1, 2\}$$

Демонстрація Sub Main() - Sub Product та Sub Main1() - Function SQ

Передача аргументів до підпрограм та функцій

- ▶ `Function fSum (nItem1 As Integer, nItem2 As Integer, Optional nItem3 As Integer)` - опціональні параметри
`nResult = fSum (3, 2, 1)`
`nResult = fSum (3, 2)`
`nResult = fSum (nItem1: = 3, nItem2: = 2)`
- ▶ Якщо параметр передається *за посиланням* (ByRef), то фактично в процедуру, що викликається, передається посилання на цю змінну в оперативній пам'яті і якщо значення змінної в процедурі змінюється, то воно зміниться і в процедурі, яка викликає дану процедуру. Це - прийнята в VBA поведінка за замовчуванням.
- ▶ Якщо параметри передаються *за значенням* (ByVal), то фактично в оперативній пам'яті створюється копія цієї змінної і ця копія передається процедурі, що викликається. Звичайно ж, щоб ви не зробили з цією копією, на значення первинної змінної у процедурі, яка викликає дану процедуру, це ніяк не вплине.

Демонстрація Sub TestProc() - Function fSum()

- ▶ Існує можливість вказати компілятору VBA, що нас не цікавить значення, що повертає функція. Для цього досить не укласти параметри функції у круглі дужки, наприклад: `MsgBox "Test"`

Функції обробки числових даних

- ▶ **Abs(число)** - абсолютне значення числа (модуль);
- ▶ **Atn(число)** - повертає арктангенс (в радіанах) аргументу;
- ▶ **Cos(число)** - косинус кута, аргумент число задає кут в радіанах;
- ▶ **Sin(число)** - синус кута, аргумент число задає кут в радіанах;
- ▶ **Exp(число)** - експонента (результат піднесення до ступеня числа e);
- ▶ **Log(число)** - натуральний логарифм числа;
- ▶ **Rnd[(число)]** – повертає рівномірно розподілене випадкове число в інтервалі [0-1], не включаючи 1 (необхідно викликати функцію Randomize).
- ▶ **Sng(число)** - знак числа (якщо число більше нуля - 1, дорівнює нулю - 0, менше нуля - -1);
- ▶ **Sqr(число)** - квадратний корінь;
- ▶ **Tan(число)** - тангенс кута, аргумент число задає кут в радіанах.
- ▶ **Int()**, **Fix()** і **Round()** дозволяють по різному округляти числа: Int() повертає найближче менше ціле, Fix() відкидає дробову частину, Round() округлює до зазначеної кількості знаків після коми (краще використовувати Format()).
- ▶ **Демонстрація RoundNumbers()**

Функції перетворення та перевірки типів даних

- ▶ CBool(), CByte(), CCur(), CDate(), CDBl(), CDec(), CInt(), CLng(), CSng(), CStr(), CVar(), CVDate(), CVerErr(). Переглянути роботу цих функцій зручно, використавши функцію TypeName().
- ▶ Str() - дозволяє перевести числове значення в строкове. Робить майже те ж саме, що і CStr(), але при цьому вставляє пропуск попереду для позитивних чисел.
- ▶ Val() - "витягує" з суміші цифр і букв тільки числове значення. При цьому ця функція читає дані зліва направо і зупиняється на першому нечисловому значенні.
- ▶ IsNumeric(), IsDate(), IsArray(), IsEmpty(), IsError(), IsMissing(), IsNull(), IsObject().
- ▶ Hex(), Oct() - перетворюють десяткові дані в строкове представлення шістнадцятиричних і вісімкових значень

(&O12, &NA)

Демонстрація ConvertType()

Run-time помилка Type mismatch (невідповідність типу)

Функції обробки рядків

- ▶ Операція *конкатенації* використовується для зчеплення двох або більше рядків. Позначається дана операція знаком «+» або знаком «&». У разі, якщо застосовується перший знак, то в якості аргументів виразу, що визначає операцію конкатенації, повинні виступати змінні або константи строкового типу. У разі застосування другого знаку - один з аргументів може бути змінною або константою типу число або дата
- ▶ При порівнянні текстових рядків за замовчуванням порівнюються просто двійкові коди символів - які більше або менше. Якщо потрібно використовувати той порядок, який йде в алфавіті, необхідно на початку модуля помістити інструкцію:

`Option Compare Text`

За замовчуванням працює режим

`Option Compare Binary`

В останньому випадку виконується порівняння з урахуванням регістру символів.

Функції обробки рядків

- ▶ `рядок Like рядок_шаблон`

рядок може включати символи підстановки

Символ підстановки	Значення
#	Будь-яка цифра (тільки одна) від 0 до 9
*	Будь-яка кількість будь-яких символів (включаючи нульовий)
?	Будь-який символ (тільки один)
[a, b, c]	Будь-який символ (тільки один) з наведеного списку
[! a, b, c]	Будь-який символ (тільки один), окрім наведених у списку

Якщо рядок збігається з `рядок_шаблон`-ом, результат - `True`; якщо ні - `False`. Якщо або рядок, або шаблон є `Null`, оператор повертає `Null`. Демонстрація `LikeStudy()`

- ▶ `ASC(string)` повертає числовий код для першого символу рядка-параметра
- ▶ `Chr(char_code)` повертає символ за його числовим кодом

▶ Демонстрація `PrintLetters()`

Функції обробки рядків

- ▶ `InStr ([start,] string1, string2 [, compare])` визначає позицію першого входження одного рядка всередині іншого рядка при пошуку зліва, у разі якщо рядок не знайдено, повертається 0.
- ▶ `InStrRev (string1, string2 [start,] [, compare])` визначає позицію першого входження одного рядка всередині іншого рядка при пошуку праворуч, у разі якщо рядок не знайдено, повертається 0.
- ▶ `Len(string)` повертає довжину рядка (число символів), який заданий аргументом `string`.
- ▶ `Left(string, length)` повертає з рядку `string` вказане число `length` символів зліва.
- ▶ `Right(string, length)` повертає з рядку `string` вказане число `length` символів праворуч.
- ▶ `Mid(string, start [, length])` повертає з рядку `string` підрядок довжиною `length`, починаючи з позиції `start`.
- ▶ `Ltrim(string)`, `Rtrim(string)`, `Trim(string)` повертають копію рядка-параметра, з якого видалені пробіли, що знаходяться на початку рядка, в кінці рядка або на початку і в кінці рядка, відповідно.

Демонстрація `InStrStudy()`

Функції обробки рядків

- ▶ `LCase(string)`, `UCase(string)` повертають копію рядка-параметра, символи якого перетворені до нижнього або до верхнього регістру, відповідно.
- ▶ `Replace(string, find, replace [, start] [,count] [, compare])` повертає рядок, у якому послідовність символів вихідного рядку замінено іншою послідовністю символів:
- ▶ `Space(number)` повертає рядок із зазначеною кількістю пробілів `number`;
- ▶ `String(number, character)` дозволяє отримати рядок із зазначеною кількістю `number` символів `character`;
- ▶ `Spc(number)` вставляє `number` пробілів перед відображенням на екрані або принтері наступного виразу у списку.
- ▶ `StrComp(string1, string2 [, compare])` дозволяє порівнювати два рядки, повертаючи `-1`, якщо `string1 < string2`, `1`, якщо `string1 > string2`, `0`, якщо вони однакові.
- ▶ Функція `StrReverse(string)` "перегортає" рядок `string`, розмістивши його символи в зворотному порядку.
- ▶ `Tab([number])` розмножує символи табуляції у кількості `number`.

Робота з датами і часом - присвоєння значень

- ▶ Представлення дати та часу займає 8 байтів пам'яті і зберігається, як Double. При цьому ціла частина зберігає число днів від початкової дати, а дробова частина зберігає час від півночі. Початкова дата – 30.12.1899.
- ▶ При присвоєнні значень типу Date слід укласти дату в спеціальні обмежники "# " (існує можливість конвертувати дату з рядка функцією CDate).

```
Dim dat1 As Date, dat2 As Date  
    dat1 = #2/17/2021#  
    dat2 = CDate("2/17/2021")
```

Демонстрація DateAssign ()



Робота з датами і часом – вбудовані функції

- ▶ `Date` - повертає `Variant (Date)` - поточну дату з календаря комп'ютера;
- ▶ `Time` - повертає `Variant (Date)` - поточний час з годинника комп'ютера;
- ▶ `Now` - повертає `Variant (Date)` - поточну дату і час з календаря та годинника комп'ютера.
- ▶ `DateAdd(interval, number, data)` - додає/віднімає вказаний часовий інтервал зі значення дати. `interval` - рядок, який вказує тип значення часового інтервалу, що додається, `number` - кількість часових інтервалів, на які треба змінити дату, `data` - дата, до якої додається зазначений часовий інтервал. Допустимі значення аргументу `interval` наведені в Табл.

Значення	Опис
y	рік
q	квартал
m	місяць
y	день року
d	день місяця

Значення	Опис
w	день тижня
ww	тиждень
h	години
n	хвилини
s	секунди

Робота з датами і часом – вбудовані функції

- ▶ `DateDiff (interval, date1, date2 [, firstdayofweek [, firstweekofyear]])` - визначає час, що минув між двома датами. `interval` задає тип часового інтервалу при обчисленні різниці між датами `date1` і `date2`, `firstdayofweek` - константа, яка вказує перший день тижня (за замовчуванням вважається, що тиждень починається з неділі), `firstweekofyear` - константа, яка вказує перший тиждень року (за замовчуванням першим тижнем вважається тиждень, що містить перше січня).
- ▶ `DatePart (interval, date [, firstdayofweek, [, firstweekofyear]])` - повертає `Variant (Integer)` з зазначеною частиною дати (наприклад, тільки рік, тільки місяць або тільки день тижня, параметри такі ж, як у `DateDiff`).
- ▶ `DateSerial(year, month, day)` - обчислює значення дати типу `Variant (Date)` за її компонентами - роком, місяцем, днем. Значення кожного аргументу має лежати у відповідному діапазоні: 100–9999 для року, 1–31 для днів, 1–12 для місяців.
- ▶ `DateValue(date)` переводить аргумент-рядок в дату.



Робота з датами і часом – вбудовані функції

- ▶ `Day (date)`, `Month (date)`, `Year (date)` - з аргумента-дати повертають, відповідно, номер дня, місяця і року.
- ▶ `Hour (time)`, `Minute (time)`, `Second (time)` - з аргумента-часу, який є числовим виразом або рядком, повертають, відповідно, кількість годин, хвилин і секунд.
- ▶ `MonthName (month, [abbreviate])` - повертає ім'я місяці рядком по його номеру. Значення, що повертається залежить від регіональних налаштувань.
- ▶ `TimeSerial (hour, minute, second)` - обчислює результат типу `Variant (Date)`, що містить значення часу, відповідне до вказаних години, хвилини і секунди.
- ▶ `TimeValue (time)` - повертає значення типу `Variant (Date)`, що містить час. Аргумент `time` задається строковим виразом, що представляє час від 0:00:00 (12:00:00 AM) до 23:59:59 (11:59:59 PM).
- ▶ `Timer` - повертає кількість секунд, що пройшли з півночі.

▶ Демонстрація `DateFunctions ()`

Функції форматування даних – вбудовані функції

► `Format (Expression [, Format[, Firstdayofweek [, Firstweekofyear]])`

`Expression` - будь-який припустимий вираз (обов'язковий аргумент).

`Format` - назва іменованого або визначеного користувачем формату.

`FirstDayOfWeek` - константа, яка визначає перший день тижня.

`FirstWeekOfYear` - константа, яка визначає перший тиждень року.

Іменований формат	Дія
General Date	Форматує інформацію про дату і час в послідовне число дати, використовуючи установки формату дати і часу для даного комп'ютера.
Long Date	Форматує в послідовну дату тільки частину, яка містить дату, використовуючи установки комп'ютера для Long-формату дати.
Medium Date	Форматує в послідовну дату тільки частину, яка містить дату, використовуючи установки комп'ютера для Medium -формату дати.
Short Date	Форматує в послідовну дату тільки частину, яка містить дату, використовуючи установки комп'ютера для Short -формату дати.
Long Time	Форматує в послідовну дату тільки частину, яка містить час, використовуючи установки комп'ютера для Long-формату часу.
Medium Time	Форматує в послідовну дату тільки частину, яка містить час, використовуючи установки комп'ютера для Medium -формату часу.

Функції форматування даних – вбудовані функції

Іменованій формат	Дія
Short Time	Форматує в послідовну дату тільки частину, яка містить час, використовуючи установки комп'ютера для Short -формату часу.
General Number	Форматує число в рядок без будь-яких особливих символів.
Currency	Форматує число з символом грошової одиниці, роздільником тисяч і тільки двома десятковими розрядами.
Fixed	Форматує число так, щоб завжди була, як найменш, одна цифра перед десятковим роздільником і завжди дві цифри після нього.
Standard	Форматує число з роздільником тисяч так, щоб була, як найменш, одна цифра перед десятковим роздільником і завжди дві цифри після нього.
Percent	Форматує число як процентне відношення, помножуючи його на 100 і додаючи символ відсотка.
Scientific	Форматує число в звичайний експонентний формат.
Yes/No	Повертається рядок "Да" якщо число, що форматується, не дорівнює нулю, інакше - "Нет".
True/False	Повертається рядок "Истина" якщо число, що форматується, не дорівнює нулю, інакше - "Ложь".
On/Off	Повертається рядок "Вкл" якщо число, що форматується, не дорівнює нулю, інакше - "Выкл".

Демонстрація FormatOptions ()

Функції форматування даних – символи-заповнювачі користувацьких форматів

Символ-заповнювач	Дія (як приклад використовується число 1234.45)
0	Цифровий символ, відображає цифру, якщо така знаходиться в цій позиції, або 0, якщо - ні. Можна використовувати символ 0 для відображення початкових нулів для цілих чисел і кінцевих нулів в десяткових дробах: "00000.000" відображає 01234.450
#	Цифровий символ, відображає цифру, якщо така знаходиться в цій позиції, інакше - не відображує нічого. Символ-заповнювач # еквівалентний 0, крім того, що початкові і кінцеві нулі не відображаються: "#####.###" відображає 1234.45
\$	Відображає знак долара: "\$###,###.00" відображає \$1 234,45
.	Десятковий символ-заповнювач, відображає десяткову точку в позначеній позиції в рядку символів-заповнювачів 0: "#####.#" відображає 1234.5
%	Символ відсотка, примножує значення на 100 і додає знак відсотка в позицію, зазначену символом-заповнювачем %: "##0.00%" відображає число 0.12345 як 12.35% (відбувається округлення)
,	Роздільник тисяч, додає встановлені в регіональних налаштуваннях символи як роздільники тисяч: "###,###,###.00" відображає 1 234,45



Функції форматування даних – символи-заповнювачі користувацьких форматів

Символ-заповнювач	Дія (як приклад використовується число 1234.45)
E- e-	Відображає значення в експоненційному форматі зі знаком порядку тільки для негативних значень: "#.####E-00" відображає 1.23445E03
E+ e+	Відображає значення в експоненційному форматі зі знаком порядку як для негативних, так і для позитивних значень: "#.####E+00" відображає 1.23445E+03
/	Відокремлює день, місяць і рік для форматування значень дат.
m	Вказує, як відображати місяці в датах: m відображає 1, mm - 01, mmm - янв, mmmm – Январь
d	Вказує, як відображати дні в датах: d відображає 1, dd - 01, ddd - Вс, dddd - воскресенье
y	Показує день року як число від 1 до 366
yy	Вказує, як відображати роки в датах: yy - 21, yyyy - 2021
q	Відображає квартал року як число від 1 до 4
w	Показує день тижня як число (1 - неділя)
ww	Відображає тиждень року як число від 1 до 54
:	Відокремлює години, хвилини і секунди в значеннях формату часу: hh:mm:ss - 1:01:01, h:m:s - 1:1:1

Функції форматування даних – символи-заповнювачі користувацьких форматів

Символ-заповнювач	Дія (як приклад використовується число 1234.45)
h	Вказує, як відображати години: hh - 01, h - 1
n	Вказує, як відображати хвилини: nn - 01, n - 1
s	Вказує, як відображати секунди: ss - 01, s - 1
AM/PM	Показує час у 12-годинному форматі часу з доданими AM або PM
@	Символьний заповнювач, відображає пробіл, якщо у поточній позиції немає символу в рядку, що форматується.
<	Показує всі символи в нижньому регістрі.
>	Показує всі символи в верхньому регістрі.



Функції для організації взаємодії з користувачем – вбудовані функції

```
► MsgBox (Prompt [, Buttons] [, Title] [, HelpFile,  
Context])
```

Prompt – рядок з інформацією, яка виводиться в діалоговому вікні.

Title - рядок для заголовка діалогового вікна.

HelpFile - рядок, що містить ім'я довідкового файлу (який вже повинен бути створений).

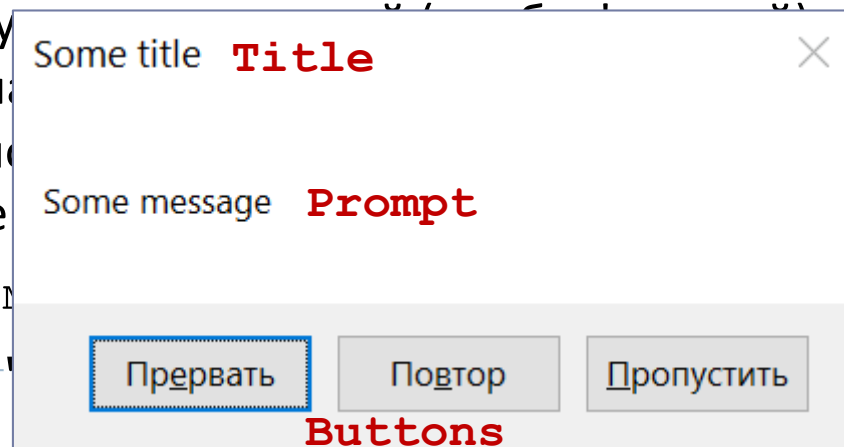
Context - чисельний вираз, який вказує розділ в довідковому файлі.

Buttons - чисельний вираз, який задає кількість і тип кнопок у діалоговому вікні. Також вказує кнопку за замовчуванням і, чи містить це діалогове вікно стандартні значки Windows для попереджувальних повідомлень і запитів користувача.

Слід звернути увагу, що `add_argument` можна викликати декілька разів, щоб додати декілька аргументів. Якщо потрібно додати аргумент, який має певні параметри, можна використати `add_argument` з параметрами `type` та `choices`. Наприклад, якщо потрібно додати аргумент, який має певні параметри, можна використати `add_argument` з параметрами `type` та `choices`. Наприклад, якщо потрібно додати аргумент, який має певні параметри, можна використати `add_argument` з параметрами `type` та `choices`.

► MsgBox "Повідомлення"

```
MsgBox Title:='
```



Константи аргумента Buttons функції MsgBox

Константа	Призначення
vbAbortRetryIgnore	Відображає командні кнопки Прервать, Повтор, Пропустить
vbApplicationModal	Користувач повинен відповісти на запит діалогового вікна для продовження роботи з додатком (до надання відповіді додаток призупиняється)
vbCritical	Відображається вікно з попереджувальним значком критичної помилки 
vbDefaultButton1 vbDefaultButton2 vbDefaultButton3 vbDefaultButton4	Номер командної кнопки в діалоговому вікні, яка є кнопкою за замовчуванням.
vbExclamation	Відображається вікно з попереджувальним значком  Використовується у вікні з інформацією, яка не вимагає відповіді.
vbInformation	Відображається вікно з попереджувальним значком  Використовується у вікні з довідковою інформацією.
vbMsgBoxHelpButton	Додає до діалогового вікна кнопку Справка. При натисканні на цю кнопку відкривається файл, який заданий в аргументі HelpFile, в розділі, заданому аргументом Context
vbOKCancel	Відображає кнопки ОК і Отмена.
vbOKOnly	Відображає тільки кнопку ОК - аналогічна пропуску аргументу Buttons.

Константи аргумента Buttons функції MsgBox

Константа	Призначення
vbQuestion	Відображається вікно з попереджувальним значком  . Використовується разом з повідомленням-запитом.
vbRetryCancel	Відображає кнопки Повтор і Отмена.
vbSystemModal	Користувач повинен відповісти на запит діалогового вікна для продовження роботи з додатком (до надання відповіді всі відкриті додатки призупиняються, діалогове вікно завжди залишається попереду інших вікон доки не буде закрито).
vbYesNo	Відображає кнопки Да і Нет.
vbYesNoCancel	Відображає кнопки Да, Нет і Отмена.

Значення констант, що повертаються функцією MsgBox

Назва константи	Значення константи	Призначення
vbAbort	3	Користувач обрав кнопку Прервать
vbCancel	2	Користувач обрав кнопку Отмена
vbIgnore	5	Користувач обрав кнопку Пропустить
vbNo	7	Користувач обрав кнопку Нет
vbOK	1	Користувач обрав кнопку ОК
vbRetry	4	Користувач обрав кнопку Повторить
vbYes	6	Користувач обрав кнопку Да

Демонстрація
MsgBoxOptions ()

Функції для організації взаємодії з користувачем – вбудовані функції

- ▶ `InputBox (Prompt [, Title] [, Default] [, XPos] [, YPos] [, HelpFile, Context])`

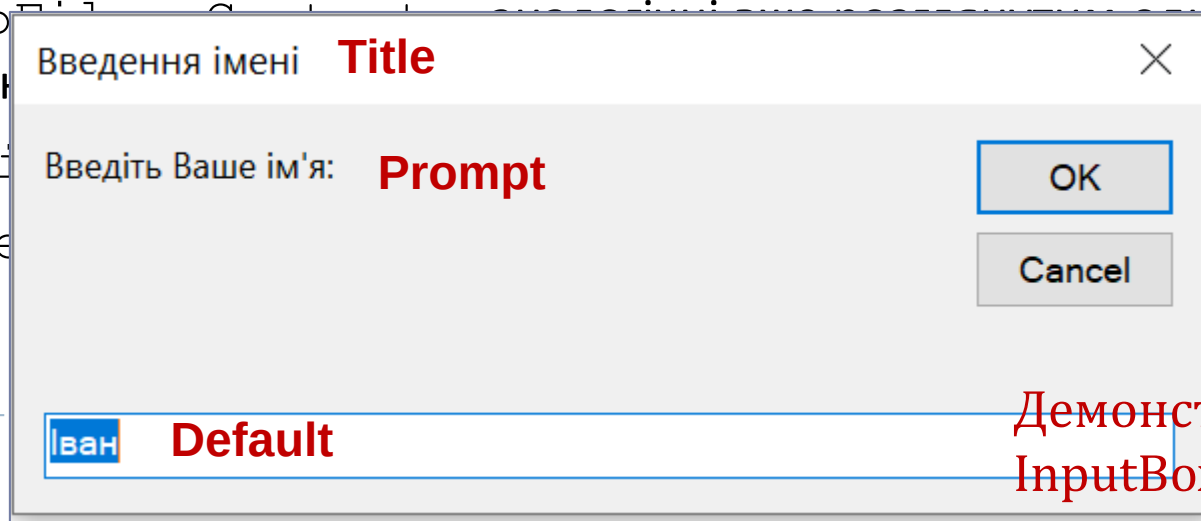
`Prompt` – рядок з інформацією, яка підказує користувачеві, що потрібно ввести у діалоговому вікні.

`Default` - вираз-рядок, що використовується як значення за замовчуванням для введення користувача.

`XPos`, `YPos` - чисельні вирази, що задають координати виведення діалогового вікна на екрані монітора. `XPos` - горизонтальна відстань від лівого краю вікна. `YPos` - вертикальна відстань від верхнього краю вікна. Обидва відстані вимірюються в *twip*х (1 *twip* = 1/20 точки).

`Title`, `HelpFile`, `Context` - вирази, що задають заголовок, файл допомоги та контекстний аргумент функції.

- ▶ `MsgBox "Повідомлення"`
`MsgBox Title`



Демонстрація
`InputBoxOptions ()`

Функції-замінники синтаксичних конструкцій – вбудовані функції

- ▶ `Choose` - приймає число (номер значення) і кілька значень. Повертає значення, порядковий номер якого відповідає переданому числу.
- ▶ `IIF` - розшифровується як *Immediate IF* (Негайний *IF*). Являє собою спрощений варіант `IF`, коли перевіряється умова і повертається одне з двох значень.
- ▶ `Switch` - приймає необмежену кількість пар типу вираз/значення, перевіряє кожний вираз на істинність і повертає значення для першого виразу, який виявився істинним, наприклад:

Демонстрація `FuncReplacers ()`



Функції для роботи з файловою системою

– вбудовані функції

- ▶ `Open <шлях> for <режим>` - відкриває файл по зазначеному шляху з відповідним режимом: `Input`, `Output`, `Append` - послідовне читання, запис та додання у текстовий файл, `Random` - читання та запис записів (records), `Binary` – читання та запис у довільну байтову позицію бінарного файлу.
- ▶ `Print #filename, [ outputlist ]`
`Write #filename, [ outputlist ]` – вставляє коми та `vbCrLf`
`#filename` – цілочисельний номер файлу,
`outputlist` - один або декілька, відокремлених комами рядків або чисел, які записуватимуться у файл.
- ▶ `Input #filename, varlist`
`Line Input #filename, varname`
перший оператор зчитує підрядки, розділені комами (зазвичай вони пишуться в файл оператором `Write #`), а другий - зчитує увесь рядок до символу кінця рядку (`vbCrLf`).
`#filename` - цілочисельний номер файлу,
`varlist` - одна або декілька, відокремлених комами змінних, у які буде записуватись зчитані з файлу підрядки,
`varname` - змінна `String` або `Variant`, у яку буде зчитуватись рядок з файлу.

Функції для роботи з файловою системою

– вбудовані функції

- ▶ При відкритті файлу у режимах `Random` або `Binary` для запису та читання даних використовуються оператори:
`Put [#]filenumber, [recnumber або bytenumber], varname`
`Get [#]filenumber, [recnumber або bytenumber], varname`
- ▶ `FreeFile(rangenummer)` - функція, яка дозволяє визначити наступне вільне цілочисельне значення, яке можна використовувати як номер файлу при його відкритті, при `rangenummer=0` – від 0 до 255, при `rangenummer=1` – від 256 до 511.
- ▶ `FileAttr(filenumber, returntype)` - функція, яка дозволяє визначити, як саме був відкритий файл, зазначений параметром `filenumber` (параметр `returntype` повинен бути встановлений у 1). Функція повертає ціле число, яке вказує на режим відкриття файлу: `Input (1)`, `Output (2)`, `Random (4)`, `Append (8)`, `Binary (32)`.
- ▶ `LOF(filenumber)` - від *length of file* - функція, яка повертає довжину відкритого файлу в байтах.
- ▶ `Loc (filenumber)` - функція, яка повертає число, що визначає для режимів `Random` та `Binary` поточне місце (номер запису або номер байту) вставки або читання у відкритому файлі.

Функції для роботи з файловою системою

– вбудовані функції

- ▶ `Seek (filenumber)` - функція, яка повертає номер (позиції байту - для режимів `Input`, `Output`, `Append`, `Binary` або номеру запису - для режиму `Random`), з якої буде виконуватися наступна операція читання або вставки.
- ▶ `EOF (filenumber)` - при запису в файл ця функція поверне `True` в кінці файлу. Використовується при запису в файл свого власного формату.
- ▶ `Close [filenumberlist]` - функція, яка закриває файл з указаним номером `#N`, можуть бути вказані декілька номерів через кому.
- ▶ `Reset` - функція, яка закриває всі файли, відкриті функцією `Open`.
- ▶ Описані функції читання-запису доцільно використовувати зі звичайними текстовими файлами наприклад, логами і т.і. Для читання-запису файлів додатків Microsoft Office (документів Word, таблиць Excel і т.і.) треба використовувати засоби, які надають об'єктні моделі цих додатків.

Функції для роботи з файловою системою

– вбудовані функції

- ▶ `CurDir[ (drive) ]` - повертає рядок-шлях до поточного каталогу, в якому будуть зберігатися файли нашого додатку за замовчуванням, може використовувати параметр-рядок `drive` зі значенням імені диску.
- ▶ `Dir[ (pathname, [attributes]) ]` - повертає рядок, що представляє ім'я файлу або каталогу, який відповідає вказаному шаблону повного шляху до файлу (каталогу) `pathname` або атрибуту файлу `attributes`, або мітці тому диску. `Dir` повертає перше ім'я файлу, яке відповідає `pathname`. Щоб отримати додаткові імена файлів, що відповідають `pathname`, викличте `Dir` знову без аргументів. Коли більше імена файлів не збігаються з `pathname`, `Dir` повертає рядок нульової довжини (""). `Dir` підтримує використання (*) для позначення декількох довільних символів та (?) - для позначення одного довільного символу з метою вказівки декількох файлів. Для перебору всіх файлів у каталозі вкажіть порожній рядок `Dir ( "" )`.
- ▶ `ChDir path` - оператор, який змінює поточний каталог на зазначений параметром `path`.

Функції для роботи з файловою системою

– вбудовані функції

- ▶ `MkDir path` - оператор, який створює каталог у зазначеному місці файлової системи, якщо він не існує.
- ▶ `RmDir path` - оператор, який видаляє зазначений каталог, якщо він пустий.
- ▶ `Kill pathname` - оператор, який видаляє один або більше зазначених файлів. `Kill` підтримує використання (*) для позначення декількох довільних символів та (?) - для позначення одного довільного символу з метою вказівки декількох файлів.
- ▶ `FileCopy source, destination` - оператор, який копіює файл на шляху `source` у файл на шляху `destination`.
- ▶ `Name oldpathname As newpathname` - оператор, який перейменовує (та/або переміщує) зазначений параметром `oldpathname` файл у зазначене параметром `newpathname` нове ім'я (та/або місце розміщення).
- ▶ `FileLen (pathname)` - функція, яка повертає довжину файлу-параметру в байтах.
- ▶ `FileDateTime (pathname)` - функція, яка дозволяє отримати інформацію про останні дату та час звернення до зазначеного як параметр файлу. Якщо до файлу після створення жодного разу не зверталися, то повернеться дата та час створення файлу.

Функції для роботи з файловою системою

– вбудовані функції

- ▶ `GetAttr (pathname)` - функція, яка повертає цілочисельне значення, у якому кодується інформація про атрибути файлу-параметру (прихований, доступний тільки для читання, архівний і т.і.).
- ▶ `SetAttr pathname, attributes` - оператор, який дозволяє встановити значення атрибутів файлу (тих же, що і для `GetAttr`).

Константи, що кодують атрибути файлу

Ім'я константи	Значення константи	Опис атрибуту
<code>vbNormal</code>	0	звичайний (Normal)
<code>vbReadOnly</code>	1	доступний тільки для читання (Read-only)
<code>vbHidden</code>	2	прихований (Hidden)
<code>vbSystem</code>	4	системний (System)
<code>vbDirectory</code>	16	Каталог
<code>vbArchive</code>	32	Архівний (Файл змінився після останньої резервної копії)
<code>vbAlias</code>	64	Вказане ім'я файлу є псевдонімом (alias). Доступно лише на Macintosh



Демонстрація `DirFileAction ()`

Інші функції – вбудовані функції

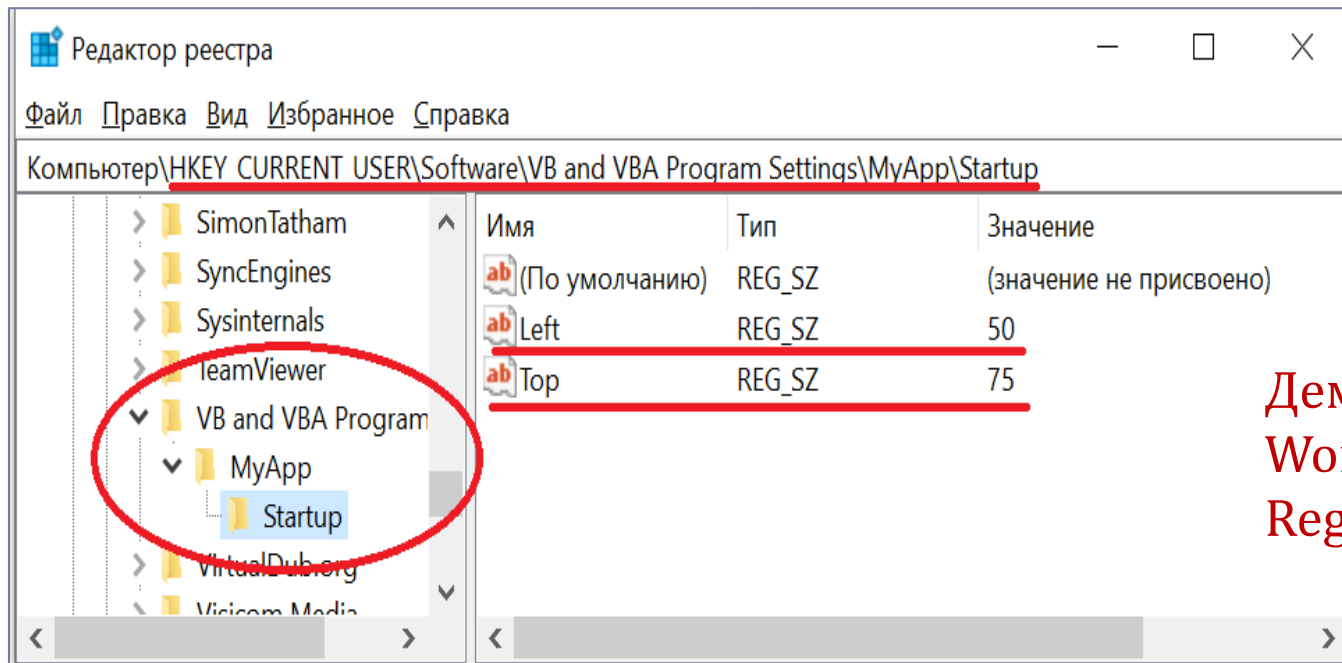
- ▶ `DoEvents()` - дозволяє передати управління операційній системі, щоб обробити події, які накопичилися в операційній системі (наприклад, натискання клавіш користувачем).
- ▶ `Environ({envstring | number})` - повертає абсолютний шлях для змінних оточення комп'ютера `envstring`. Наприклад, абсолютний шлях до тимчасового каталогу можна отримати так:

```
MsgBox Environ("TEMP")
```
- ▶ `SaveSetting appname, section, key, setting` – дозволяє записати до розділу реєстру `section` параметри налаштування додатку `appname` у вигляді ключ `key` - значення `setting`.
- ▶ `GetSetting(appname, section, key, [default])` дозволяє отримати значення параметра-налаштування додатку `appname` з зазначеного розділу `section` реєстру по ключу `key`. Параметр `default` містить значення параметра-налаштування за замовчуванням, воно повертається функцією `GetSetting` у разі, якщо для вказаного ключа значення не встановлене.



Інші функції – вбудовані функції

- ▶ `GetAllSettings (appname, section)` - дозволяє отримати (у вигляді двовірного масиву) з розділу реєстру, зазначеного параметром `section`, всі параметри, які відносяться до зазначеного параметром `appname` додатку.
- ▶ `DeleteSetting appname, section, [key]` - дозволяє видалити з розділу реєстру `section` певні налаштування додатку `appname`, вказавши ключ `key` параметра-налаштування. Якщо ключ не вказати, будуть видалені всі параметри-налаштування додатку `appname` з зазначеного розділу `section` реєстру.



татися тільки до
and VBAProgram

Демонстрація
WorkWithAppSettingsIn
Registry

Інші функції – вбудовані функції

- ▶ `RGB(red, green, blue)` - функція для роботи з кольором. Дозволяє повернути колірний код, який можна використовувати для присвоєння кольору в кодї, прийнявши три значення для квітів: червоного (`red`), зеленого (`green`) і синього (`blue`). Значення для кожного з основних кольорів можуть варіюватися від 0 до 255.
- ▶ `Shell(pathname, [windowstyle])` - функція, яка дозволяє запустити з VBA зовнішній програмний файл і повернути інформацію про його `ProgramID` в операційній системі. Необов'язковий параметр `windowstyle` - цілочисельна константа, що визначає як буде запускатись зовнішня програма (у відкритому або згорнутому вікні, з передачею фокуса до цього вікна або ні і т.і.). Наприклад оператор:

```
programId = Shell("calc.exe", vbNormalFocus)
```

запустить додаток Windows Калькулятор у відкритому вікні з передачею йому фокуса.



Інші функції – вбудовані функції

- ▶ `TypeName (varname)` - функція, яка повертає ім'я типу даних переданої їй змінної.
- ▶ `VarType (varname)` - робить майже те ж саме, але замість імені повертає числовий код, який позначає тип даних. Можна використовувати для програмних перевірок типів даних для змінних.

Константи, що визначають типи даних

Назва константи	Значення константи	Опис
<code>vbEmpty</code>	0	Empty (не ініціалізована)
<code>vbNull</code>	1	Null (не дійсні дані)
<code>vbInteger</code>	2	Ціле
<code>vbLong</code>	3	Довге ціле
<code>vbSingle</code>	4	З плаваючою крапкою одинарної точності
<code>vbDouble</code>	5	З плаваючою крапкою подвійної точності
<code>vbCurrency</code>	6	Значення валюти
<code>vbDate</code>	7	Значення дати
<code>vbString</code>	8	Рядок
<code>vbObject</code>	9	Object
<code>vbError</code>	10	Значення помилки
<code>vbBoolean</code>	11	Булеве

Інші функції – вбудовані функції

Константи, що визначають типи даних

Назва константи	Значення константи	Опис
vbVariant	12	Variant (використовується тільки з масивом Variant)
vbDataObject	13	Об'єкт доступу до даних
vbDecimal	14	Decimal значення
vbByte	17	Значення байту
vbLongLong	20	LongLong ціле (тільки для 64-бітових платформ)
vbUserDefine dType	36	Variant, що містить тип визначений користувачем
vbArray	8192	Масив (завжди супроводжується іншою константою)

Демонстрація
[WorkWithAppSettingsInRegistry](#)

