



Програмування у Visual Basic

д.е.н., професор Ставицький А.В.



План

- Редактор VB
- Змінні VB
- Процедури та функції
- Оператори та цикли
- Робота з файлами
- Форма та її елементи у VB
- Основні об'єкти програм MS Office


















Мова VBA

- При роботі з MS Office часто потрібно розв'язувати нестандартні задачі або автоматизувати процес обробки об'єктів. Частково цю проблему вирішують макроси, проте написати велику програму за їх допомогою досить важко.
- В таких випадках користуються мовою Visual Basic (VBA), яка дозволяє створювати професійно оформлені проєкти, автоматизувати операції з програмами, здійснювати обробку інформації, працювати лише з необхідними даними. Кожна з програм MS Office підтримує мову програмування VBA.
- За допомогою мови VBA можна створювати надзвичайно потужні та корисні інструменти для аналізу та обробки інформації з мінімальними витратами на навчання персоналу та обладнання робочого місця.



TIOBE Index for October 2023

Oct 2023	Oct 2022	Change	Programming Language		Ratings	Change
1	1			Python	14.82%	-2.25%
2	2			C	12.08%	-3.13%
3	4	▲		C++	10.67%	+0.74%
4	3	▼		Java	8.92%	-3.92%
5	5			C#	7.71%	+3.29%
6	7	▲		JavaScript	2.91%	+0.17%
7	6	▼		Visual Basic	2.13%	-1.82%
8	9	▲		PHP	1.90%	-0.14%
9	10	▲		SQL	1.78%	+0.00%
10	8	▼		Assembly language	1.64%	-0.75%
11	11			Go	1.37%	+0.10%
12	23	▲▲		Scratch	1.37%	+0.69%
13	18	▲		Delphi/Object Pascal	1.30%	+0.46%
14	14			MATLAB	1.27%	+0.09%
15	15			Swift	1.07%	+0.02%
16	19	▲		Fortran	1.02%	+0.23%
17	12	▼▼		R	0.96%	-0.26%
18	28	▲▲		Kotlin	0.96%	+0.53%

www.andriy-stav.cc.ua

Джерело:

<https://www.tiobe.com/tiobe-index/>



Приклад

- [Інтерфейс інформаційної системи](#)



1. Редактор VB



Запуск редактора

- Для запуску редактора VB з будь-якої програмами MS Office необхідно натиснути Alt-F11 і зайти в програму редагування макросів. Відкриється спеціальний редактор, що дозволяє створювати модулі (програмні блоки) та форми (вікна, що містять необхідну користувачеві інформацію).



Microsoft Visual Basic for Applications - Книга1

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

Project - VBAProject

- atpvbaen.xls (ATPVBAEN.XLAM)
- Solver (SOLVER.XLAM)
- VBAProject (FUNCRES.XLAM)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - Лист1

Лист1 Worksheet

Alphabetic | Categorized

(Name)	Лист1
DisplayPageBreaks	False
DisplayRightToLeft	False
EnableAutoFilter	False
EnableCalculation	True
EnableFormatConditionsCalculation	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestrictions
Name	Лист1
ScrollArea	
StandardWidth	8,43
Visible	-1 - xlSheetVisible

Watches

Expression	Value	Type	Context
------------	-------	------	---------



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Save Книга1 Ctrl+S

Import File... Ctrl+M

Export File... Ctrl+E

Remove ЭтаКнига...

Print... Ctrl+P

Close and Return to Microsoft Excel Alt+Q

Ln 1, Col 1

(General) (Declarations)

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

Watches

Expression	Value	Type	Context
------------	-------	------	---------



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Can't Undo Ctrl+Z

Can't Redo

Cut Ctrl+X

Copy Ctrl+C

Paste Ctrl+V

Clear Del

Select All Ctrl+A

Find... Ctrl+F

Find Next F3

Replace... Ctrl+H

Indent Tab

Outdent Shift+Tab

List Properties/Methods Ctrl+J

List Constants Ctrl+Shift+J

Quick Info Ctrl+I

Parameter Info Ctrl+Shift+I

Complete Word Ctrl+Space

Bookmarks

Project - VB

Properties -

ЭтаКнига

Alphabetic

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

(General) (Declarations)

Watches

Expression	Value	Type	Context
------------	-------	------	---------

Ln 1, Col 1



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Code F7

Object Shift+F7

Definition Shift+F2

Last Position Ctrl+Shift+F2

Object Browser F2

Immediate Window Ctrl+G

Locals Window

Watch Window

Call Stack... Ctrl+L

Project Explorer Ctrl+R

Properties Window F4

Toolbox

Tab Order

Toolbars

Microsoft Excel Alt+F11

Project - VBAProject

atpvbaen...

Solver (SO...

VBAProject

VBAProject

Microsoft...

Лист

ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

(General) (Declarations)

Watches

Expression	Value	Type	Context
------------	-------	------	---------

Ln 1, Col 1

www.andriystav.cc.ua



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Ln 1, Col 1

Project - VBAProject

- atpvbaen.xls (AT)
- Solver (SOLVER.XL)
- VBAProject (FUNC)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

(General) (Declarations)

Watches

Expression	Value	Type	Context
------------	-------	------	---------



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Ln 1, Col 1

Project - VBAProject

- atpvbaen.xls (ATPVBAEN)
- Solver (SOLVER.XLAM)
- VBAProject (FUNCRES.XLA)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic | Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

(General) (Declarations)

Watches

Expression	Value	Type	Context
------------	-------	------	---------

www.andriystav.cc.ua



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Project - VBAProject

- atpvbaen.xls (ATPVBAEN.XLAM)
- Solver (SOLVER.XLAM)
- VBAProject (FUNCRES.XLAM)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic | Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

Debug Run Tools Add-Ins Window Help

Compile VBAProject

- Step Into F8
- Step Over Shift+F8
- Step Out Ctrl+Shift+F8
- Run To Cursor Ctrl+F8
- Add Watch...
- Edit Watch... Ctrl+W
- Quick Watch... Shift+F9
- Toggle Breakpoint F9
- Clear All Breakpoints Ctrl+Shift+F9
- Set Next Statement Ctrl+F9
- Show Next Statement

(Declarations)

Watches

Expression	Value	Type	Context
------------	-------	------	---------

www.andriystav.cc.ua



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Run Sub/UserForm F5
Break Ctrl+Break
Reset
Design Mode

Project - VBAProject

- atpvbaen.xls (ATPVBAEN.XLAM)
- Solver (SOLVER.XLAM)
- VBAProject (FUNCRES.XLAM)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

Watches

Expression	Value	Type	Context
------------	-------	------	---------

Ln 1, Col 1

(Declarations)

www.andriystav.cc.ua



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

References...

- Additional Controls...
- Macros...
- Options...
- VBAProject Properties...
- Digital Signature...

Project - VBAProject

- atpvbaen.xls (ATPVBAEN.XLAM)
- Solver (SOLVER.XLAM)
- VBAProject (FUNCRES.XLAM)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

Declarations

Watches

Expression	Value	Type	Context
------------	-------	------	---------



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Add-In Manager...

Project - VBAProject

- atpvbaen.xls (ATPVBAEN.XLAM)
- Solver (SOLVER.XLAM)
- VBAProject (FUNCRES.XLAM)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

(General) (Declarations)

Watches

Expression	Value	Type	Context
------------	-------	------	---------



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Project - VBAProject

- atpvbaen.xls (ATPVBAEN.XLAM)
- Solver (SOLVER.XLAM)
- VBAProject (FUNCRES.XLAM)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic | Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

Split

- Tile Horizontally
- Tile Vertically
- Cascade
- Arrange Icons
- ☒ 1 ЭтаКнига (Code)

(Declarations)

Watches

Expression	Value	Type	Context
------------	-------	------	---------

www.andriystav.cc.ua



Microsoft Visual Basic for Applications - Книга1 - [ЭтаКнига (Code)]

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

Project - VBAProject

- atpvbaen.xls (ATPVBAEN.XLAM)
- Solver (SOLVER.XLAM)
- VBAProject (FUNCRES.XLAM)
- VBAProject (Книга1)
 - Microsoft Excel Objects
 - Лист1 (Лист1)
 - ЭтаКнига

Properties - ЭтаКнига

ЭтаКнига Workbook

Alphabetic | Categorized

(Name)	ЭтаКнига
AccuracyVersion	0
AutoUpdateFrequency	0
ChangeHistoryDuration	0
ChartDataPointTrack	True
CheckCompatibility	False
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShapes
DisplayInkComments	True
DoNotPromptForConvert	False
EnableAutoRecover	True
EncryptionProvider	
EnvelopeVisible	False
Final	False
ForceFullCalculation	False
HighlightChangesOnScreen	False
InactiveListBorderVisible	True
IsAddin	False
KeepChangeHistory	True
ListChangesOnNewSheet	False
Password	*****
PersonalViewListSettings	True
PersonalViewPrintSettings	True
PrecisionAsDisplayed	False
ReadOnlyRecommended	False
RemovePersonalInformation	False
Saved	True
SaveLinkValues	True
ShowConflictHistory	False

Help

- Microsoft Visual Basic for Applications Help F1
- MSDN on the Web
- About Microsoft Visual Basic for Applications...

(General) (Declarations)

Watches

Expression	Value	Type	Context
------------	-------	------	---------

www.andriystav.cc.ua



Запуск програм

- Для запуску написаної програми чи процедури достатньо вибрати команду „**Запуск–Запуск процедури/форми/**” або натиснути клавішу F5.



Відладка програм – 1

- Іноді доводиться редагувати написану програму. В таких випадках корисним стає можливість покрокового обходу всіх команд „**Debug–Step into**” (клавіша F8).
- При цьому можна виконувати процедури і функції як одну команду за допомогою оператора „**Debug–Step over**” (Shift+F8).
- Команда „**Debug–Step out**” (Ctrl+Shift+F8) дозволяє виконати частину процедури, що залишалася, і повернутися до точки виклику цієї процедури.
- Команда „**Debug–Run To Cursor**” (Ctrl +F8) виконує набір команд від поточної точки до рядка, де знаходиться курсор.



Відладка програм – 2

- Для установки та знищення точки безумовної зупинки програми використовується команда „**Debug–Toggle Breakpoint**” (клавіша F9).



Перегляд значень змінних

- Наявність перерахованих засобів покрокового виконання команд дозволяє спостерігати за зміною об'єктів під час роботи програми. Для створення вікна зі значеннями змінних використовується команда „**Debug–Add Watch...**”



Зупинка програми

- Зупинити виконання програми можна в будь-який момент комбінацією клавіш **Ctrl-Break**. При цьому редактор запропонує перейти до редагування того рядка, на якому зупинилося виконання програми.



Знищення змінних

- Для знищення всіх створених програмою внутрішніх змінних, виходу з усіх циклів та підпрограм використовується команда **„Run-Reset”**.



Приклад

- [Редактор VBA](#)



2. Змінні VB



Змінні VB

- Основою будь-якої мови є змінні. Для вдалого написання програм їх необхідно вміти створювати та обробляти.



Boolean

- Логічне значення
- розмір 2 байти
- приймає значення True (-1) або False (0)



Byte

- Однобайтне ціле число
- розмір 1 байт
- приймає значення від 0 до 255



Currency

- Грошовий вираз
- розмір 8 байт
- приймає значення від
-922,337,203,685,477.5808 до 922,337,203,685,477.5807



Date

- Дата/час
- розмір 8 байт
- приймає значення від Січень, 1, 100 до Грудень, 31, 9999



Decimal

- Десяткове число
- розмір 14 байт
- приймає значення
 $\pm 79,228,162,514,264,337,593,543,950,335$ без десяткової коми



Double

- Число з плаваючою комою подвійної точності
- розмір 8 байт
- приймає значення від $-1.79769313486231E308$ до $-4.94065645841247E-324$ для від'ємних значень;
 $4.94065645841247E-324$ до $1.79769313486232E308$ для додатних значень



Integer

- Ціле число
- розмір 2 байти
- приймає значення від -32,768 до 32,767



Long / LongLong

- Довге ціле число
- Розмір 4 / 8 байт
- приймає значення
 - від 2,147,483,648 до to 2,147,483,647 /
 - від -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807



Object

- Об'єкт
- розмір 4 байт
- посилання на об'єкт



Single

- Число з плаваючою комою
- розмір 4 байт
- приймає значення
від $-3.402823E38$ до $-1.401298E-45$ для від'ємних значень;
від $1.401298E-45$ до $3.402823E38$ для додатних значень



String

(довільна довжина)

- Текстова змінна
- розмір 10 байт + довжина рядка
- містить від 0 до приблизно 2 млрд. символів



String

(задана довжина)

- Текстова змінна
- розмір - довжина рядка
- приймає значення від 1 до приблизно 65,400 символів



Variant (текстове)

- Довільний
- розмір - 22 байт (24 байти для 64-бітних систем)+ довжина рядка
- Містить будь-яке значення, величина якого не перевищує String



Variant (числове)

- Довільний
- розмір 16 байт
- містить будь-яке значення, точність якого не перевищує Double



Визначена користувачем структура

- Заданий користувачем
- розмір - сума розмірів вкладених елементів
- значення визначається користувачем



Створення змінних

- Для створення нової змінної слід використовувється оператор Dim:
- Dim name [As type]



Ім'я змінної

- починається на літеру;
- максимальна довжина – 255 символів;
- ім'я містить літери, цифри, символ підкреслення;
- такого імені не існує в системі (не є зарезервованим).



Приклад – 1

- Нехай потрібно створити змінну A, яка буде мати містити цілі значення, та задамо їй значення 7:

```
Dim A As Integer
```

```
A=7
```



Приклад – 2

- Змінні типу Boolean мають містити одне зі значень True або False. Чисельно False відповідає 0, а True – будь-яким ненульовим значенням:

```
Dim B As Boolean
```

```
B=True                    'результат: True
```



Числові змінні – 1

- Змінні типу Byte, Integer, Long містять лише цілі числові значення з різних діапазонів.
- Якщо в результаті обчислень такій змінній присвоюється дробове число, то воно спочатку округлюється за звичайними правилами математики.



Приклад

```
Dim A As Byte
```

```
Dim B As Integer
```

```
Dim D As Long
```

```
A = 1.2          'результат: 1
```

```
B = -9.7        'результат: -10
```

```
D = 2.9         'результат: 3
```



Числові змінні – 2

- Дані типів Single та Double містять числа з плаваючою комою з різних діапазонів значень. Дані типу Currency також служать для представлення чисел з плаваючою комою, проте кількість розрядів у них обмежена чотирма.



Приклад

```
Dim k1 As Single
```

```
Dim k2 As Double
```

```
Dim m As Currency
```

```
k1 = 2.9          'результат: 2.9
```

```
k2 = -13.3       'результат: -13.3
```

```
m = 123.459      'результат: 123.4590
```



Змінні типу Date

- Дані типу Date спеціально призначені для обробки інформації про час та дату. Для коректного вводу даних необхідно використовувати знак „#” або вводити дату за встановленим у поточній системі Windows форматі.



Приклад

```
Dim t1 As Date, t2 As Date, t3 As Date, t4 As Date
```

```
t1 = #10/29/2004# 'результат: 29.10.2004
```

```
t2 = "29.10.2004" 'результат: 29.10.2004
```

```
t3 = #4:45:00 PM# 'результат: 16:45
```

```
t4 = "16:45" 'результат: 16:45
```



Дані типу String

- Дані типу String слугують для зберігання текстових рядків. Перші два символи динамічного рядка зберігають його довжину. Для вказування явної довжини рядка, необхідно вказати максимально можливу довжину при об'яві цієї змінної.



Приклад

```
Dim S1 As String
```

```
Dim S2 As String * 30
```

```
S1 = "Привіт!"
```

```
S2 = "До побачення!"
```



Дані типу Decimal

- Дані типу Decimal можуть використовуватися лише всередині змінної типу Variant, тобто не можна оголосити змінну такого типу в операторі Dim.



Зауваження

- VB не потребує явного визначення змінних, якщо тип не вказано, то вибирається Variant, який може містити змінні будь-якого типу: їй можна присвоювати змінні будь-якого типу, включаючи об'єктного.



Приклад

```
Dim V As Variant
```

```
V = "25" 'результат: текст „25”
```

```
V = V + 4 'результат: число 29
```

```
V = V & " студентів" 'результат: текст „29 студентів”
```



Недолік типу Variant

- Слід зауважити, що хоча зміна типу Variant є зручною, часте її використання є недоречним, оскільки такі змінні використовують багато системних ресурсів, а також його не завжди можна передати як параметр до процедури.



Явне створення змінних

- Якщо на початку кожного модуля можна вказати опцію

Option Explicit

то VB вимагатиме явного визначення типів змінних.



Перетворення змінних

Функції

- CBool,
- CByte,
- CCur,
- CDate,
- CDbt,
- CDec,
- CInt,
- CLng,
- CLngLng,
- CLngPtr,
- CSng,
- CStr,
- CVar.



Доступність змінних

- локальні (local) – змінна доступна тільки в поточній процедурі;
- контейнерні (private) – змінна доступна лише в поточній формі, модулі чи класі;
- глобальні (public) – змінна доступна у всьому проекті.



Константи

- Аналогічно до змінних оголошуються і константи. Основна відмінність констант від змінних полягає в тому, що їх значення не можна міняти в процесі роботи програми.
- Також розрізняються локальні, глобальні та контейнерні константи.

```
[Public | Private] Const name = value
```

- Глобальні константи можна декларувати лише в модулі.



Приклад

Const $M_y = 4.56$



Масиви

- Масив – це набір елементів визначеного типу, кожен з яких має свій номер (індекс).
- Розрізняють статичні та динамічні масиви.
- Перші не змінюють свого розміру, другі – змінюють його під час роботи програми.



Статичні масиви

- При статичному оголошенні масиву слід точно вказати максимальну кількість елементів, що може знаходитися в ньому:

```
Dim name(Size) As type
```



Приклад

- Нехай потрібно створити масив, що містить 100 прізвищ студентів:

```
Dim Students (100) As String
```



Адресація масивів

- Для звернення до елемента масиву необхідні індекси вказуються у круглих дужках:

```
Students(4)="Петренко"
```



Границі масивів

- Нижня межа для масиву, як правило, 0. Якщо потрібно провести індексування з 1 необхідно вказати опцію:

Option Base 1

- Для встановлення специфічних меж масиву необхідно записати
[Static | Public] Dim name ([Low To] High)



Приклад

- Створимо масив під назвою my, що містить 51 ціле число (з 1950 по 2000).

```
Dim my (1950 to 2000) As Integer
```



Багатовимірні масиви

- Для створення багатовимірних масивів слід вказати всі розміри. Наприклад, для створення двовимірного масиву, що містить текстові елементи можна скористатися таким записом:

```
Dim name (10, 25) As String
```



Зауваження

- Кількість вимірів масиву не може перевищувати 60.



Приклад

- Записи в багатовимірному масиві можуть комбінуватися:

```
Dim A (10, 80 to 120, 40 to 45, 1 to 255, 17) As Byte
```

- В цьому прикладі створюється масив A, який містить $10 * 41 * 6 * 255 * 17 = 10\ 664\ 100$ елементів типу Byte.



Динамічні масиви

- Спочатку оголошується назва масиву без оголошення його розміру:

```
Dim A() As String
```

- Коли потрібно, встановлюється його новий реальний розмір:

```
ReDim [Preserve] name (границі) [As type]
```



Оператор ReDim

- Оператор ReDim використовується лише в процедурах. Тип даних вказувати необов'язково, адже змінити тип даних, вказаних в операторі Dim неможливо (за виключенням типу Variant).
- Для багатовимірних масивів оператор ReDim дозволяє змінювати розмір лише останнього виміру масиву.



Опція Preserve

- Для того, щоб не втратити інформацію масиву, яка вже зберігається в ньому, вказують слово Preserve.



Присвоювання масивів

- Однакові за розмірами та типом даних масиви можна присвоювати один одному:

```
Dim oldArr() As Byte, newArr() As Byte  
newArr=oldArr
```



Типи даних, що створюються користувачем

- Для визначення такого типу потрібно вказати типи даних, які комбінуються:

```
[Private | Public] Type ім'я_типу
```

```
    елемент1 [[[Розмірність]]] As type
```

```
    елемент2 [[[Розмірність]]] As type
```

```
    ...
```

```
    елемент_n [[[Розмірність]]] As type
```

```
End Type
```



Оголошення типу – 1

- Визначення загального (public) типу даних можливе лише у секції (General – Declarations) модуля. В цьому випадку цей тип буде доступний у всіх процедурах всіх форм, модулів та класів.



Оголошення типу – 2

- Для визначення типу даних у формі чи модулі класу слід використовувати ключове слово `Private`, оскільки оголошення загального типу в даній ситуації не допускається.
- При цьому область видимості такого типу буде обмежена тим контейнером, де він оголошений.



Доступ

- Доступ до елементів змінної створеного користувачем типу здійснюється за допомогою знаку „.”. Змінні однакового типу можна також присвоювати не поелементно, а напряму.



Приклад

```
'Секція General –  
Declarations модуля  
Type Student  
  Fio As String * 40  
  Birth As Date  
  Speciality As Integer  
  Course As Byte  
  Info As String  
End Type
```

```
'Фрагмент програми  
Sub TTT()  
  'Оголошення змінної  
  Dim S As Student  
  
  Dim S1 As Student  
  'Заповнення елементів  
  запису  
  S.Fio = "Смолянов"  
  
  S.Birth = "10.04.1985"
```

```
S.Speciality = 4  
S.Course = 2  
S.Info = "Призер  
олімпіади"  
'Пряме присвоювання  
S1 = S  
  
End Sub
```



3. Процедури та функції



Процедури

- Під процедурами розуміється послідовність об'яв змінних та інструкцій, об'єднаних для виконання. Вони поділяються на процедури обробки подій та процедури загального призначення.



Процедури обробки подій – 1

- Процедури обробки подій використовуються для реагування на дії користувача чи системи. Вони завжди належать деякому об'єкту і можуть викликатися лише з цього об'єкту.



Процедури обробки подій – 2

- Процедуру подібного типу легко відрізнити по її імені, в якому присутня назва об'єкта-володаря та типу події. Наприклад, реагування програми на натискання користувачем мишкою на кнопку управління здійснюється такою процедурою:

```
Private Sub Command1_Click()
```

```
End sub
```



Процедури загального призначення

- Процедури загального призначення можуть викликатися або користувачем або іншими процедурами.

```
Sub MyProc()
```

```
End sub
```



Передача параметрів

- Параметри до процедур можуть передаватися двома шляхами:
 - по посиланню;
 - по значенню.
- В першому випадку значення змінних можна буде змінювати, в другому – ні.

```
Sub My([ByRef | ByVal] A)
```




Зауваження

- Якщо не вказати жодного способу, то VB вибирає передачу по посилянню.



Необов'язкові параметри – 1

- У випадку, коли один або декілька параметрів є необов'язковими, перед ними становиться опція Optional:

```
Sub AAA(B As Byte, Optional A As String)
```

```
End sub
```



Необов'язкові параметри – 2

- Пропущені змінні ініціалізуються початковими значеннями відповідного типу даних, тобто числові змінні дорівнюють 0, текстові – порожньому рядку тощо.



Функція IsMissing

- Функція

`IsMissing(назва змінної)`

визначає, чи був переданий до процедури аргумент типу Variant.



Приклад

```
Sub Main()
```

```
    Call Test(1)
```

```
    Call Test(1, 3)
```

```
End Sub
```

```
Sub Test(A As Byte, Optional B)
```

```
    If IsMissing(B) Then B = 12
```

```
    MsgBox A + B
```

```
End Sub
```



Функції – 1

- Функції VB записуються з ключовим словом Function замість Sub. Крім цього вони обов'язково мають повертати результат:

```
Function My(A As Byte, B As byte) As Byte
```

```
My=A+B
```

```
End Function
```



Функції – 2

- Повертати можна навіть масиви.



Приклад

```
Function GetAr(A As Byte) As Byte()
```

```
Dim m(2) As Byte
```

```
m(0) = A
```

```
m(1) = A + 12
```

```
m(2) = A + 3
```

```
GetAr = m
```

```
End Function
```




Функція InputBox

```
Result = InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile,  
context])
```



Функція MsgBox

```
Result=MsgBox(prompt[, buttons] [, title] [, helpfile, context])
```



Приклад

- [Комунікація з користувачем](#)



4. Оператори та цикли



Умовний оператор – 1

- Найбільш часто використовується умовний оператор, який дозволяє перевірити виконання однієї або декількох умов. Він може бути записаний у короткому (однорядковому) вигляді або у блочному (багаторядковому).



Умовний оператор – 2

- Основною відмінністю між ними є наявність кінцевого службового слова

End If.

- До того ж, в першому випадку після слів Then та Else може стояти лише один оператор для виконання, що не завжди є зручним.



Приклад – 1

- Однорядковий синтаксис

If умова Then оператор [Else оператор]



Приклад – 2

- Блочний синтаксис

```
If умова1 Then
```

```
    [оператори1]
```

```
[Elself умова2 Then
```

```
    оператори2]
```

```
[Else оператори3]
```

```
End if
```




Приклад – 3

- Виводимо на екран оцінку студенту в залежності від результату його тестування.



Приклад – 3

```
Sub Q1(Result As Integer)
```

```
Dim S As String
```

```
If Result > 90 Then
```

```
S = "Відмінно"
```

```
Elseif Result > 75 Then
```

```
S = "Добре"
```

```
Elseif Result > 66 Then
```

```
S = "Задовільно"
```

```
Else
```

```
S = "Незадовільно"
```

```
End If
```

```
MsgBox S
```

```
End Sub
```



Оператор вибору

- Якщо потрібно вибрати дію в залежності від багатьох умов, то використання великої кількості умовних операторів є недоцільним: достатньо використати оператор вибору.



Синтаксис оператора вибору

```
Select Case тестовий_вираз  
[Case значення1  
    [Оператори1]]  
[Case значення2  
    [Оператори2]]  
[Case значення_n  
    [Оператори_n]]  
[Case Else  
    [Оператори_n+1]]  
End Select
```



Зауваження

- В якості блоку вибору можна вказувати не тільки одне значення, а й:
 - декілька (розділені комою),
 - діапазон (3 to 6),
 - команду вибору (ls >=9).
- Якщо при цьому умові відповідає декілька блоків, то виконується перший з них. Якщо жодний блок не задовольняє умові, то виконується блок, що йде зразу після слів Case Else.



Приклад

```
Sub Q2(A As Single)
```

```
Dim S As String
```

```
Select Case A
```

```
Case 1
```

```
    S = "дорівнює 1"
```

```
Case 2, 3
```

```
    S = "дорівнює 2 або 3«
```

```
Case 4 To 6
```

```
    S = "знаходиться в межах від 4
```

```
до 6"
```

```
Case Is >= 9
```

```
    S = "не менше 9"
```

```
Case Else
```

```
    S = "невідоме"
```

```
End Select
```

```
MsgBox S
```

```
End Sub
```



Цикли проходу

- Для багатократного виконання одного чи декількох операторів призначені цикли. Цикл безумовного проходу For ... Next дає можливість встановлювати число проходів циклу, а цикл Do...Loop завершається при досягненні заданої умови.



Синтаксис циклу безумовного проходу

```
For name=value1 To value 2 [Step value3]
```

```
[Оператори]
```

```
next [name]
```




Робота циклу – 1

- На початку виконання циклу значення змінної `name` встановлюється у початкове значення `value1`. Після кожного проходу (досягненні оператора `next`) змінна `name` збільшується на значення кроку – `value3`. Якщо після цього значення `name` дорівнює `value2`, то цикл завершується, і виконується наступна команда після слова `next`.



Робота циклу – 2

- Слід пам'ятати, що для коректної роботи циклу необхідно, щоб при кожному новому проході значення `name` наближалось до `value2`, в протилежному випадку програма буде виконуватися вічно.
- Для безумовного виходу з циклу використовується оператор `Exit`.



Приклад

```
Sub My()  
Dim A(1997 to 2005)  
For i = LBound(A) To UBound(A)  
    If A(i) <> 0 Then  
        MsgBox Str(A(i))  
    Else: Exit For  
    End If  
Next i  
End Sub
```



Приклад

- Функції $LBound(A)$ та $UBound(A)$ визначають нижню та верхню границі масиву A . Процедура Mu виводить на екран елементи масиву, доки не буде знайдено нульовий елемент або не закінчиться перебір масиву. Якщо нульовий елемент знайдено, то цикл негайно завершується.



УМОВНІ ЦИКЛИ

- Якщо кількість проходів має залежати від умови, то використовують цикл Do...Loop. В залежності від розташування службових слів розрізняють два варіанти циклу.



Перевірка умови на початку циклу

Do [{While | Until} умова]

[Оператори]

[Exit Do]

[Оператори]

Loop



Перевірка умови в кінці циклу

Do

[Оператори]

[Exit Do]

[Оператори]

Loop [{While | Until} умова]



Зауваження

- На відміну від першого запису, другий цикл виконається принаймні один раз. Якщо вибрано службове слово `While`, то цикл буде виконуватися, поки умова залишатиметься вірною. Якщо використати службове слово `Until`, то цикл буде виконуватися, поки умова буде не стане вірною.



Приклад

```
Dim i As Integer
```

```
i = 1
```

```
Do
```

```
i = i + 1
```

```
Loop While i < 9
```

```
MsgBox i 'На екран виводиться  
        значення 9
```



Оператор перехвату помилок – 1

- Для реагування на помилки, що можуть виникнути в програмі, призначений оператор On Error:

On Error [Goto Label | Resume Next | Goto 0]



Оператор перехвату помилок – 2

- Виконання оператора перехвату помилок активізує обробку помилок всередині програми.



Перехід на мітку

- При виникненні помилки управління передається на рядок, що містить мітку MyLabel.

On Error Goto MyLabel

- Як правило, така мітка стоїть в кінці процедури або функції.



Продовження роботи

- Оператор

On Error Resume Next

- продовжити роботи програми з наступного рядка за тим, де виникла помилка.



Вимикання режиму обробки помилок

- Якщо ж вказати

`On Error Goto 0`

- то вимикається обробка помилок всередині програми.



Обробка помилки

- Обробка помилки починається зразу ж при її виникненні. У VB існує об'єкт Err, властивість якого Number містить код останньої помилки.
- Завершення блоку обробки помилки завершується оператором Resume:

```
Resume [{Next | Label}]
```



Приклад

- Ділення на 0!



Приклад

```
Sub MyError()  
Dim A As Integer  
On Error GoTo MyHandler  
For A = -1 To 2  
    B = 7 / A  
Next A  
Exit Sub  
MyHandler:  
Select Case Err.Number  
Case 11
```

```
    A = 1  
    Err.Clear  
    Resume  
Case Else  
    MsgBox Err.Description  
    Stop  
End Select  
End Sub
```



5. Робота з файлами



Процес роботи з файлом у VB

- Отримання дескриптора файлу (handle).
- Відкриття файлу.
- Читання або запис даних.
- Закриття файлу.



Отримання дескриптору

- Для отримання вільного дескриптору використовується функція `FreeFile([RangeNumber])`:

```
intFH = FreeFile()
```



Відкриття файлу

- Для послідовного доступу до файлу відкриття здійснюється наступним чином:

Open name For [Input | Output | Append] As filehandle



Читання та запис

- Читання та запис здійснюються командами Line Input та Print відповідно:

```
Line Input intFH, strString
```

```
Print intFG, strText1, strText2, ...
```



Закриття файлу

- Закриття файлу здійснюється командою Close

Close intFH



Приклад – 1

- Створимо файл „file.txt”, запишемо в нього фразу „Привіт вінчестеру!” та закриємо файл.



Приклад – 1

```
Sub OutputFile()
```

```
intFH = FreeFile()
```

```
Open "file.txt" For Output As #intFH
```

```
Print #intFH, "Привіт вінчестеру!"
```

```
Close #intFH
```

```
End Sub
```



Зауваження

- Якщо файл вже існував, то вся попередня інформація буде знищена, тому потрібно бути обережним при відкритті файлу на запис.



Приклад – 2

- Прочитати файл і вивести на екран його зміст.



Приклад – 2

```
Sub ReadFile()  
Dim S As String  
intFH = FreeFile()  
Open "file.txt" For Input As #intFH  
Do While Not EOF(intFH)  
    Line Input #intFH, S  
    MsgBox S  
Loop  
Close #intFH  
End Sub
```



Функція EOF

- В прикладі використана функція EOF(filehandle), яка повертає значення True, якщо досягнуто кінець файлу, в протилежному випадку повертається False.



Приклад – 3

- Додамо до файлу ще один рядок:

```
Sub AppendFile()
```

```
intFH = FreeFile()
```

```
Open "file.txt" For Append As #intFH
```

```
Print #intFH, "Привіт процесору!"
```

```
Close #intFH
```

```
End Sub
```



Поелементний доступ

- Часто існує потреба розглядати файл як масив елементів однакової довжини – 1 байт. В такому випадку говорять про бінарний доступ до файлу. В такому випадку слід відкривати файл з ключовим словом Binary:

Open name For Binary [Доступ] [Блокування] As filehandle



Опція Доступ

- Не вказана
- Читання (Read)
- Запис (Write)
- Читання та запис (Read Write)



Опція Блокування

- Shared - Файл може використовуватися всіма процесами для читання та запису
- Lock Read - Жоден інший процес не може отримувати дані з файлу
- Lock Write - Жоден інший процес не може записувати дані до файлу
- Lock Read Write - Жоден процес не може ні зчитувати, ні записувати дані



Приклад – 4

- Відкрити файл "file.txt" як бінарний і прочитати всі його символи.



Приклад – 4

```
Sub CharReadFile()
```

```
Dim S As Byte
```

```
Dim i As Integer
```

```
i = 1
```

```
intFH = FreeFile()
```

```
Open "file.txt" For Binary As #intFH
```

- Do While i < LOF(#intFH)
- Get #intFH, i, S

- MsgBox Chr(S)
- i = i + 1
- Loop
- Close intFH
- End Sub



Функція LOF

- Функція `LOF(filehandle)` повертає довжину файлу у байтах, тому буде виведено стільки символів, скільки байтів містить файл "file.txt".



Функції читання-запису

- Функція

`Get filehandle, Pos, var`

- читає з файлу байт, що розташований на позиції Pos.

- Для запису байту використовується процедура Put:

`Put filehandle, Pos, var`



Приклад – 5 – А

- Модифікувати програму таким чином, щоб замість знаку оклику поставити знак питання.



Приклад –

```
Sub ReplaceChar()
```

```
Dim S As Byte
```

```
Dim i As Integer
```

```
i = 1
```

```
intFH = FreeFile()
```

```
Open "file.txt" For Binary As #intFH
```

```
Do While i < LOF(intFH)
```

```
    Get #intFH, i, S
```

```
    If Chr(S) = "!" Then Put #intFH, i,  
    "?"
```

```
    i = i + 1
```

```
Loop
```

```
Close #intFH
```

```
End Sub
```



Файли з довільним доступом

- Файли з довільним доступом дуже схожі на бінарні файли, тільки в якості одного елементу може бути змінна зазначеної довжини. Зручно використовувати такі файли, якщо в програмі використовуються записи, створені користувачем.



Відкриття файлу

- Відкривається такий файл командою Open з ключовим словом Random:

Open name For Random [Доступ] [Блокування] As filehandle
[Len=Довжина_запису]



Функції читання-запису

- Процедури Put та Get замість номера байту в файлі використовують номер запису довжини, який визначається параметром Len.



Приклад – 6 – А

- Записати до файлу інформацію про одного студента.



Приклад – 6 – Б

```
Sub RecordsSave()
```

```
Dim Stud As Student
```

```
Stud.Person = "Іваненко Іван"
```

```
Stud.Group = 3
```

```
Stud.Course = 2
```

```
IntFH = FreeFile()
```

```
Open "students.dat" For Random  
Access Read Write As #IntFH Len =  
Len(Stud)
```

```
Put #IntFH, 2, Stud
```

```
Close #IntFH
```

```
End Sub
```



5. Форма та її елементи у VB



Форми

- Форми відіграють основну роль у спілкуванні користувача з програмою.
- Нова форма створюється так само як і новий модуль: Insert-UserForm.



Основні властивості форми

- позиція на екрані
(координати Left, Top, Height, Width)
- поточна видимість (Visible)
- ім'я (name)



Найважливіші події форми - 1

- Click - Натискання кнопки мишки
- DbClick - Подвійне натискання кнопки мишки
- MouseDown - Натискання і утримування кнопки мишки
- MouseUp - Відпускання кнопки мишки
- MouseMove - Пересування мишки до нової позиції



Найважливіші події форми - 2

- KeyPress - Натискання клавіші клавіатури
- KeyDown - Натискання і утримування клавіші клавіатури
- KeyUp - Відпускання клавіші клавіатури



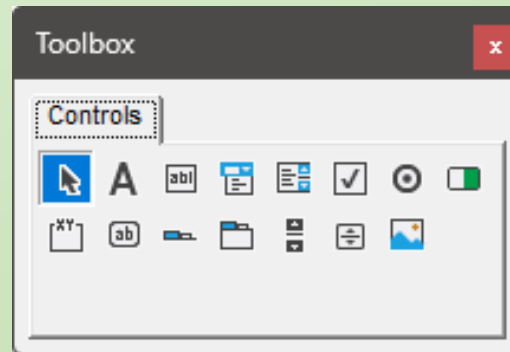
Створення процедури-реакції на подію

- Для введення коду реагування на подію необхідно перейти в модуль процедур форми (клавіша F7 або подвійний клік миші на формі)



Елементи форми

- Звичайно, для виводу інформації однієї форм замало, тому необхідно вміти додавати до форми нові елементи. Для вставки нових об'єктів зручно включити вікно View-Toolbox:





Вставка об'єкта

- Для вставки об'єкта необхідно вказати мишкою його розташування та розмір. При цьому можна змінити деякі властивості об'єкта.
- За допомогою редактора створюються процедури обробки подій.



CheckBox



- Click
- Value (0-не відмічено, 1- відмічено, 2 – відмічено, але недоступно)



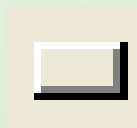
ComboBox



- Click,
- ChangeListBox+TextBox



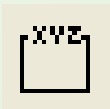
CommandButton



- Click
- Caption, Default, Cancel



Frame



- Click
- Caption



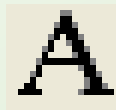
Image



- Click
- Picture, Visible



Label



- -
- Caption, AutoSize, WordWrap, BorderStyle



ListBox



- Click,
- AddItem(Value, Index), RemoveItem(Index), Clear
- List(Index), ListCount



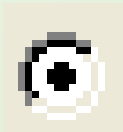
MultiPage



- Click
- Enabled, Value



OptionButton



- Click
- Value (true або false)



ScrollBar



- Change
- Value, Min, Max



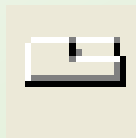
SpinButton



- Change, SpinDown, SpinUp
- Min, Max, Value



TabStrip



- Click
- Enabled, Value



TextBox

abl

- Change
- Text



ToggleButton



- Click
- Caption, Value



Інші важливі об'єкти

- Колекція Dialogs – будь-які вбудовані діалогові вікна;
- Колекція CommandBars – зміна меню програми.



Формування звіту 5.xlsm - Module1 (Code)

(General) Choose

```
Public V As Integer
Public wd As Object
Public f As String
Public N As Integer

Sub Choose()
On Error GoTo bd

Application.DisplayAlerts = False
f = Application.GetOpenFilename("файл xlsm, *.xlsm,Всі файли, *.*")
na = ActiveWorkbook.Name
Application.Workbooks.Open (f)
nb = ActiveWorkbook.Name
Windows(nb).Activate
Sheets(1).Select
Cells.Select
Selection.Copy
Windows(na).Activate
Sheets("Data").Select
Cells.Select
ActiveSheet.Paste
Windows(nb).Close
Application.DisplayAlerts = True

UserForm2.Show

Exit Sub
bd:
MsgBox ("Перевірте правильність вибору файлу")

End Sub
```

www.andriystav.cc.ua



7. Основні об'єкти програм MS Office



Колекція Tasks

- При роботі можуть бути відкритими багато програм, тому весь їх перелік знаходиться у колекції Tasks.
- За її допомогою можна перевірити, чи відкрита необхідна програма та отримати до неї доступ.



Об'єкт Application

- Кожна програма є об'єктом Application, що містить всю інформацію про відповідну програму.



Приклад

- Перевірити чи відкрита програма MS Word. В разі знаходження програми MS Word передати їй управління, розмір вікна програми збільшити до максимального.



Приклад

```
If Tasks.Exists("Microsoft Word") = True Then  
    With Tasks("Microsoft Word")  
        .Activate  
        .WindowState = wdWindowStateMaximize  
    End With  
Else  
    MsgBox "Microsoft Word не запущено."  
End If
```



Колекція Documents

- Всі відкриті документи програми MS Word містяться у колекції Documents, що містить об'єкти типу Document.
- Для визначення кількості елементів довільної колекції використовується змінна Count.



Приклад

- В прикладі відкривається файл "E:\DB\MyDoc.doc" тільки для читання, виводиться назва кожного відкритого файлу, закривається відкритий файл.

Приклад *TestsWord2*



Робота з колекцією

- Для переключення між документами достатньо вказати або номер, або ім'я необхідного документа:

```
Documents(1).Activate
```

```
Documents("MyDoc.doc").Activate
```



Створення нового документа

- Створення нового документа здійснюється командою

`Documents.Add`



Збереження файлу

- Збереження файлу на диску здійснюється командою

```
Documents("MyDoc.doc ").Save
```

- Якщо потрібно зберегти файл під іншим ім'ям, то запис здійснюється командою

```
ActiveDocument.SaveAs FileName:="Temp.doc"
```



Об'єкт Document

- Кожен документ має всю інформацію про текст, стилі, опції, що зберігаються у відповідному файлі. Звернення до цієї інформації можливе за допомогою методів і властивостей об'єкта Document.



Колекції об'єкта Document

- символів (Characters),
 - слів (Words),
 - абзаців (Paragraphs),
 - розділів (Sections),
 - коментарів (Comments),
 - закладок (Bookmarks),
 - полів (Fields),
 - об'єктів, включаючи OLE
- (InlineShapes),
 - приміток (Footnotes)
 - таблиць (Tables),
 - списків (Lists),
 - гіперпосилань (Hyperlinks),
 - індексів (Indexes).



Інші об'єкти Word

- Більшість дій проводиться не з усім документом, а з виділеним фрагментом (Selection) або заданим фрагментом (Range).



Приклад: зміна шрифту

- Наступний приклад ілюструє можливості цих двох об'єктів. В якості заданого фрагмента (змінна `rngParagraph`) вибрано два повних перших абзаци (з початку другого до кінця третього). Цей фрагмент був виділений (метод `Select`), і в ньому формат тексту був визначений як курсив.



Приклад

```
Dim rngParagraph As Range
```

```
Set rngParagraph = ActiveDocument.Paragraphs(2).Range
```

```
rngParagraph.SetRange Start:=rngParagraph.Start,  
End:=ActiveDocument.Paragraphs(3). Range.End
```

```
rngParagraph.Select
```

```
Selection.Font.Italic = True
```



Приклад

- [Форматування тексту у VBA](#)



MS Excel

- Програма MS Excel також являє собою об'єкт Application.
- Всі відкриті файли містяться в об'єкті Workbooks, який являє собою колекцію робочих книг (Workbook).



Файли MS Excel

- Кожна робоча книга складається з одного чи декількох листів (колекція листів книги Worksheets містить об'єкти типу Worksheet) та графіків (колекція графіків Charts містить об'єкти типу Chart).



Об'єкт Worksheet

- графіки (колекція ChartObjects)
- об'єкти (колекція OLEObjects)
- коментарі (Comments)
- гіперпосилання (Hyperlinks)
- зведені таблиці (PivotTables)
- таблиці запитів (QueryTables)



Чарунки листа

- Кожен лист складається з чарунок, за масив яких відповідає властивість Cells об'єкту Worksheet.
- Для звернення до чарунки вказується номер її рядка та стовпчика:

```
MsgBox Worksheets("Лист1").Cells(5, 3)
```




Об'єкт Range

- Для звернення до діапазону чарунок слід використовувати об'єкт Range:

```
ActiveSheet.Range("A1:D10") = 10
```



Приклад

```
Sub TestExcel()
```

```
With ActiveSheet.Range("A1:D12")
```

```
.Formula = "=10*RAND()"
```

```
.Font.Italic = True
```

```
.NumberFormat = "0.00"
```

```
End With
```

```
End Sub
```



Колекція AddIns

- Колекція AddIns містить всі додаткові модулі MS Excel.
- Наступний приклад виводить список назв всіх таких модулів та інформацію про їхнє поточне завантаження.



Приклад

```
Sub DisplayAddIns()
```

```
    rw = 1
```

```
    For Each ad In Application.AddIns
```

```
        ActiveSheet.Cells(rw, 1) = ad.Name
```

```
        ActiveSheet.Cells(rw, 2) = ad.Installed
```

```
        rw = rw + 1
```

```
    Next
```

```
End Sub
```



MS PowerPoint

- Об'єкт Application містить програму MS PowerPoint.
- Всі відкриті файли містяться в об'єкті Presentations, який являє собою колекцію презентацій (Presentation).



Презентація

- Кожна презентація містить колекцію слайдів Slides. Кожен слайд (Slide) містить інформацію про розташовані на ньому об'єкти, кольорові схеми, інформацію про правило показу слайду.



Приклад

- Запустити презентацію з іншого середовища.

```
Set ppt = GetObject(,"Powerpoint.Application")
```

```
ppt.Visible = True
```

```
ppt.Presentations.Open FileName:="E:\DB\MyPres.pptx",  
ReadOnly:=msoFalse
```

```
ppt.ActivePresentation.SlideShowSettings.Run
```



MS Outlook

- Об'єкт Application є визначальним для роботи програми MS Outlook. Він містить об'єкти управління контактами, поштою тощо.



Приклад

- Запустити MS Outlook та створити вікно для написання нового листа.

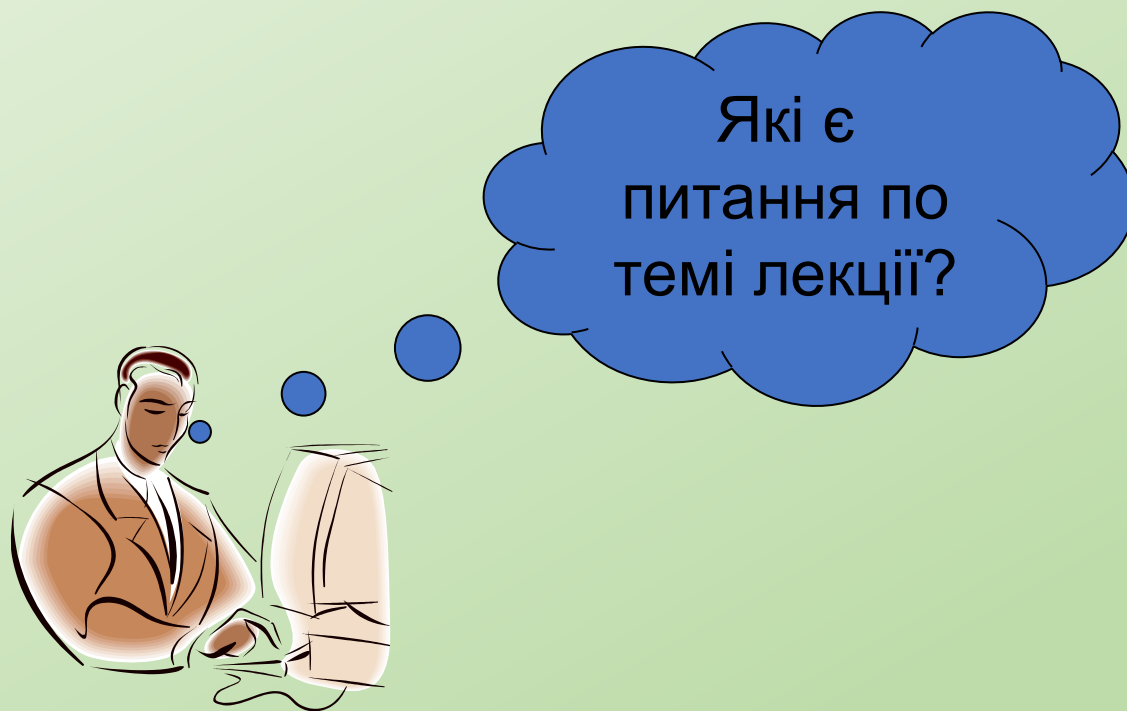
```
Set myOlApp = CreateObject("Outlook.Application")
```

```
Set myItem = myOlApp.CreateItem(olContactItem)
```

```
myItem.Display
```



Закінчення





Дякую за увагу!