

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Київський національний університет будівництва і архітектури

# **ПРОГРАМУВАННЯ МОВОЮ VISUAL BASIC**

Методичні вказівки  
до виконання лабораторних робіт  
для здобувачів першого (бакалаврського)  
рівня вищої освіти спеціальності  
193 «Геодезія та землеустрій»



Київ 2025

УДК 528.

П51

Укладач В. В. Зіборов, канд. техн. наук, доцент

Рецензент Н. Ю. Лазоренко, канд. техн. наук, доцент

Відповідальний за випуск Ю. О. Карпінський, д-р техн. наук, професор

*Затверджено на засіданні кафедри геоінформатики і фотограмметрії, протокол № 2 від 28 серпня 2024 року.*

В авторській редакції.

**Програмування мовою Visual Basic [Електронний ресурс]:**  
П51 методичні вказівки до виконання лабораторних робіт / уклад.  
В. В. Зіборов. – Київ : КНУБА, 2025. – 32 с.

Містять загальні положення, теоретичні відомості, методичні вказівки до виконання лабораторних робіт, список літератури.

Призначені для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 193 «Геодезія та землеустрій»

©КНУБА, 2025

## ЗМІСТ

ЗМІСТ.....	3
ЗАГАЛЬНІ ПОЛОЖЕННЯ .....	4
<b>БЛОК 1. НАЙПРОСТІШЕ ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ .....</b>	<b>4</b>
БЛОК 1. ЗАВДАННЯ 1.....	4
БЛОК 1. ЗАВДАННЯ 2.....	7
<b>БЛОК 2. ЕЛЕМЕНТИ УПРАВЛІННЯ НА ФОРМІ.....</b>	<b>8</b>
БЛОК 2. ЗАВДАННЯ 3.....	8
БЛОК 2. ЗАВДАННЯ 4. ОДИННА ДЛЯ ВСІХ .....	11
БЛОК 2. ЗАВДАННЯ 5. ОДИННА ДЛЯ ВСІХ .....	13
БЛОК 2. ЗАВДАННЯ 6.....	16
<b>БЛОК 3. МАСИВИ І ЦИКЛИ .....</b>	<b>22</b>
БЛОК 3. ЗАВДАННЯ 7. ОДИННА ДЛЯ ВСІХ .....	22
БЛОК 3. ЗАВДАННЯ 8. ОДИННА ДЛЯ ВСІХ .....	22
БЛОК 3. ЗАВДАННЯ 9. ОДИННА ДЛЯ ВСІХ .....	23
БЛОК 3. ЗАВДАННЯ 10.....	23
<b>БЛОК 4. БІЛЬШ СКЛАДНІ ЗАВДАННЯ.....</b>	<b>27</b>
БЛОК 4. ЗАВДАННЯ 11.....	27
БЛОК 4. ЗАВДАННЯ 12.....	29
<b>ПРИМІТКИ .....</b>	<b>29</b>
<b>СПИСОК ЛІТЕРАТУРИ.....</b>	<b>31</b>

## ЗАГАЛЬНІ ПОЛОЖЕННЯ

Метою розробки методичних вказівок до виконання лабораторних робіт є формування у студентів першого курсу практичної бази знань з основ інформатики і програмування, умінь та навичок ефективного використання сучасних комп'ютерно-інформаційних технологій у своїй діяльності, що має забезпечити формування у студентів основ інформаційної культури та інформативно-комунікативної компетентності.

Використовуючи ці методичні вказівки, необхідно в MS Word написати звіт за виконаними завданнями, наведеними нижче. За кожним завданням звіт має містити такі пункти:

1. Постановка задачі.
2. Програмний код.
3. Фрагменти роботи програми.
4. Пояснення до програми.

Також необхідно продемонструвати навичку налагодження програми (F8 і показати поточні значення змінних).

## БЛОК 1. НАЙПРОСТІШЕ ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ

### БЛОК 1. ЗАВДАННЯ 1

Принцип вирішення цього завдання можна переглянути на відео:

<https://youtu.be/AGhxvaV7o-8>

Інші відео автора:

[https://www.youtube.com/channel/UCFZglBgYat1GGPMCPQ4XbJQ?sub\\_confirmation=1](https://www.youtube.com/channel/UCFZglBgYat1GGPMCPQ4XbJQ?sub_confirmation=1)

Кожну нову програму ми пишемо у новій папці. Створимо першу програму (Windows-додаток ) мовою Visual Basic, яка складає два числа і виводить результат на екран зі словом "Сума = ":

1. У будь-якому текстовому редакторі (але не Word), наприклад, Блокнот наберіть наступний текст програми на VB:

- ' **Моя перша програма**
- ' **Коментарії роблять програму більш зрозумілою**

**Sub Main()**

- ' ім'я процедури, Main - основна процедура,
- ' з неї починається виконання програми
- ' Оголошення змінних цілого типу:

```
Dim A As Integer
Dim B As Integer
Dim S As Integer
```

- ' Введення даних:

```
A = InputBox("Введіть перше число")
B = InputBox("Введіть друге число")
```

- ' Розрахунок:

```
S = A + B
```

- ' Виведення результату у вікно:

```
MsgBox ("Сума = " & S)
```

- ' Кінець програми:

```
End Sub
```

2. Цей текст слід набрати в будь-якому редакторі (Блокноті) і записати на диск у нову папку під Вашим ім'ям, наприклад, VA1, Ім'я файлу має розширення \*.bas (Ветров Андрій, перша програма), у папці Вашої групи, наприклад ГКЗ-11.

3. Як видно з тексту VB програми, після апострофу пишуть коментарі. Коментарі ігноруються компілятором. Багато коментарів – це гарний стиль програмування. Стартова точка програми – **Sub Main()**, а кінець програми – **End Sub**.

4. Кожен оператор пишуть в одному рядку. Щоб написати кілька операторів в одному рядку, між ними ставлять двокрапку, наприклад:

```
S=A+B : S=S+1
```

Якщо оператор не поміщається на одному рядку, то після символу "\_" його продовжують на слід рядку.

5. На початку – всі змінні оголошують. **Dim** — від слова **dimension**, що означає величина. Якщо, наприклад, не оголосити A=3 та B=5, то після **S=A+B**, у змінній S буде записано "35". Тобто транслятор зрозумів A та B як як строкові змінні і "зчепив" ці дві строки.

6. Введення даних з клавіатури організовано через функцію **InputBox()**. Виведення даних у стандартне Windows-вікно організовано через функцію **MsgBox()**. Дужки у **InputBox()** і **MsgBox()** означають, що це функції.

7. Після набору тексту VA1.bas у Провіднику слід зробити DoubleClick по файлу. При цьому буде запущено програмну систему **Visual Basic**. І після натискання **F5** програма буде виконуватись. Якщо українські букви будуть спотворені, спробуйте зробити встановлення — Tools — Options — Editor Format — вибрати — Font — Courier New CYR.

8. Цю програму слід виконати також пооператорно – F8 (Трасувати програму). Прослідкуйте за значеннями змінних A, B та S. Для цього зробіть правий Click по кожній змінній, потім – Add Watch . Зауважте, що програма виконується не відразу, а пооператорно зліва направо, а також зверху донизу.

9. Якщо Ви переконалися, що програма працює так, як Ви її задумали, необхідно в папці Вашої групи ( наприклад ГКЗ-11) створити EXE-файл. Щоб Створити EXE-файл треба: File — Make Project1.exe. При цьому необхідно змінити назву файлу EXE: VA1.exe.

10. Це завдання можна вирішувати у середовищі MS Word (чи Excel, чи Access), що містить у собі VB 6.3. Викликати VB можна так: Сервіс — Макрос — VB (або Alt-F11).

Прочитати текст програми в редакторі Word – VB можна командою File — Import File. Далі ліворуч — у вікні проект — папка Modules — вказати Module1, а потім — View-Code (або F7).

Зберігати файл \*. bas слідує через File — Export File. Є деякі переваги роботи в VB вбудованому в Word, зокрема, тут працює F1-довідка за функціями та ключовими словами. Виконання макросу в Excel – **Alt+F8**. В Word можна програмувати форму. Для цього: меню Insert — UserForm.

11. Вийдіть з Visual Basic через File або Alt+Q. Знайдіть Ваш каталог і Вашу першу програму (EXE-файл), запустіть її на Виконання.

12. Якщо зробити першу програму в Word, то студент зможе виконувати всі наступні програми вдома на своєму комп'ютері (загальні та за варіантами). У першому пункті також навести створений cmd-файл для резервного копіювання даних.

Студенти повинні досліджувати, що буде після  $S=A+B$ , якщо A і B не оголошувати. У кожній програмі, має бути авторство за допомогою параметрів **InputBox** та **MsgBox**, наприклад:

```
MsgBox "Сума = " & S, vbOKOnly , "Вітрів"
```

```
MsgBox "Сума = " & S, vbInformation , "Вітрів"
```

або MsgBox "Сума = " & S, , "Студент Вітрів".  
Налагодження.

13. Запитання студенту:

1. Змінні якого типу використовуються у програмі?
2. Що означає "трасувати програму" ?
3. Як увімкнути змінні у вікно Add Watch ?
4. Правий Click за функції Definitin .
5. Ctrl+Space, Ctrl+i, Ctrl+j.

14. Покажіть програму викладачеві. Продемонструйте викладачеві Ваші навички трасування. Будьте готові відповісти на питання.

## БЛОК 1. ЗАВДАННЯ 2

Зразок вирішення цього завдання можна переглянути на відео:

<https://youtu.be/1BxM3DNd2ok>

Інші відео автора:

[https://www.youtube.com/channel/UCFZglBgYat1GGPMCPQ4XbJQ?sub\\_confirmation=1](https://www.youtube.com/channel/UCFZglBgYat1GGPMCPQ4XbJQ?sub_confirmation=1)

1. Вирішити попереднє завдання ввівши діагностику даних, що вводяться з використанням функції **IsNumeric**.

2. Нижче наведено фрагмент перевірки на **IsNumeric** у циклі **Do - Loop**:

```
Do
рядок$ = InputBox ("Введіть перше число", _
                    "Ветров , ГД- 11")
If IsNumeric(рядок) = True Then Exit Do
Loop
x% = CInt(рядок)
' або x# = CDb1(рядок)
' CStr() - повертає String
```

3. Вивести результат рішення за допомогою функції **MsgBox** у два рядки, перший рядок: "Сума = " & S, а другий рядок — "Програмував — прізвище студента". Перенесення на інший рядок організувати за допомогою **Chr(10)** або **vbCrLf** (або **vbNewLine**). Тут **Chr(10)** — переведення на інший рядок; замість **Chr(10)** можна використовувати **vbCrLf**.

## БЛОК 2. ЕЛЕМЕНТИ УПРАВЛІННЯ НА ФОРМІ

### БЛОК 2. ЗАВДАННЯ 3

Зразок вирішення цього завдання можна переглянути на відео:  
<https://youtu.be/A9Dt7KkSOAU>

Якщо студент програмує у середовищі VBA, слід вирішувати завдання в наступною постановкою (оскільки в VBA немає функції print). Створимо програму (Windows-додаток) мовою **Visual Basic**. Ця програма виводить у центр екрану стандартне Windows-вікно (форму). Вгорі форми – прізвище студента. На формі текстове поле **TextBox** та командна кнопка "Витягнути корінь". Користувач вводить у текстове поле число, і при натисканні на кнопці виводиться результат за допомогою функції **MsgBox()**.

Якщо ні:

Створимо програму (Windows-додаток) мовою Visual Basic. Ця програма виводитиме в центр екрана стандартне Windows-вікно (Форму). Внизу форми – одна кнопка: На ній написано "Вихід". Click у якому місці форми викликає поява тексту "Прізвище студента" (використовуємо функцію print). DoubleClick очищає форму від усіх написи. Click на кнопці "Вихід" призводить до виходу із програми.

Приступаємо до програмування поставленого завдання.

1. Запустіть VB:

Пуск — Програми — Microsoft Visual Basic 6.0 — Microsoft Visual Basic.  
В MS Excel, в MS Word та інших додатках MS Office також можна програмувати форму. Для цього: меню Insert — UserForm

2. Для створення програми — створюємо новий проєкт (New Project).

До складу проєкту входять:

- Файл проєкту \*.vbp ;
- Файли форм \*.frm
- Файли основних модулів \*.bas
- Файли модулів класів \*.cls ;

3. Щоб створювана програма відповідала стандартам Windows—додатки, вибираємо (відкриваємо) тип проєкту — Standard.exe. У результаті — отримуємо на екрані форму — стандартне вікно (заготівлю), яке лежатиме в основі інтерфейсу користувача.

4. Форма на екрані — це об'єкт, як і об'єктами будуть кнопки на цій формі, текстові поля, смуги прокручування і так далі.

Кожен об'єкт має властивості. Праворуч від форми розташовується вікно цих властивостей (Properties) об'єкта. Якщо це вікно відсутнє, його слід відновити через правий клік об'єктом. У цьому вікні перераховані всі характеристики об'єкта, наприклад, його геометрія (Position) та ін. Ці властивості – характеристики можна переглянути в алфавітному порядку (alphabetic) або за категоріями. Виберіть властивість Caption (заголовок, напис) та напишіть тут текст "Прізвище студента". Click у довільному місці форми завершує виведення тексту у форму.

5. Ця програма повинна мати два об'єкти: форму та кнопку. Об'єкт містить у собі властивості та процедури (методи). Крім того, для кожного об'єкта можна створювати процедуру обробка події.

6. У властивостях форми також задамо параметр StartUpPosition , який вказує розташування форми під час старту Вашої програми. Задайте CenterOwner, цей параметр розташує форму на центрі екрана.

7. Зліва форми маємо меню General, це меню елементів управління, які ви можете встановити на формі. Наприклад, організувати керуючу кнопку на формі (**CommandButton**), організувати введення тексту (текстове поле — **TextBox**), написати будь-який текст (Label), просто вивести на формі відрізок прямий (Line) і так далі. Меню General можна сховати шляхом "x", а відновити — View — Toolbox або кнопкою Toolbox на стандартній панелі (Standard). Сховати-відновити панель Standard можна через View - Toolbars - Customize...- Standard.

8. Відповідно до умови завдання необхідно організувати кнопку виходу з Вашої програми. Для цього в меню General відзначити вказівником мишки на елементі **CommandButton**. Далі розтягуючи прямокутник на формі, вкажемо розташування кнопки.

9. У вікні праворуч, у властивості Caption цієї кнопки вкажемо її назву: "Вихід".

10. До цього моменту ми запрограмували два об'єкти: форму та кнопку. Для кожного об'єкта існує набір стандартних подій. Подією є будь-який вплив від миші або клавіатури. Їх можна переглянути за допомогою об'єкта DoubleClick. Наприклад, DoubleClick за кнопкою "Вихід" покаже вікно коду, де праворуч вгорі перераховані всі можливі стандартні події, є натискання різних кнопок, у тому числі і Click. Click на кнопці "Вихід" означає вихід із програми. Тому в тій, що з'явилася заготовці:

```
Private Sub Command1_Click()
```

```
End Sub
```

між цими двома рядками пишемо: End (або Unload Me або Unload Form1), тобто вихід із створюваної нами програми.

Всі можливі події за кнопкою "Вихід" нами запрограмовані. Можна вийти з вікна коду, закривши його - "x".

11. Ще необхідно програмувати виведення тексту на форму. вже зараз запустити те, що є. Пункт меню Run – Start або F5, або кнопку Start на стандартній панелі інструментів (стрілка). Ваша програма успішно стартувала, переконайтесь, що працюють стандартні кнопки Повернути, Відновити, Закрити, а також кнопка "Вихід".

12. Організуємо виведення тексту на форму. Для цього – подвійний Click на формі та праворуч вгорі — вибираємо подію Click. Потім між двома рядками в заготівлі, що з'явилася, напишемо:

**Print "Друга програма".**

Таким чином ми обробили подію **Click** .

13. Відповідно до умови завдання DoubleClick повинен очищати форму від усіх написів. Для цього – праворуч вгорі – вибираємо подію DblClick. Потім між двома рядками в заготівлі, що з'явилася, пишемо: Form1.Cls. Ця функція (метод) очищує форму. Закриємо вікно коду – "x".

14. Тепер Ваша перша програма має працювати. Запустіть її (наприклад, F5). Вам слід протестувати всі гілки і (можливості) програми, тобто Click на формі, подвійний Click, стандартні кнопки згорнути програму, відчинити вікно, закрити програму, кнопка "Вихід".

15. Якщо Ви переконалися, що програма працює так, як Ви її задумали, необхідно в папці Вашої групи (наприклад GST11) створити

EХЕ- файл . Щоб Створити EХЕ- файл : File - Make Project1.exe.

При цьому необхідно змінити назву EХЕ-файлу – VA2.exe.

16. Вийдіть з Visual Basic через File або Alt+Q . Знайдіть Ваш каталог та Вашу першу програму (EХЕ-файл), запустіть її на Виконання.

17. У поясненнях до програми (у 4-му пункті) необхідно намалювати у Paint форму з авторством (скриншот з VB), кнопку, текстове поле, текстові позначки. Від кожного елемента намалювати стрілку в Paint і написати назву кожного елемента керування.

18. Запитання студенту:

1. Як сховати, а потім відновити меню General ?
2. Як сховати, та був відновити властивості об'єкта ?
3. Як перемикається між властивостями об'єктів ?
4. Що містять у собі об'єкти ?

5. Як (чим) управляються об'єкти ?
6. Як створити процедуру обробки події ?

19. Покажіть програму викладачеві. Будьте готові відповісти на питання.

## БЛОК 2. ЗАВДАННЯ 4. ОДИННА ДЛЯ ВСІХ

Зразок вирішення цього завдання можна переглянути на відео:

<https://youtu.be/YwaGF8EirzE>

Інші відео автора:

[https://www.youtube.com/channel/UCFZglBgYat1GGPMCPQ4XbJQ?sub\\_confirmation=1](https://www.youtube.com/channel/UCFZglBgYat1GGPMCPQ4XbJQ?sub_confirmation=1)

Створимо програму (Windows-додаток) мовою Visual Basic. Ця програма виводитиме на екран стандартне Windows-вікно (Форму). У центрі цього вікна – текстова мітка "Прізвище студента". Необхідно на цю мітку вивести текст червоним кольором: `Label1.ForeColor = vbRed`. Текст у мітці зробити жирним: `Label1.FontBold = True`, розміром: `Label1.FontSize = 16`. Внизу — дві кнопки: На одній написано "Текст", на іншій — "Вихід". Click на кнопці "Текст" призводить або до зникнення тексту "Прізвище студента", або до появи знову цього тексту. Click на кнопці "Вихід" призводить до виходу із програми.

Приступаємо до програмування поставленого завдання.

1. Запустіть VB.
2. Створюємо новий проєкт (New Project) типу Standard.exe. При цьому отримуємо на екрані форму – стандартне вікно (заготівлю), яке буде в основі інтерфейсу користувача.
3. Відповідно до умови завдання необхідно приблизно у центрі форми вивести текст "Прізвище студента". Для цього в меню General натисніть мишею по елементу Label , а на формі розтягнемо прямокутник, де згідно на нашому задуму розташовуватиметься текст "Прізвище студента".
4. Праворуч, у вікні властивостей (Properties) об'єкта Label виберіть властивість **Caption** (заголовок, напис) та напишіть тут текст "Третя" програма".

5. Клацніть знову по тексту та у властивості об'єкта "(Name)" напишіть lbl (скорочення від Label ). Ім'я об'єкта lbl знадобиться нам у посилання на нього з кнопки, що управляє.

6. Далі цього тексту вкажемо, що текст невидимий. Для цього в полі властивості Visible вкажемо False .

7. Переходимо до організації кнопок. Для цього в меню General натисніть мишею по елементу **CommandButton** та шляхом розтягування прямокутника на формі вкажемо розташування однієї та другої кнопки.

8. У властивостях Caption цих кнопок вкажемо їх назви: "Текст" та "Вихід". Нагадаємо, що кнопка "Текст" має або виводити текст "Прізвище студента" або ховати цей текст.

9. До цього моменту ми запрограмували чотири об'єкти: форма, текст "Прізвище студента" та дві кнопки. Для кожного об'єкту існує набір стандартних подій у тому числі і Click . Click по кнопці "Вихід" означає вихід із програми. Тому в заготівлі, що з'явилася:

```
Private Sub Command2_Click()
```

```
End Sub
```

між цими двома рядками пишемо: End (або Unload Me або Unload Form1), тобто вихід із створюваної нами програми. Всі можливі події за кнопкою "Вихід" нами запрограмовані. Можна вийти з вікна коду, заклавши його - "x".

10. Складніше із кнопкою "Текст". DoubleClick на цій кнопці видасть вікно коду:

```
Private Sub Command1_Click()
```

```
End Sub
```

Між цими двома рядками коду слід записати: якщо властивість **Visible** об'єкта **lbl=False**, то цю властивість змінити на **True**, а інакше – навпаки. Мовою **Visual Basic** це виглядатиме так:

```
If lbl.Visible = False Then  
    lbl.Visible = True  
Else  
    lbl.Visible = False  
End If
```

Вам слід вставити ці п'ять рядків між рядками вікна коду.

11. Тепер Ваша друга програма має працювати. Надішліть її на Виконання - F5. Вам слід протестувати усі гілки (можливості) програми, тобто Click на кнопки "Текст", "Вихід", стандартним кнопкам: згорнути програму, розкрити вікно, закрити програму.

На кнопку можна натискати мишкою, клавішею Enter та клавішею Space. Перемикається між формою та програмним кодом можна, наприклад, таким чином: F7 та Shift-F7. Спробуйте.

12. Якщо Ви переконалися, що програма працює так, як Ви її задумали, необхідно у папці Вашої групи створити EXE-файл. Щоб створити EXE-файл: File - Make Project3.exe. При цьому слід змінити назву файлу EXE, наприклад, VA3.exe. Запустіть її на виконання.

13. У поясненнях до програми (у 4-му пункті) необхідно в Paint на фоні інтерфейсу програмного коду показати меню, що розкривається об'єктів проекту та вказати стрілкою будь-який з цих об'єктів.

14. Крім того, намалювати інший малюнок, також на тлі інтерфейсу програмного коду Тут показати меню подій, що розкривається та вказати стрілкою будь-якої з цих подій.

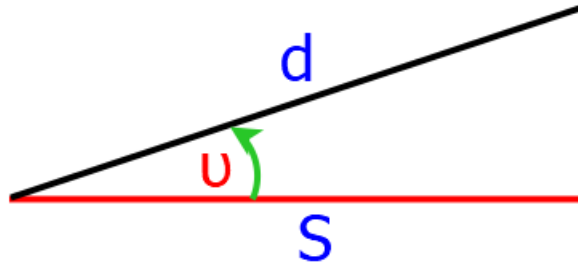
15. Запитання студенту:

1. Як отримати вікно коду для того чи іншого об'єкта ?
2. Як обробити ту чи іншу подію (**Click**, **Double Click** і т.д.) для того чи іншого об'єкта ?
3. Чи можна число 2.4 записати в змінну типу Integer (Long, Single , Double , Boolean, Byte )?

16. Покажіть програму викладачеві. Будьте готові відповісти на питання.

## БЛОК 2. ЗАВДАННЯ 5. ОДИННА ДЛЯ ВСІХ

Між двома точками за допомогою електронного тахеометра виміряна похила [наклонное расстояние] відстань  $d$  та кут нахилу  $v$  (ню). Програма має обчислити горизонтальне прокладання  $S = d \cdot \cos v$ .



Необхідно організувати інтерфейс користувача у вигляді стандартного Windows-вікна (форми), де буде передбачено введення похилої відстані та кута нахилу за допомогою елемента управління **TextBox** (Текстове поле). Виведення результату обчислень (тобто горизонтального прокладання) на екран (на форму), слід організувати за допомогою елемента керування **Label** (Мітка).

Приступаємо до програмування поставленого завдання.

1. Запустіть VB.
2. Створюємо новий проєкт типу Standard.exe. В результаті — отримуємо на екрані форму – стандартне вікно (заготівлю).
3. У заголовку форми (Caption ) напишемо: "Обчислення горизонтального прокладання".
4. Починаємо організовувати введення даних. За допомогою Label (Мітка) вгорі форми пишемо текст: "Введення похилої відстані :".
5. Навпаки цього тексту передбачаємо текстове поле — **TextBox**, яке заповнюватиме користувач Вашої програми.
6. Наступним рядком аналогічно організуємо введення кута нахилу (тільки градуси).
7. Таким чином, на екрані ми маємо 5 об'єктів: форма, дві мітки Label і два текстові поля.
8. У такому вигляді можна вже запускати на виконання даний проєкт (F5) та спробувати як працює введення даних, їх редагування, перехід з одного текстового поля в інше е (у тому числі за допомогою клавіші Tab).
9. У властивостях Text обох текстових полів характеристики Text1 та Text2 замінимо на пробіли. Таким чином при запуску Вашої програми текстові поля будуть порожніми.

10. Додамо дві керуючі кнопки (**CommandButton**). На одній напишемо "Обчислити", а на другий — "Вихід".

11. У програмному кодї кнопки "Вихід" обробимо подію — Click на кнопці "Вихід". Аналогічно, як ми робили у попередньому завданні, між цими двома рядками коду пишемо: End (або Unload Me , або Unload Form1).

12. Додамо за допомогою Label (Мітка) останній рядок на формі "Горизонтальне прокладання дорівнює:" .

13. Також за допомогою Label протягом останнього рядка ще одну мітку, де виводитиметься безпосередньо результат. Її заголовок зробимо порожнім.

14. Розрахунок буде здійснюватись при обробці події Click за кнопкою "Обчислити". Для цього DoubleClick за цією кнопкою, і для події Click між рядками вводимо такий код:

```
' Оголошення змінних:  
    Dim d!, nu !, PI#  
' !-суфікс типу Single , #- суфікс типу Double  
    PI = 4 * Atn(1)  
    d = Text1.Text  
    nu = Text2.Text  
    nu = nu * PI / 180  
    d = d * Cos (nu)  
    Label4.Caption = d
```

У цьому кодї — перші три рядки — це оголошення змінних: d — похила відстань; nu - кут нахилу; PI – це значення  $\Pi = 3.14159\dots$  Для отримання більшої кількості вірних знаків обчислюємо  $\Pi$  як 4 помножити на  $\arctg(1)$ . В операторі `d = Text1.Text` - Text1 - це ім'я об'єкта - першого текстового поля, а Text – це одна з його властивостей.

Оператор `nu = nu * PI / 180` переводить кут нахилу із градусів у радіани. Після обчислення d, виводимо його у четвертій мітці – властивість — Caption.

Докладніше про будь-яку функцію можна з'ясувати наступним шляхом: правий Click за функцією та definition (визначення) або Shift+F2.

15. Стартуйте Вашу програму, задайте значення похилої відстані 123 м, а значення кута нахилу – 3 градуси. Після "Обчислити" - отримане значення горизонтального прокладення проконтролюйте за допомогою калькулятора.

16. Створіть файл EXE.

17. Студент знає, як працювати з файлами \*.bas у середовищі MS Excel. (Див. п. 10 програми 1). Працювати з формою та елементами управління (текстовими полями, мітками, кнопками тощо) можна також у середовищі Excel. Для цього: запустити Excel => Alt + F11 => Insert => UserForm. Зберігати файли форми \*.frm слід через File — Export File.

18. У поясненнях до програми (у 4-му пункті) необхідно в MS Paint на фоні інтерфейсу програмного коду показати меню, що розкривається властивостей та методів будь-якого об'єкта та вказати стрілкою будь-який метод та іншою стрілкою — якась властивість.

19. Запитання студенту:

1. Для чого в програмі використано функцію Ath (арктангенс) ?

2. Намалюйте малюнок, покажіть: похила відстань, кут нахилу, горизонтальне прокладання.

3. Що подаємо на вхід функції Cos() і що отримуємо на виході ?

20. Покажіть програму викладачеві. Будьте готові відповісти на питання.

## БЛОК 2. ЗАВДАННЯ 6

### ВАРІАНТИ ДЛЯ ЗАВДАННЯ 6 (17 варіантів)

#### ВАРІАНТ 1

На формі мітка та дві кнопки "Вихід" та "Змінити". На мітці написано: "Незабаром літо !" За командою "Змінити" програма змінює напис на мітці "Незабаром зима !". При вирішенні цього завдання можна використовувати змінну **t** типу **Boolean** (її поставити як зовнішню змінну).

Її початкове значення **t = True** можна задати під час обробки з побуту завантаження форми **Sub Form Load ()**, а змінювати її значення на протилежне можна при обробці події **Click** за кнопкою "Змінити", наприклад, таким чином: **t = t Xor True**, або **t = Not t**.

При зависанні мишею над кнопкою з'являється підказка "Натисніть мене". Підказку забезпечує властивість кнопки

**Command1.ToolTipText = "Натисніть мене",**

Цю властивість слід вказати під час завантаження форми.

#### ВАРІАНТ 2

На формі мітка та кнопка а "Вихід". На мітці написано: "Хочу в Париж !" клік по мітці викликає зміну тексту мітки з "Хочу в Париж !" на "Хочу в Лондон!" При вирішенні цього завдання можна використовувати змінну **t** типу **Boolean** (її поставити як зовнішню змінну). Її початкове значення **t =**

True можна задати при обробці події завантаження форми **Sub Form\_Load()**, а міняти її значення на протилежне можна при обробці події Click за міткою, наприклад таким чином:

**t = t Xor True, або t = Not t.**

### ВАРІАНТ 3

На формі етикетці і три кнопки. На мітці написано: "Хороша погода", на одній кнопці написано: "Напис 1", на іншій кнопці — "Напис 2". Клік по першій кнопці викликає зміну тексту мітки на "Погана погода", а клік по другій кнопці — "Гарна погода". Клік по кнопці "Вихід" викликає вихід з програми. При зависанні мишею над кнопкою з'являється підказка, наприклад, "Натисні мене". Підказку забезпечує властивість кнопки

**Command1.ToolTipText = "Натисніть мене",**

Цю властивість слідую вказати під час завантаження форми.

### ВАРІАНТ 4

На формі мітка, текстове поле **TextBox** та кнопка "Скопіювати". На мітці та в текстовому полі нічого не написано, тобто при обробці події завантаження форми **Form\_Load** задаємо **Label1.Caption = ""** (або = **Empty**) та **Text1.Text = ""**. При старті програми користувач вводить у текстове поле будь-який текст, наприклад, своє прізвище і при натисканні на кнопку "Скопіювати" цей текст копіюється на позначку. При зависанні мишею над кнопкою з'являється підказка, наприклад, "Натисніть мене".

Підказку забезпечує властивість кнопки

**Command1.ToolTipText = "Натисніть мене",**

Цю властивість слідую вказати під час завантаження форми.

### ВАРІАНТ 5

На формі мітка, текстове поле **TextBox** та дві кнопки: "Скопіювати" та "Скинути". На мітці написано: "Напишіть що-небудь у текстовому полі", а в текстовому полі нічого не написано, тобто при обробці події завантаження форми **Form\_Load** задаємо **Label1.Caption = "Напишіть що-небудь у текстовому полі"** та **Text1.Text = ""**. При старті програми користувач вводить у текстове поле будь-який текст, наприклад, своє прізвище і при натисканні на кнопку "Скопіювати" цей текст копіюється на позначку. При натисканні на кнопку "Скинути" текстове поле очищається, на мітці знову написано: "Напишіть щось у текстовому полі". При зависання мишею над кнопкою з'являється підказка, наприклад, "Натисни мене". Підказку забезпечує властивість кнопки

**Command1.ToolTipText = "Натисніть мене",**

Цю властивість слідую вказати під час завантаження форми.

## ВАРІАНТ 6

На формі мітка, два текстові поля **TextBox** та кнопка "Скласти". У ці два текстові поля користувач вводить будь-які числа, а програма за командою "Скласти" виводить результат складання в **MsgBox** зі словами "Сума =". У мітці при обробці події завантаження форми `Form_Load` задаємо **Label1.Caption = "Введіть два числа"**. Дозволити користувачу вводити лише числа для цього слід передбачити діагностику даних, що вводяться в текстові поля. Для цього створити порожній обробник події перевірки введеного текстове поле `Text1_Validate`. В обробнику вкажемо: **If IsNumeric (Text1) = False Then MsgBox "Неправильне введення"**. Крім того, при зависанні мишею над кнопкою з'являється підказка, наприклад, "Натисні мене". Підказку забезпечує властивість кнопки **Command1.ToolTipText = "Натисніть мене"**, Цю властивість слід вказати під час завантаження форми.

## ВАРІАНТ 7

Написати програму, яка вводить через текстове поле число при натисканні на командну кнопку витягує з нього квадратний корінь та виводить результат на мітку `Label1`. Дозволити користувачеві вводити лише числа для цього слід передбачити діагностику даних, що вводяться в текстові поля. Для цього створити порожній обробник події перевірки введеного текстове поле `Text1_Validate`. В обробнику вкажемо:

```
If IsNumeric(Text1) = False Then _  
    MsgBox "Неправильне введення".
```

Є ще одна кнопка "Очищення" для обнулення текстового поля та мітки. Для цього скористайтеся, наприклад, такою командою:

```
"Label1.Caption = Empty".
```

При зависанні мишею над кнопкою з'являється підказка, наприклад, "Натисні мене". Підказку забезпечує властивість кнопки

```
Command1.ToolTipText = "Натисніть мене",
```

Цю властивість слід вказати під час завантаження форми.

## ВАРІАНТ 8

Написати програму для введення пароля в текстове поле, причому при введенні замість символів, що вводяться, " перебуває за спиною користувача" побачить лише зірочки. Програма складається з форми `Form1`, текстового поля `Text1`, мітки `Label1`, куди ми копіюватимемо паспорт (секретні слова) та кнопки `Command1` "Покажи паспорт". Тут при завантаженні форми задати: **Text1.PasswordChar = "\*" Text1.Text = ""**. При зависанні мишею над кнопкою з'являється підказка, наприклад, "Натисні мене". Підказку забезпечує властивість кнопки

```
Command1.ToolTipText = "Натисніть мене",
```

Цю властивість слід вказати під час завантаження форми.

## ВАРІАНТ 9

На формі мітка та дві кнопки. На мітці написано: "Стиль шрифту". На одній кнопці "Змінити", на іншій "Вихід". При натисканні на кнопку "Змінити" стиль шрифту мітки змінюється на жирний, тобто `Label1.FontBold = True`. Для написання цієї програми слід опрацювати подію клік по кнопці "Змінити". Міняти властивість мітки `FontBold` на протилежне доцільно такою командою:

```
Label1.FontBold = True Xor Label1.FontBold.
```

При зависанні мишею над кнопкою з'являється підказка, наприклад, "Натисни мене". Підказку забезпечує властивість кнопки

```
Command1.ToolTipText = "Натисніть мене".
```

Цю властивість слід вказати під час завантаження форми.

## ВАРІАНТ 10

На формі мітка, на якій написано прізвище та ім'я студента та кнопка "Сховати-Показати". Користувач за допомогою кнопки робить текст мітки невидимим, та був знову видимим тощо. При зависанні мишею над кнопкою з'являється підказка "Натисни мене". Підказку забезпечує властивість кнопки

```
Command1.ToolTipText = "Натисніть мене",
```

Цю властивість слід вказати під час завантаження форми. Забезпечити видимість чи невидимість тексту мітки можна властивістю `Label1.Visible` (видимий) з використанням оператора умови або побітового оператора `Xor`:

```
Label1.Visible = True Xor Label1.Visible
```

або

```
Label1.Visible = Not Label1.Visible
```

при обробці події клік на кнопці.

## ВАРІАНТ 11

Напишіть програму, яка запрошує користувача ввести у текстове поле радіус  $R$ , щоб обчислити довжину кола. Результат обчислень виведемо у діалогове вікно `MsgBox()`. Дозволити користувачеві вводити лише числа для цього слід передбачити діагностику даних, що вводяться в текстові поля. Для цього створити порожній обробник події перевірки введеного текстового поля `Text1_Validate`. В обробнику вкажемо:

```
If IsNumeric(Text1) = False Then MsgBox "Невірне введення".
```

Є ще одна кнопка "Очищення" для обнулення текстового поля та мітки. Для цього скористайтеся, наприклад, такою командою:

```
"Label1.Caption = Empty".
```

При зависанні мишею над кнопкою з'являється підказка "Натисни мене". Підказку забезпечує властивість кнопки

```
Command1.ToolTipText = "Натисніть мене",
```

яке слід вказати під час завантаження форми.

## ВАРІАНТ 12

На формі мітка та дві кнопки. На мітці написано: "Стиль тексту". На одній кнопці " Жирний - Не жирний", на іншій – "Курсів – Вертикальний". При натисканні на кнопку стиль шрифту мітки змінюється на курсив, тобто `Label1.FontItalic = True` . Для написання цієї програми слід опрацювати подію клік по кожній із кнопок. Міняти властивість мітки `FontItalic` на протилежне доцільно такий командою:

```
Label1.FontItalic = True Xor Label1.FontItalic або  
Label1.FontItalic = Not Label1.FontItalic
```

Аналогічно для властивості `Label1.FontBold`.

## ВАРІАНТ 13

Маємо у своєму каталозі текстовий файл `txt.txt`, у якому записані три рядки: ваше прізвище, ім'я, по батькові. Маємо форму, кнопку "Читати", кнопку "Очистити" та кнопку "Вихід". При натисканні на кнопки "Читати" вміст файлу повинен виявитися на формі. При натисканні на кнопки "Очистити" форма звільняється від усіх записів на ній. Для цього відкриваємо файл для читання: **"Open "txt.txt" For Input As #1"**.

Далі в циклі `Do — Loop` читаємо файл за рядками: `Input #1, стор "` в рядкову змінну `З` тр. Цю змінну виводимо на форму: `Print стор "`. Цикл з передумовою працює до досягнення кінця файлу: **While EOF(1) = False** ". Після роботи оператора циклу текстовий файл закриваємо: **"Close #1"**. Очевидно, програмуємо та інші дві кнопки.

## ВАРІАНТ 14

Маємо у своєму каталозі текстовий файл `txt.txt`, у якому записані три рядки: ваше прізвище, ім'я, по батькові. Маємо форму, кнопку "Читати", кнопку "Очистити" та кнопку "Вихід". Крім того, на формі маємо текстову позначку `Label1`. При натисканні на кнопки "Читати" вміст файлу має опинитися на мітці. Вивести на мітку ті кст червоним кольором: `Label1.ForeColor = vbRed` . При натисканні на кнопку "Очистити" всі записи на мітці повинні зникнути. Для цього відкриваємо файл для читання:

```
"Open "txt.txt" For Input As #1"
```

Далі в циклі `Do – Loop` читаємо файл по рядкам:

```
"Line Input #1, Стр"
```

в строкову змінну `Стр`. Далі в тілі цього ж циклу властивості мітки `Caption` присвоюємо значення змінної `Стр`:

```
"Label1.Caption = Label1.Caption + Стр + vbCrLf"
```

Тут `vbCrLf` (або `vbNewLine`) — це невидимий символ для перенесення на наступний рядок. Цикл з передумовою працює до досягнення кінця файлу:

```
"Do While EOF(1) = False"
```

Після роботи оператора циклу текстовий файл закриваємо: **"Close #1"**. Аналогічно програмуємо інші дві кнопки.

## ВАРІАНТ 15

Маємо у своєму каталозі текстовий файл txt.txt, у якому записані три рядки: ваше прізвище, ім'я, по батькові. Маємо форму, кнопку "Читати", кнопку "Очистити" та кнопку "Вихід". Крім того, у формі маємо текстове поле **Text1**. При натисканні на кнопці "Читати" вміст файлу має опинитися на мітці. При натисканні на кнопці "Очистити" текстове поле звільняється від усіх записів на ній (**Text1.Text = Empty**). Для цього відкриваємо файл для читання: "Open "txt.txt" For Input As #1". Далі в циклі **Do – Loop** читаємо файл рядок за рядком: "**Line Input #1, Стр**" в рядкову змінну **Стр**. Далі в тілі цього ж циклу: властивості текстового поля **Text** присвоюємо значення змінної **Стр**:

**"Text1.Text = Text1.Text + Стр + vbCrLf"**.

Тут **vbCrLf** (або **vbNewLine**) — це невидимий символ для перенесення наступного рядка. Цикл з передумовою працює до досягнення кінця файлу:

**"Do While EOF(1) = False"**.

Після роботи оператора циклу текстовий файл закриваємо: "Close #1". Властивості **MultiLine** текстового поля слід присвоїти значення **True**. Але зробити це потрібно в конструкторі, але не програмному коді. Очевидно, програмуємо та інші дві кнопки.

## ВАРІАНТ 16

На формі маємо командну кнопку "Вихід", текстову мітку та таймер (Timer). Поза всіма процедурами оголошуємо змінну і (лічильник) цілого типу. При обробці події завантаження форми обнулюємо лічильник, задаємо **FontSize** мітки рівним 16, задаємо інтервал таймера = 1000 мілісек = 1 сек. При обробці єдиного події таймера **Timer** наращуємо лічильник на одиницю та виводимо значення лічильника на текстову мітку **Label1**. При обробці події **Click** на кнопці "Вихід" організуємо вихід із програми командою **Unload Form1**.

## ВАРІАНТ 17

На формі маємо командну кнопку "Вихід", текстову мітку та таймер (Timer). Поза всіма процедурами оголошуємо змінну і (лічильник) цілого типу. При обробці події завантаження форми обнулюємо лічильник, задаємо **FontSize** мітки рівним 16, на мітці задаємо свої прізвище, ім'я та по батькові, задаємо інтервал таймера = 1000 мілісек = 1 сек. При обробці єдиної події таймера **Timer** наращуємо лічильник на одиницю. Наступним оператором обчислюємо значення **k**. Значення змінної **k** дорівнюватиме залишку від поділу на три: **k = Mod 3**. Далі при **k = 0** призначаємо колір мітки **ForeColor = vbRed** при **k=1** призначаємо колір мітки = **vbBlue** і при **k=2** призначаємо колір мітки = **vbGreen**. При обробці події **Click** на кнопці "Вихід" організуємо вихід із програми командою **Unload Form1**.

## БЛОК 3. МАСИВИ І ЦИКЛИ

### БЛОК 3. ЗАВДАННЯ 7. ОДИННА ДЛЯ ВСІХ

Зразок вирішення цього завдання можна переглянути на відео:

<https://youtu.be/jdmOnIEviKI>

1. Типова задача про масиви та цикли: визначити суму елементів масиву **a(1 To 5)**. У масиві записано п'ять чисел: 6,7,8,4,5.

2. Щоб не відволікатися від суті завдання і щоб не ускладнювати налагодження не створюватимемо громіздкий інтерфейс введення даних. У головній процедурі **Main()** оголошіть масив **a(1 To 5)**. Потім надайте початкові значення елементам масиву:

**a(1) = 6: a(2) = 7: a(3) = 8: a(4) = 4: a(5) = 5**

3. Далі змінної S, куди Ви накопичуватимете суму елементів, призначте значення нуль.

4. Потім у тілі циклу **For i=1 To 5 ... Next** організуйте накопичення суми S.

5. Після налагодження Вами цієї програми покажіть викладачу результат.

### БЛОК 3. ЗАВДАННЯ 8. ОДИННА ДЛЯ ВСІХ

Зразок вирішення цього завдання можна переглянути на відео:

<https://youtu.be/UyiH9hOXuIY>

У масиві записано п'ять чисел: **6, -7, 8, -4, 5**. Написати програму для обчислення суми лише позитивних елементів масиву. При цьому ініціалізацію масиву (тобто присвоєння початкових значень) виконати через функцію

**a = Array(6, -7, 8, -4, 5).**

Зауважте, що при цьому згідно з синтаксисом VB цей масив не оголошується. Тут рахунок елементів масиву від 0 до 4, тобто.  $a(0)=6$  і т.д. Після налагодження Вами даної програми покажіть викладачеві результат.

### БЛОК 3. ЗАВДАННЯ 9. ОДИННА ДЛЯ ВСІХ

Програма має форму та дві кнопки. Під час обробки події завантаження форми на першій кнопці програмуємо напис "Виведення тексту", а на другий — "Вертикальний" (шрифт). При натисканні на першій кнопці спочатку очищаємо форму від попередніх написів: **Form1.Cls**. Потім призначаємо шрифт: `Font.Name = "Times New Roman" ' "Consolas"` (спробувати обидва).

Далі у циклі *i* (лічильник циклу) від 1 до 10 призначити `Font.Size = 10 + i` та `Print "Своє прізвище"; 10+i`.

Далі при обробці події `Click` на другій кнопці організувати перемикання шрифту вертикальний – похилий шрифт:

**Font.Italic =Not Font.Italic.**

При цьому має змінюватися напис на другій кнопці з "Вертикальний" на "Похилий" і навпаки.

### БЛОК 3. ЗАВДАННЯ 10

#### ВАРІАНТИ ДЛЯ ЗАВДАННЯ 10 (16 ВАРІАНТІВ)

##### ВАРІАНТ 1

Відомо, що число  $\pi$  можна знайти за такою формулою:

$$\pi = \sqrt{6 \cdot \sum \frac{1}{i^2}}.$$

Написати програму для обчислення числа  $\pi$  за цією формулою. Оскільки цей ряд сходиться повільно, і для того, щоб отримати 6 вірних знаків після коми потрібно 7 млн доданків цієї суми, необхідно лічильник циклу оголосити як **Long**, наприклад `i&`). Порівняйте отримане число  $\pi$  з числом  $\pi$ , одержаним за формулою `Pi = 4 * Atn(1)`. Покажіть викладачеві результат. Зауважте, що число  $\pi$  можна отримати за формулою  $\pi = (90 \cdot \sum \frac{1}{i^4})^{0.25}$ . У цьому випадку для отримання 9 вірних знаків знадобиться лише 1000 ітерацій.

##### ВАРІАНТ 2

Знайти суму числового ряду:

$$S = \sum_{k=1}^{1000} \frac{1}{k \cdot (k + 1)}.$$

Значення  $S$  має наближатися до **одиниці**.

### ВАРІАНТ 3

Відомо, що число  $\pi$  можна знайти за такою формулою:

$$\pi = (90 \cdot \sum \frac{1}{i^4})^{0.25}.$$

Написати програму для обчислення числа  $\pi$  за цією формулою. У цьому випадку для отримання 9 вірних знаків потрібно всього 1000 ітерацій. Порівняйте отримане число  $\pi$  з числом  $\pi$ , отриманим за формулою

$$\text{Pi} = 4 * \text{Atn}(1).$$

### ВАРІАНТ 4

Позначки точок траси представлені як елементи масиву. В цьому масиві знаходять різницю максимального та мінімального елементів для визначення розмаху епюри траси. Розмір розмаху враховується під час вибору розмірів (формату) креслення. У цьому завданні студентам слід вирішити лише першу частину проблеми: визначити максимальний елемент масиву.

### ВАРІАНТ 5

Відомо, що число  $\pi$  можна знайти за такою формулою:

$$\pi = \sqrt{8 \cdot \sum \frac{1}{(2 \cdot i - 1)^2}}$$

Написати програму для обчислення числа  $\pi$  за цією формулою.

Порівняйте отримане число  $\pi$  з числом  $\pi$ , отриманим за формулою:

$$\text{Pi} = 4 * \text{Atn}(1).$$

### ВАРІАНТ 6

Відомо, що число  $e = 2.718\dots$  (основу натуральних логарифмів) можна обчислити за такою формулою:

$$e = 1 + \sum \frac{1}{k!}.$$

Тут ! — факторіал, тобто, наприклад,  $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$ . Кожен наступний доданок суми  $A(i)$  залежить від попереднього таким чином:

$$A(i) = A(i - 1) * (1/i).$$

Написати програму для обчислення числа  $e$  за першою формулою. Порівняйте отримане число  $e$  числом  $e$ , отриманим за формулою

$$e = \text{Exp}(1).$$

*Рекомендація:* у тілі циклу написати дві Рядки:

$$a = a * (1 / i)$$

i

$$S = S + a.$$

Для початку організуйте цикл від 1 до 10 та вкажіть скільки вірних знаків забезпечує такий цикл.

#### ВАРІАНТ 7

Відомо, що число  $\pi$  можна знайти за такою формулою:

$$\pi = \sqrt{\sqrt{90 \cdot \sum \frac{1}{i^4}}}$$

Написати програму для обчислення числа  $\pi$  за цією формулою.

Порівняйте отримане число  $\pi$  з числом  $\text{Pi}$ , отриманим за формулою  $\text{Pi} = 4 * \text{Atn}(1)$ .

#### ВАРІАНТ 8

Знайти суму числового ряду:

$$S = \sum_{k=2}^{1000} \frac{1}{(k-1)(k+1)}$$

Відповідь має наближатися до 0.75.

#### ВАРІАНТ 9

Дуже зручно за допомогою циклів розв'язувати рекурентні рівняння. У цьому завданні необхідно вирішити рівняння

$$\text{Sin}(X/6) - 0.5 = 0$$

методом Ньютон. Метод Ньютона призводить до рекурентного співвідношення:

$$X_{i+1} = X_i - 6 \operatorname{tg} (X_i/6) + 3(1/\cos (X_i/6)).$$

Програмуючи цю формулу, отримаємо:

```

для i% = 1 до 3
x = x - 6 * Tan (x / 6) + 3 * (1 / Cos (x / 6))
Next

```

Метод Ньютона забезпечує швидку збіжність ітераційного процесу. Переконайтеся, що коренем рівняння виявилось всюдисуще число  $\pi$ . Зауважте, що тут достатньо лише трьох ітерацій для отримання 13 вірних знаків після коми. Покажіть викладачеві результат.

#### ВАРІАНТ 10

Знайти суму ряду  $1 + 2 + 3 + 4 + \dots$  до 15. Перевірити значення суми за такою формулою:

$$S = \frac{n \cdot (n + 1)}{2}$$

#### ВАРІАНТ 11

Знайти суму низки непарних чисел  $1 + 3 + 5 + \dots + (2n - 1)$ .  
Перевірити значення суми за такою формулою:  $S = n^2$ .

#### ВАРІАНТ 12

Позначки точок траси представлені як елементи масиву. В цьому масиві знаходять різницю максимального та мінімального елементів для визначення розмаху епюри траси. Розмір розмаху враховується під час вибору розмірів (формату) креслення. У цьому завданні студентам слід вирішити лише перша частина проблеми: визначити *мінімальний* елемент масиву.

#### ВАРІАНТ 13

Кут виміряний 10-ма круговими прийомами. Є результати вимірювань, виражені лише за секунди:

8.5" 8.3" 7.3" 7.1" 8.7"  
7.6" 8.4" 6.9" 8.2" 7.9"

Написати та налагодити програму для визначення середнього арифметичного значення з 10 вимірів, а також значення середньої квадратичної помилки (СКП) за формулою Бесселя:

$$m = \sqrt{\frac{\sum v^2}{n - 1}}$$

Результати вимірювань подати як масив використовуючи функцію **Array()**, а обчислення середнього значення та СКП організувати із застосуванням циклів **For – Next ( або Do – Loop)**.

#### ВАРІАНТ 14

Знайти суму числового ряду:

$$S = \sum_{k=1}^{1000} \frac{1}{(2 \cdot k - 1) \cdot (2 \cdot k + 1)}$$

Значення  $S$  має наближатися до **0.5**.

Знайдіть суму першої тисячі елементів цього ряду, і подивіться, чи вона наближається до 0.5.

#### ВАРІАНТ 15

Контрольний базис вимірянний мірним дротом 10 разів. Є результати вимірювань, виражені тільки в міліметрах:

<b>12</b>	<b>2</b>	<b>9</b>	<b>27</b>	<b>31</b>
<b>21</b>	<b>18</b>	<b>-3</b>	<b>14</b>	<b>25</b>

Для обох варіантів написати та налагодити програму для визначення середнього арифметичного значення з 10 вимірів, а також значення середньої квадратичної помилки (СКП) за формулою Бесселя:

$$m = \sqrt{\frac{\sum v^2}{n - 1}}$$

Результати вимірювань подати як масив використовуючи функцію **Array()**, а обчислення середнього значення та СКП організувати із застосуванням циклів

**For – Next ( або Do – Loop)**.

## БЛОК 4. БІЛЬШ СКЛАДНІ ЗАВДАННЯ

### БЛОК 4. ЗАВДАННЯ 11

Зразок вирішення цього завдання можна переглянути на відео:

<https://youtu.be/zOJXrJqfV6s>

1.Процедура зрівнювання геодезичних мереж МНК передбачає розв'язання нормальної системи лінійних алгебраїчних рівнянь (СЛАР). У цьому завданні студенту необхідно написати програму рішення СЛАР

$$\begin{cases} 10 \cdot x + 2 \cdot y + z = 17 \\ 2 \cdot x + 10 \cdot y + 5 \cdot z = 37 \\ x + 5 \cdot y + 10 \cdot z = 41 \end{cases}$$

методом Зейделя. Кожен студент вирішує цю систему попередньо змінивши a11 наступним чином: a11 = 10 + номер студента за списком. У цьому необхідно підібрати вільні члени те щоб ми знали заздалегідь розв'язання системи: x = 1; y = 2; z = 3. Слід враховувати, що ітераційний процес сходиться якщо в перетвореній матриці будь-яка сума абсолютних значень елементів стовпців < 1, а також хоча б для одного з рядків сума була б < 1.

2. Введення коефіцієнтів та правої частини організувати не треба. Насамперед напишіть т.зв. перетворені рівняння. Так для наведеної системи перетвореною буде система:

$$\begin{aligned} X &= 1.7 - 0 \cdot X - 0.2 \cdot Y - 0.1 \cdot Z; \\ Y &= 3.7 - 0.2 \cdot X - 0 \cdot Y - 0.5 \cdot Z; \\ Z &= 4.1 - 0.1 \cdot X - 0.5 \cdot Y + 0 \cdot Z. \end{aligned}$$

3. Вставте ці рівняння у тіло циклу:

**Do While умова роботи циклу**  
**Тіло циклу**  
**Loop**

4. Цей цикл буде працювати до тих пір, поки коріння в наступній ітерації не відрізняться від коренів у попередній ітерації на невелику величину. Критерієм відмінності коріння може бути S – сума квадратів різниць наступних та попередніх коренів:

$$S = [X(i+1) - X_i]^2 + [Y(i+1) - Y_i]^2 + [Z(i+1) - Z_i]^2$$

5. Умовою роботи циклу може бути S > 0.0000001. Крім того, вставте в тіло циклу лічильник **i=i+1** щоб дізнатися скільки ітерацій (повторень) знадобилося вирішення завдання.

6. Вивести рішення системи за допомогою **MsgBox()** у стовпчик:

**X = ...;**

**Y = ...;**

**Z = ...;**

**Кількість ітерацій = ...**

Перенесення на інший рядок організувати за допомогою **Chr(10)**, або **vbCrLf**, або **vbNewLine**.

Наприклад:

```
St = "X = " & x & Chr(10) & "Y = " & y & vbCrLf & "Z ="
```

Тут **Chr(10)** - переведення на інший рядок; замість **Chr(10)** можна використовувати константи **vbCrLf** або **vbNewLine**.

## БЛОК 4. ЗАВДАННЯ 12 ПОРІВНЯННЯ ЧИСЕЛ З РУХОМОЮ (ПЛАВАЮЧОЮ) КРАПКОЮ

Маємо п'ять змінних типу **Single**:

```
t! = 3:
```

```
x! = 1/t: y! = 4/t:
```

```
U! = 5 / t: V! = x + y
```

Тут знак "!" означає тип **Single** відповідних змінних.

Чи рівні змінні **U** і **V** між собою? (Це перша програма)? Також написати другу програму, яка порівнює їх різницю з дуже невеликим числом і робить висновок чи рівні змінні **U** і **V** між собою.

### ПРИМІТКИ

#### ПРИМІТКА 1. ТАЙМЕР З ФОРМОЮ

```
Option Explicit
```

```
' Запустимо VB, оберемо Standart.EXE
```

```
' На форму додамо мітку та таймер
```

```
Dim i As Integer
```

```
Private Sub Form_Load ()
```

```
    Me.Caption = " Зіборов "
```

```
    Command1.Caption = " Вихід "
```

```
    i = 0
```

```
    Label1.Caption = i
```

```
    Label1.FontSize = 16
```

```
    ' Задаємо інтервал таймера = 1 сек ,
```

```
    ' тобто 1000 мілісек :
```

```
    Timer1.Interval = 1000
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    ' Подія задзвенів таймер :
```

```
    ' Він дзвенить через кожну сік :
```

```
    i = i + 1
```

```
    Label1.Caption = i
```

```
End Sub
```

```
Private Sub Command1_Click()  
Unload Me ' або Unload Form1  
End Sub
```

#### ПРИМІТКА 2

Щоб відкрити VBA в Word 2016 спочатку потрібно налаштувати стрічку.

Тобто зробити так, щоб серед пунктів меню Файл

Вставка і т.д. з'явився пункт Розробник.

Для цього потрібно: Файл – Параметри – Налаштування стрічки.

Тут у таблиці Основні вкладки слід встановити прапорець поблизу Розробник — ОК.

Після цього серед пунктів меню Файл, Головна ,

Вставка і т.д. з'явиться пункт Розробник.

З цього пункту можна запускати VBA.

#### ПРИМІТКА 3

Якщо заблокувався VBA, слід закрити всі вордівські документи. Далі

відкрити файл \*.dost . Файл — Установки — Центр керування безпекою

— Установки центру керування безпекою — Увімкнути.

Преобразование Double: CDBl()

Преобразование **Cint**

**C**

## СПИСОК ЛІТЕРАТУРИ

1. І. М. Дудзяний Програмування мовою Visual Basic/VBA: навч. посіб. – Львів : ЛНУ, 2019. – 240 с.

[https://ami.lnu.edu.ua/wp-content/uploads/2017/05/VB\\_VBA.pdf](https://ami.lnu.edu.ua/wp-content/uploads/2017/05/VB_VBA.pdf)

2. В.С. Меркулов, І.Г. Бізюк, О.В. Чаленко Програмування в середовищі Visual Basic 6.0: конспект лекцій. – Харків : УДАЗТ, 2012. – 80 с.

<http://lib.kart.edu.ua/bitstream/123456789/6106/1/Конспект%20лекцій.pdf>

3. VBA Tutorial

<https://www.tutorialspoint.com/vba/index.htm>

4. Getting started with VBA in Office

<https://learn.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

Навчально-методичне видання

# ПРОГРАМУВАННЯ МОВОЮ VISUAL BASIC

Методичні вказівки  
до виконання лабораторних робіт  
для здобувачів першого (бакалаврського)  
рівня вищої освіти спеціальності  
193 «Геодезія та землеустрій»

Укладач ЗІБОРОВ Віктор Володимирович

Комп'ютерне верстання *А. П. Селівестрової*

Ум. друк. арк. 1,86. Обл.-вид. арк. 2,0  
Електронний документ. Вид № 3/V-25.

Виконавець і виготовлювач  
Київський національний університет будівництва і архітектури

Проспект Повітряних Сил, 31, Київ, Україна, 03680

Свідоцтво про внесення до Державного реєстру суб'єктів  
видавничої справи ДК № 808 від 13.02.2002 р