



Національний університет

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ
УКРАЇНИ**

Національний університет водного господарства та
природокористування

І. М. Карнович

В. О. Савич

Ю. О. Шенетько



ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ VISUAL BASIC. ПРАКТИКУМ

Навчальний посібник

Європейська кредитно-трансферна система

Для студентів напрямів підготовки

6.030504 “Економіка підприємства”

6.030505 ”Управління персоналом та економіка праці”

6.030508 “Фінанси і кредит”

6.030509 “Облік і аудит”

6.030601 ”Менеджмент”

Рівне – 2011



Національний університет

УДК 004(075.8)

ББК 32.973Я73

К 26

*Затверджено вченою радою Національного університету водного господарства та природокористування.
(Протокол № 9 від 30 вересня 2011 р.)*

Рецензенти:

Бомба А. Я., доктор технічних наук, професор Рівненського державного гуманітарного університету;

Кузнецова Т. В., кандидат економічних наук, доцент Національного університету водного господарства та природокористування, м. Рівне.

Карпович І. М., Савич В. О., Шепетько Ю. О.

К 26. Основи програмування мовою Visual Basic. Практикум. Навчальний посібник. – Рівне: НУВГП, 2011. – 117 с.

Розглянуто основні напрямки алгоритмізації і програмування економічних задач з використанням методів візуального проектування, принципи системного підходу до опрацювання та аналізу економічної інформації на ПК.

Для студентів, які здобувають освіту в галузях знань “Економіка та підприємництво” та ”Менеджмент і адміністрування”, а також спеціалістів-практиків, які займаються проблемами автоматизації та інформатизації діяльності підприємств.

УДК 004(075.8)

ББК 32.973Я73

© Карпович І.М., Савич В.О., Шепетько Ю.О., 2011
© НУВГП, 2011



Передмова.....	4
Лабораторна робота № 1. Знайомство з середовищем розробки проектів Visual Basic	5
Лабораторна робота № 2. Розробка проектів, що реалізують лінійні обчислювальні процеси	21
Лабораторна робота № 3. Реалізація проектів, що містять обчислювальні процеси з розгалуженнями	33
Лабораторна робота №4. Розробка проектів, що реалізують циклічні обчислювальні процеси	58
Лабораторна робота №5. Розв’язування задач економічного змісту з використанням одновимірних масивів	82
Лабораторна робота №6. Вкладені цикли. Використання двовимірних масивів при розв’язуванні задач економічного характеру	95
Тестові завдання.....	104
Література.....	114



Навчальний посібник “Основи програмування мовою Visual Basic. Практикум” призначено для базової підготовки бакалаврів напрямів “Облік і аудит”, “Менеджмент”, “Фінанси і кредит”, “Економіка підприємства”, “Управління персоналом та економіка праці” відповідно до вимог державного стандарту освіти. Його завдання – формування основ інформаційних знань та практичних навичок, необхідних для засвоєння нових програмних засобів та пакетів прикладних програм при вивченні фахових дисциплін.

Розвиток ринкових засад господарювання в умовах трансформації економіки вимагає нових підходів до формування системи управління. Важливого значення набуває використання комп’ютерних технологій, яке дає змогу значно підвищити якість оцінки діяльності підприємства, обґрунтованості управлінських рішень та контролю за їх виконанням.

Навчальний посібник, на думку авторів, сприятиме розвитку у студентів системного мислення, формуватиме у майбутніх фахівців вміння використовувати методику комплексного аналітичного дослідження з використанням інформаційних засобів для відшукування оптимальних шляхів вирішення виробничих проблем.

Комплекс лабораторних робіт, викладений у посібнику, пропонує методологію формалізації та алгоритмізації економічних задач, а також технологію розробки прикладних систем для їх розв’язання засобами Visual Basic, яка проілюстрована великою кількістю прикладів. Інформаційна частина посібника дає можливість отримати загальне уявлення про інтерфейс та основні елементи системи візуального проектування.

Практикум передбачає систематичну роботу кожного студента за комп’ютером, а також теоретичну підготовку студента до виконання лабораторної роботи. Для самостійної роботи, закріплення та практичної апробації отриманих знань в кінці кожного розділу вміщено контрольні запитання. Поглибленому вивченню матеріалу сприятимуть літературні та електронні джерела, перелік яких наведено наприкінці посібника.

Навчальний посібник – результат спільної праці з рівною часткою творчого внеску кожного з авторів.



Лабораторна робота № 1. Знайомство з середовищем розробки проектів Visual Basic

Мета: Набути уміння і навички роботи з компонентами середовища розробки проектів Visual Basic (VB) та створення форм за зразком.

Робота в середовищі Visual Basic

Найпростішою та найпоширенішою мовою програмування в усі роки була алгоритмічна мова Basic (Beginner's All-purpose Symbolic Instruction Code), створена ще в далекому 1963 р. з орієнтацією на початківців у галузі програмування. Простота мови забезпечувала можливість писати комп'ютерні програми для своїх прикладних застосувань математикам, економістам, інженерам та ін.

З плином часу мови програмування розвивались та вдосконалювались. Значних змін зазнали мови і технології програмування із появою операційної системи Windows та стрімким розвитком її популярності. Справа у тому, що написати прикладний додаток до Windows звичайними засобами було практично неможливо.

Лише з появою *візуальних засобів проектування* було повністю задоволено вимоги користувачів щодо створення прикладних додатків для Windows. Ці засоби дають змогу, образно кажучи, не створювати вікна, кнопки чи інші елементи інтерфейсу самому, а зробити це зручним, передбаченим у Windows, способом – мишею, як, наприклад, у графічному редакторі. Після створення такої конструкції (каркасу) залишається лише запрограмувати *події*, які з цими об'єктами відбуваються: наприклад, натискання кнопки миші або клавіатури, подвійне натискання кнопки миші тощо.

Незважаючи на значне спрощення процесу, розробка додатків для Windows потребує серйозного підходу, адже різні програмні продукти мають свої особливості формування принципів обробки подій, правил написання програмного коду і т.д.

Серед сучасних мов програмування високого рівня, що використовуються в ПК, широкого визнання набула мова Visual Basic. Слово Visual означає, що за допомогою цієї мови реалізовано



візуальний стиль програмування, де програми не пишуть, а проєктують.

Технологія роботи у середовищі Visual Basic базується на ідеях об'єктно-орієнтованого та візуального програмування. Ідея об'єктно-орієнтованого програмування полягає в об'єднанні даних і засобів їх опрацювання у тип, який називається об'єктом. Прикладами об'єктів можуть бути елементи керування у вікні: кнопки, списки, текстові поля тощо. Середовище візуального програмування Visual Basic – це графічна автоматизована оболонка над об'єктно-орієнтованою версією мови Basic. Якщо у мові Basic структурними одиницями є дані та команди, то тут такою структурною одиницею є візуальний об'єкт, який називається компонентом. Автоматизація програмування досягається завдяки можливості переносити компонент на форму (у програму) з палітри компонентів і змінювати його властивості, не вносячи змін до програмного коду вручну.

Формою називають компонент, який має властивості вікна Windows і призначений для розташування інших компонентів. Компоненти служать для організації діалогу з іншим користувачем. Вони відображаються на екрані під час виконання програми.

Проєктом називають сукупність файлів, що використовуються для розробки додатків, які створюються в середовищі Visual Basic.

Запустити Visual Basic можна передбаченим у Windows стандартним способом: запуском піктограми, вибором зі стартового меню та ін. При запуску VB з'являється стартове діалогове вікно **New Project**. Це вікно містить три закладки:

New (Новий) – шаблони і майстер для створення нового проєкту;

Existing (Існуючий) – відкриття існуючого проєкту;

Recent (Недавно створений) – список проєктів, які відкривалися недавно.

Для створення нового проєкту використовується перша закладка. Для створення стандартного додатку слід вибрати шаблон проєкту **Standart EXE** і клацнути на кнопці **“Открыть”**. Після цього відкриється головне вікно середовища VB із засобами проєктування (рис. 1.1). Як бачимо, все, з чим доводиться працювати користувачу, є характерним для звичайних Windows–програм.

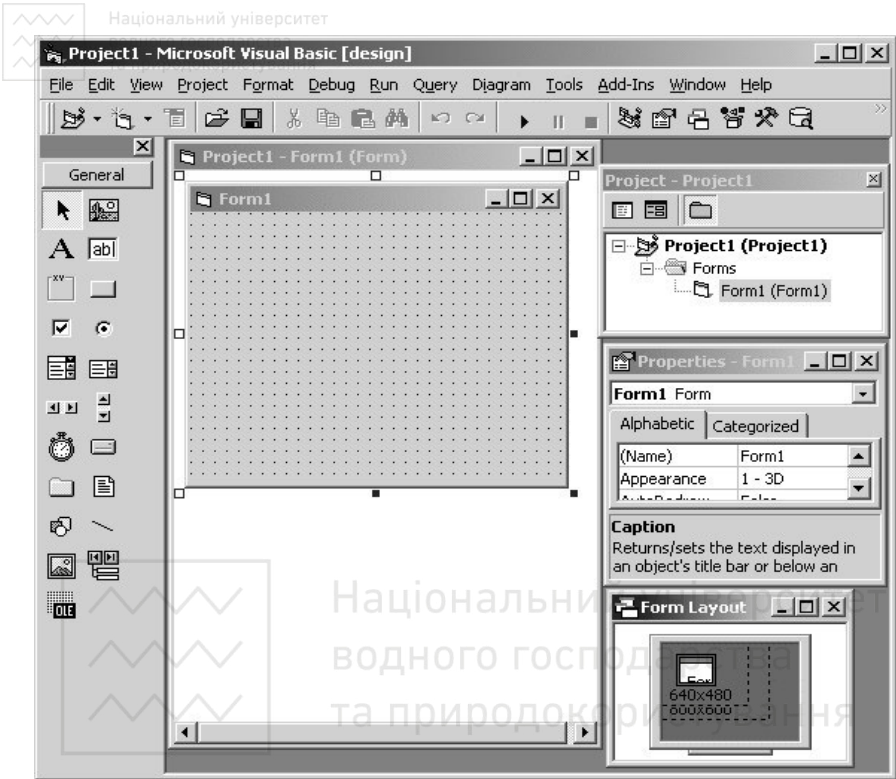


Рис. 1.1. Головне вікно середовища VB

До складу середовища проєктування входять такі основні елементи: головне меню, стандартна панель інструментів, панель елементів керування і допоміжні вікна. У рядку заголовка розміщено назву проєкту (рис.1.2). Якщо починаємо працювати “з чистого аркуша”, цією назвою буде **Project1**. Крім того, у квадратних дужках вказується, в якому режимі ми зараз перебуваємо – у режимі проєктування [*design*] чи в режимі роботи проєкту [*run*].



Рис. 1.2. Рядок заголовка головного вікна проєкту



Під рядком заголовка розмішені два рядки – рядок **Головне меню** (чи просто **Меню**) і рядок **Панель інструментів** (рис.1.3).

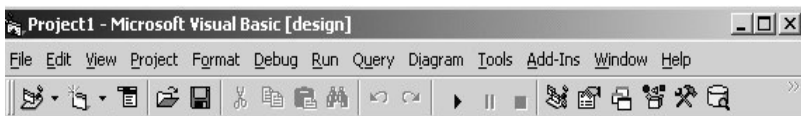


Рис. 1.3. Рядок заголовка, меню і панель інструментів головного вікна проекту

Панель інструментів можна легко перетворити у **Вікно інструментів** (рис.1.4) – для цього достатньо двічі клацнути мишею по правій частині **Панелі**.

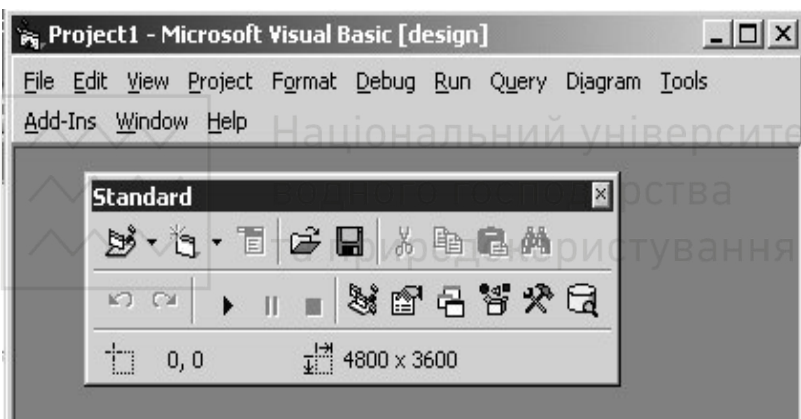




Рис. 1.4. Вікно інструментів, отримане з панелі інструментів

Під меню і панеллю інструментів у **Головному вікні** проекту розміщується **велике** (“**материнське**”) **вікно**, що може набувати різного вигляду. На рис.1.1 вид цього вікна майже збігається з тим, що ми бачимо під час першого відвідування середовища проектування Visual Basic – це так звана вхідна конфігурація середовища.

Велику частину площі **Головного вікна** проекту займає **Вікно екранної форми**, що ніби “зрослося” з **Головним вікном**. Три кнопки управління цим вікном  розмішені у верхній частині **Головного вікна** праворуч від **Меню**.



Клацнувши другу з цих кнопок  – *Restore* (відновлення), можна відокремити це вікно від *Головного вікна* проекту, зробити його “самостійно існуючим” вікном. “Самостійно існуючими” можна зробити й інші вікна середовища проектування, що пристиковані до *Головного вікна* (рис. 1.5). Для переміщення досить “схопити” мишею рядок заголовка кожного з цих вікон і перетягнути його на середину *Головного вікна*.

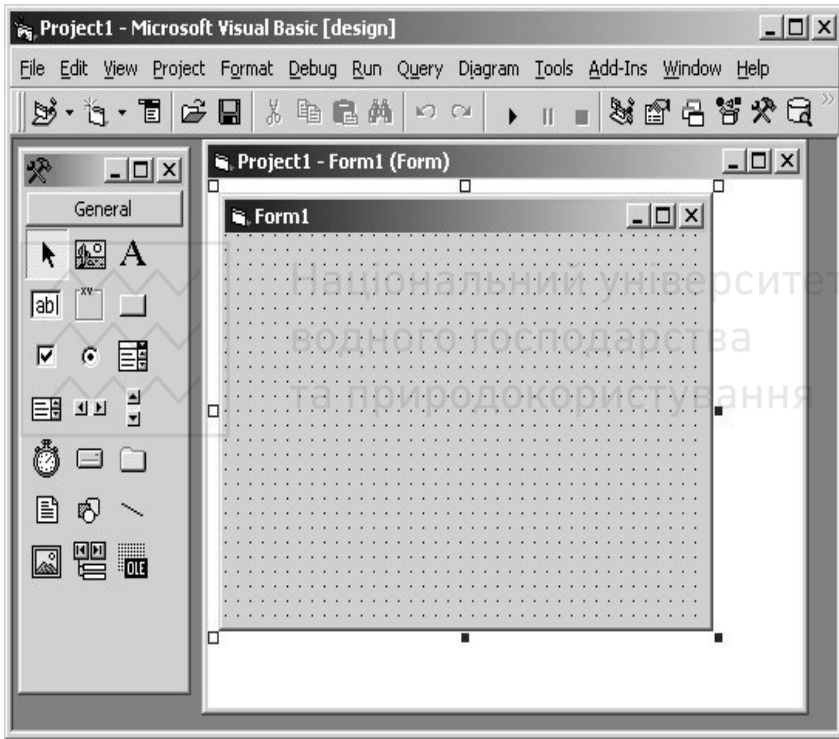


Рис. 1.5. Перетворення вікна екранної форми в “самостійно існуюче” вікно

Закріплення вікон (англ. *docking*) – це новий стандарт фірми Microsoft, що дозволяє поєднувати кілька вікон в одному вікні. На рис.1.1 вікно *Toolbox* (заголовок *General*) закріплено до лівого краю *Головного вікна*, вікно *Properties* закріплено до нижнього краю вікна *Project*, а вікно *Form Layout* – до нижнього краю вікна

Properties. Всі три вікна утворюють єдине вікно та пристиковані до правого краю **Головного вікна**.

На рис.1.6 показано **Головне вікно** проекту з незакріпленими інструментальними вікнами.

Можна провести і зворотну операцію – пристикувати (прикріпити) вікно або до **Головного вікна**, або до іншого вікна. Для цього потрібно схопити мишею за рядок заголовка цього вікна, перемістити його до краю **Головного вікна** чи на межу між двома закріпленими вікнами.

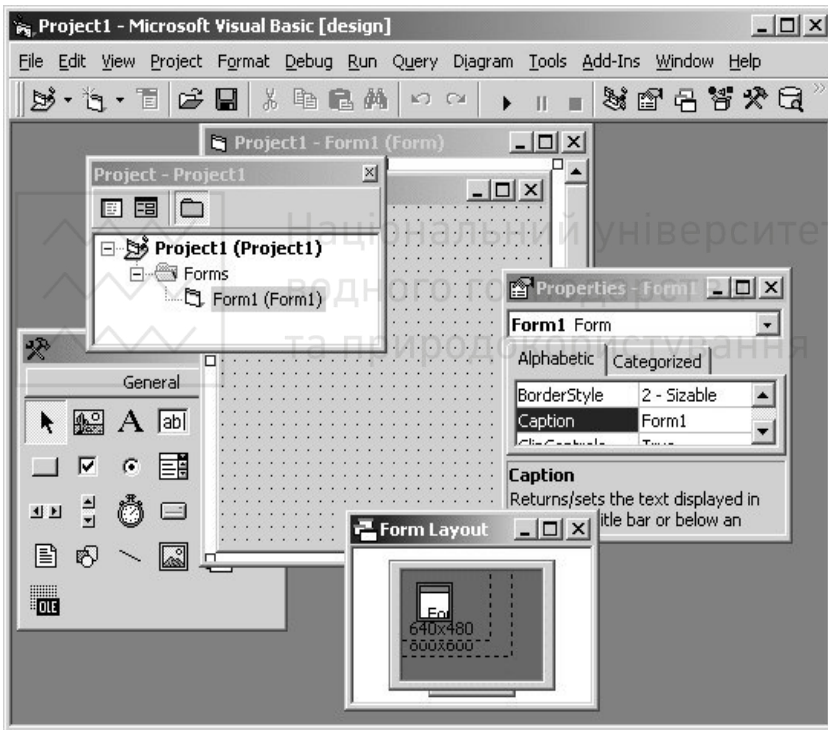


Рис. 1.6. Незакріплені інструментальні вікна головного вікна проекту

Розглянемо більш детально склад і призначення елементів **Меню** і **Панелі інструментів**. Як і більшість вікон систем, розроблених фірмою Microsoft, рядок **Меню** складається зі слів, кожне з яких позначає опцію (чи пункт) цього **Меню** (рис.1.7).



Рис. 1.7. Опції Меню головного вікна проекту Visual Basic

Багато опцій працюють так само, як і в інших додатках фірми Microsoft. Підкреслена буква в назві кожної опції дозволяє звернутися до цієї опції, якщо не працює миша і ми змушені користуватися клавіатурою (використовується комбінація клавіш **Alt** + <підкреслена буква >).

Меню **File** (Файл) містить команди, що є типовими для цієї опції в інших додатках, наприклад: **Print** (друк), **Exit** (завершення роботи). Але є серед них і такі, яких немає в інших додатках: створення нового проекту, відкриття (завантаження) вже існуючого проекту, збереження поточного проекту.

Меню **Edit** (Редагування) містить досить відомі команди редагування, зокрема: вилучення, копіювання, вставки, пошуку, відновлення, скасування.

Меню **View** (Вигляд) дозволяє розкрити в Головному вікні проекту інструментальні вікна, необхідні для подальшої роботи.

Меню **Project** (Проект) містить команди, що дозволяють керувати проектом, наприклад, команда **Add** (додати) дозволяє внести в поточний проект новий елемент (форму, файл, модуль, елемент управління тощо).

Меню **Format** (Формат) дозволяє формувати об'єкти, розміщені на екранній формі, наприклад, змінювати їхні розміри, вирівнювати розміри виділеної групи об'єктів і т.д.

Меню **Debug** (Налагодження) містить команди, призначені для налагодження програми.

Меню **Run** (Виконати) містить команди, за допомогою яких можна керувати режимом виконання програми (запустити програму, перервати її роботу, завершити виконання програми).


Меню **Tools** (Інструменти) містить команди створення нових процедур і функцій, відкриття Вікна редактора меню, зміни параметрів настроювання середовища проектування тощо.

Меню **Add – Ins** (Додавання утиліт) дозволяє розширювати можливості середовища проектування включенням у неї додаткових можливостей.

Меню **Window** (Вікно) дозволяє змінювати розташування вікон у Головному вікні, наприклад: горизонтально, вертикально чи каскадом.

Меню **Help** (Довідка) дозволяє звертатися до довідкової системи Visual Basic.

Елементи керування додатком під час проектування форми вибираються з об'єктів панелі керування.

Панель елементів керування вмикається кнопкою  (**Tools**) і містить елементи керування, що використовуються при побудові інтерфейсу користувача. Найчастіше використовуються наступні елементи:



– Pointer (Показчик) – позиціювання курсора миші;



– Label (Напис) – формування текстових об'єктів-написів, редагувати які під час виконання додатку неможливо;



– TextBox (Текстове поле) – розміщення полів для введення і виведення даних (текстової інформації, чисел, дат);




– CommandButton (Командна кнопка) – розміщення у формі кнопок управління для ініціювання дій, виконання команд і т. д.

Об'єкти керування на формі можна створювати наступним чином.




1-й спосіб. Вибрати потрібний елемент на панелі елементів керування і клацнути на ньому. У межах форми курсор миші набуде вигляду хрестика. За допомогою миші визначаються межі об'єкта і відпускається ліва клавіша миші.

2-й спосіб. Двічі клацнути на значку об'єкта на панелі об'єктів керування. Об'єкт з'явиться у формі, причому місце та розміри об'єкта визначаються за замовчуванням.

У процесі роботи над проектом він часто розростається, здебільшого, за рахунок форм. Досить зручний спосіб навігації по проекту дає вікно провідника проекту.


Вікно **Провідник проекту** вмикається кнопкою  (Project Explorer) панелі інструментів і призначене для відображення структури проекту, який розробляється, і перемикання його

окремих компонентів. Він має кнопки для активізації вікна коду

( код) або вікна форми ( форма) та зміни режиму перегляду().

Зауважимо, що вміння розуміти властивості об'єктів надзвичайно важливе при створенні додатків засобами Visual Basic. Адже кількість властивостей лише самої форми сягає понад п'ятдесят.

Вікно властивостей призначене для відображення і налаштування властивостей форми, а також розміщених у ній об'єктів. Набір властивостей у цьому вікні змінюється залежно від

обраного об'єкта. Вмикається вікно властивостей кнопкою панелі інструментів (**Properties**) .

За допомогою вкладок **Alphabetic** (За алфавітом) і **Categorized** (За категоріями) властивості об'єкта можна переглянути за абеткою чи / групами / (категоріями) відповідно. Списки властивостей складаються з двох стовпців: у першому з них вказано назву властивості, у другому – її значення. У нижній частині вікна є підказка, що пояснює призначення обраного атрибута об'єкта.

У вікні **конструктора форм** візуально конструюються елементи додатка. Для точного розміщення об'єктів у формі слугує сітка. При виконанні додатка вона не відображається. Розмір форми у вікні можна змінювати, використовуючи маркери виділення форми і мишу.

Конструювання форми здійснюється елементами керування. Для внесення елемента керування у форму потрібно клацнути на панелі елементів керування його значок і визначити мишею місце розміщення обраного елемента та у вікні властивостей задати параметри елемента. Для вилучення елемента керування потрібно виділити його і натиснути клавішу **<Delete>**.

Властивості об'єктів допускається змінювати при виконанні додатка, написавши відповідні коди в процедурах, що створюються за допомогою редактора коду.

З елементом керування, як правило, пов'язуються певні події. Для того, щоб додаток працював як слід, потрібно запрограмувати процедури обробки певних подій. Це здійснюється у вікні редактора



коду. **Редактор коду** – це вбудований у Visual Basic редактор із зручними засобами введення коду програми.




Для створення процедури опрацювання події, пов'язаної з елементом керування, досить двічі клацнути мишкою на ньому. При цьому з'являється вікно коду, у якому є заголовок і закінчення майбутньої процедури.

У заголовку процедури вказується ім'я відповідного елемента керування і події, що відбувається при роботі з цим елементом. Коли потрібно опрацювати іншу подію, її назву вибирають з правого поля із списку, який розташований під смугою заголовка вікна коду. Якщо потрібно написати процедуру для іншого елемента керування, його ім'я вибирається з лівого списку.

При **збереженні проекту** слід пам'ятати, що він складається із вхідних компонент, таких як форми, модулі коду тощо. Тому процес збереження проекту вимагає послідовного збереження усіх його компонент. Для цього в меню **File** слід вибрати команду **Save Project** чи натиснути кнопку **Save Project** на стандартній панелі інструментів. Кожен проект рекомендується зберігати в окремій папці. Після відкриття діалогового вікна **Save File as** список **Тип файла** міститиме значення **Form Files**, вказуючи, що зберігається форма, яка входить у проект. В цьому випадку після збереження форми з'явиться діалогове вікно **Save Project as** для збереження самого проекту. За замовчуванням у списку **Папка** обирається папка, в якій зберегли форму.

При наступному збереженні проекту Visual Basic за замовчуванням зберігає файли з їх іменами.

Виконання проекту регулюється кнопками панелі інструментів

 – запуск (Start),  – зупинка (Break),  – завершення (End) або відповідними командами меню.

Щоб надати розробленому проекту статусу незалежного додатка, потрібно зберегти цей проект як виконавчий (скомпільований) файл з розширенням **exe**. Для цього в меню **File** потрібно вибрати команду **Make Project1.exe**, де Project1 – ім'я активного проекту, і вказати місце збереження та ім'я файла.

Завершення роботи у середовищі Visual Basic здійснюється або виходом з головного вікна за допомогою пункту меню **File – Exit**, або комбінацією клавіш **Alt+Q**.

Завдання для виконання лабораторної роботи № 1

- I. Завантажити середовище Visual Basic для створення нового проекту.
- II. Вилучити панель елементів керування, а потім її знову відновити.
- III. Вилучити, а потім відновити вікна провідника проекту і властивостей.
- IV. Визначити значення таких властивостей форми, як *Name*, *Caption*.
- V. Подвійним клацанням миші на формі пред'явити вікно коду.
- VI. За допомогою вікна провідника проекту перейти з вікна коду у вікно форми.
- VII. Створити форму із заголовком “Лінійні алгоритми” для обчислення виразів

$$x = \frac{\cos a + e^b}{2(a+1)}, \quad y = \frac{a-b}{e}, \quad z = (x+2)^2 + y^2,$$

написати код для розрахунку значення цих виразів, розташувавши поля у формі згідно зразка №1 (рис. 1.8) (використати поля редагування **TextBox**, поля з текстовими написами **Label**, командну кнопку **CommandButton** та задати їх основні властивості: підпис **Caption**, шрифт **Font**, текст **Text**).

- VIII. Формі зразка №1 (рис.1.8) надати вигляду форми зразка №2 (рис.1.9), змінивши підписи *Caption* на об'єктах типу *Label* і *CommandButton* та очистивши поле редагування (значення *Text* замінити на порожньо). Приклад побудови алгоритму розв'язування задач наведено на рис. 1.10.



Form1

Labell

Label2	Text1
Label3	Text2
Label4	Command1
Label5	Text3
Label6	Text4
Label7	Text5

Рис. 1.8. Зразок №1

Лінійні алгоритми

Вхідні дані

a=

b=

Результати

Обчислити

x=

y=

z=

Рис. 1.9. Зразок №2

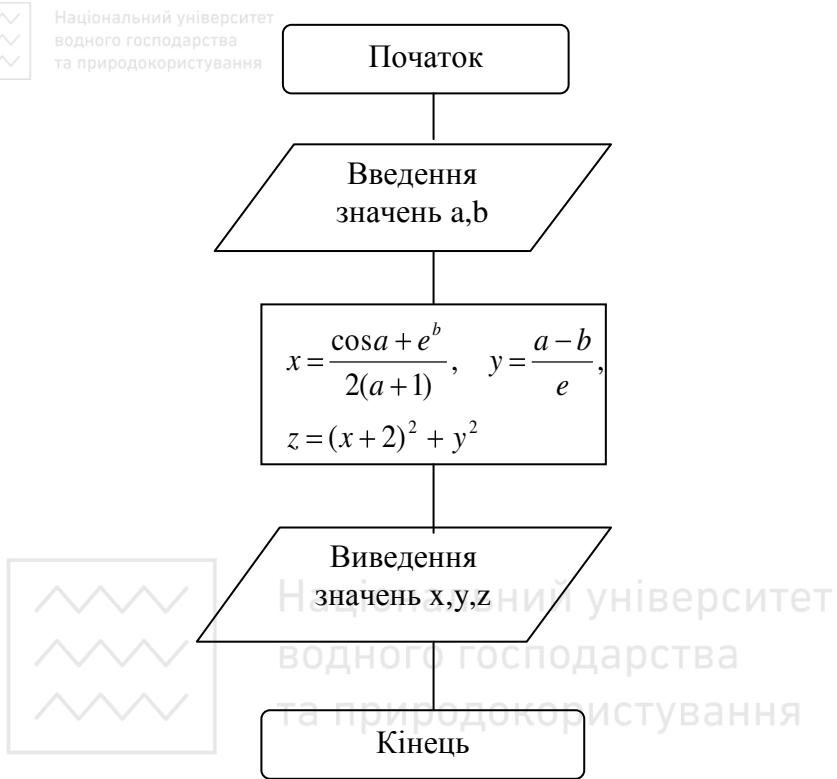


Рис. 1.10. Алгоритм розв'язування задачі

IX. Запрограмувати кнопку “**Обчислити**” згідно алгоритму.
Приклад побудови коду для обчислення виразів:

```
Private Sub Command1_Click()  
a = Val(Text1.Text)  
b = Val(Text2.Text)  
x = (Cos(a) + Exp(b)) / 2 / (a + 1)  
y = (a - b) / Exp(1)  
z = (x + 2) ^ 2 + y ^ 2  
Text3.Text = Str(x)  
Text4.Text = Str(y)  
Text5.Text = Str(z)  
End Sub
```



- X. Зберегти проект у папці Lab1_z1.
- XI. Виконати проект.
- XII. Продемонструвати створений проект викладачеві.
Записати у зошит отримані результати.
- XIII. Розробити форму для обчислення суми виплат S за формулою:

$$S = V \left(1 + \frac{P}{100} \right)^T,$$

Для розробки форми потрібно врахувати математичну модель, яка передбачає задання початкового вкладу V , терміну вкладу T і річної відсоткової ставки P та подальшого обчислення суми виплат S . Для розв'язування задачі можна використати зразки розробки форми, подані на рис. 1.11, 1.12.

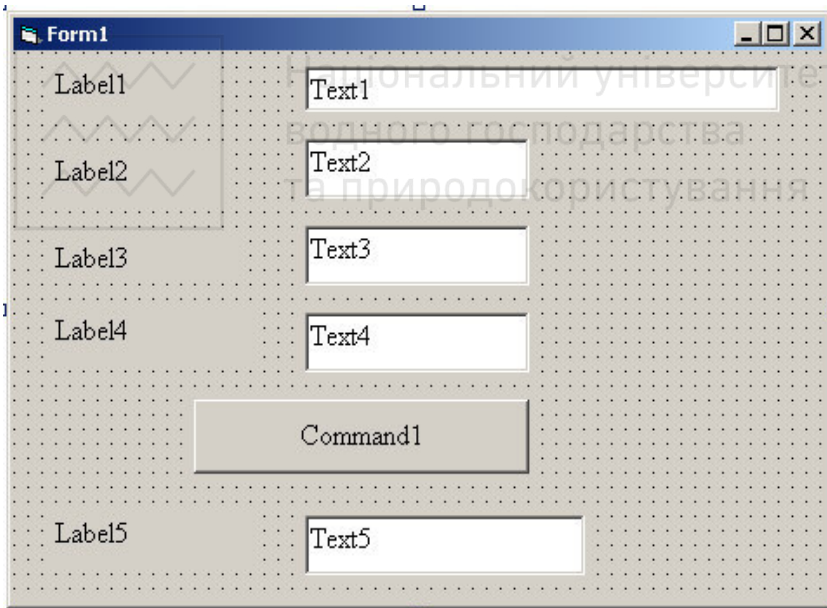


Рис. 1.11. Зразок №3

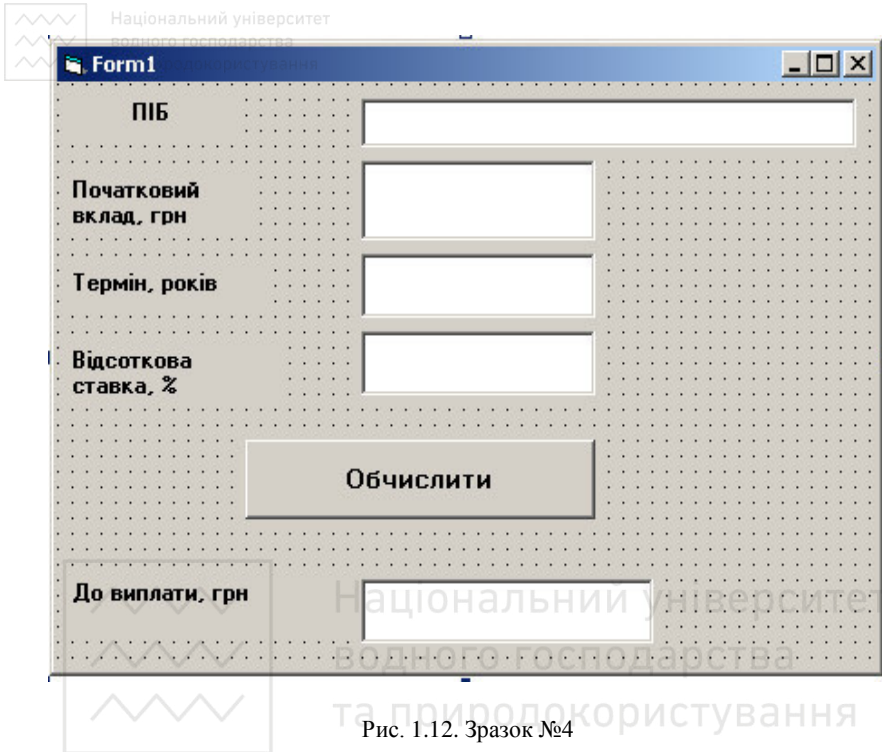


Рис. 1.12. Зразок №4

Розглянемо один з варіантів коду для розв'язування задачі:

```
Private Sub Command1_Click()
Dim V, S As Currency
V = Val(Text2.Text)
T = Val(Text3.Text)
P = Val(Text4.Text)
S = V * (1 + P / 100) ^ T
Text5.Text = Str(S)
End Sub
```

- XIV. Зберегти проект у папці Lab1_z2.
- XV. Виконати проект. Продемонструвати створений проект викладачеві. Записати у зошит отримані результати.
- XVI. Захистити виконану роботу.



Контрольні запитання

1. Які основні елементи середовища проектування Visual Basic?
2. Назвіть елементи управління додатком та їх функції.
3. Назвіть основні властивості елементів управління додатком.
4. Яке призначення форми у VB?
5. Які з об'єктів, розміщених на екранній формі, можуть бути обов'язковими, а які необов'язковими (надлишковими)?
6. Для чого потрібна сітка з крапок на екранній формі?
7. Як вмикається панель елементів керування?
8. Охарактеризуйте вікно властивостей.
9. Чим відрізняється зміст властивостей *Left* і *Top* екранної форми?
10. Які два способи зміни *розмірів* і *положення* об'єкта на екранній формі можна використовувати для їх виправлення (коригування)?
11. Яке призначення вікна властивостей?
12. Охарактеризуйте способи збереження проекту, форми.





Лабораторна робота № 2. Розробка проектів, що реалізують лінійні обчислювальні процеси

Мета: Навчитись використовувати засоби VB для розробки проектів, що реалізують лінійні обчислювальні процеси.

Поняття змінної. Типи змінних. Константи

Змінною називають зарезервовану область оперативної пам'яті, призначену для зберігання даних. Кожна змінна має *ім'я*.

Ім'я змінної – це рядок символів, що ідентифікує змінну в програмі (англ. *Identify* – розпізнавати, встановлювати ідентичність). Імена змінних у програмі часто називають ідентифікаторами змінних, що передбачає унікальність імені в програмі. Вибирати імена змінних слід так, щоб вони містили в собі інформацію про призначення змінної і не були дуже довгими. У Visual Basic існує декілька правил задання імен змінних:

- першим символом імені повинна бути велика або мала *буква*;
- в імені змінної можна використовувати знак “_”, але заборонено використовувати крапку та пропуски;
- кількість символів в імені змінної не повинна перевищувати 255;
- ім'я змінної не повинно збігатися з *ключовим словом* Visual Basic і має бути унікальним у межах зони видимості.

Значення змінної – це *дані*, які зберігаються і опрацьовуються системою Visual Basic. Спосіб збереження і опрацювання даних відбувається із врахуванням їх типу.

Типом даних називається спосіб збереження і подання даних у системі Visual Basic, який задає певний формат або розмір вмісту змінної.

Тип змінної визначається типом значення змінної. Основні типи даних, які підтримує VB, наведені в таблиці 2.1.



Основні типи даних Visual Basic

№ з/п	Тип даних	Опис і діапазон даних
1	Boolean	Тип даних, що набуває одного з двох значень: False або True
2	Byte	Цілі значення від 0 до 255
3	Currency	Дані з фіксованою крапкою від -9,22337E+14 до 9,22337E+14
4	Date	Значення дати і часу від 01.01.100 до 31.12.9999
5	Decimal	Числа з 28 десятковими розрядами точності
6	Double	Числові значення в діапазоні від -1.7E+308 до 1.7E+308
7	Integer	Цілі числові значення від -32768 до 32767
8	Long	Цілі числові значення від -2147483648 до 2147483648
9	Object	Тип даних, що містить посилання на об'єкти
10	Single	Числові значення в діапазоні від -3.402823E+38 до 3.402823E+38
11	String	Дані, що містять від 0 до 1032 алфавітно-цифрових символів
12	Variant	Дані будь-якого типу, що встановлюються для елементів, у яких тип даних невідомий

Оголошення типу змінної встановлює для неї межі значень, які визначаються цим типом. Під час складання програми мовою Visual Basic користувач сам вирішує, треба оголошувати змінну чи ні. Відзначимо, що тип змінної можна взагалі не оголошувати. У цьому випадку тип змінної визначається при першому операторі присвоєння. В цей момент для розміщення змінної буде виділений необхідний обсяг пам'яті. Якщо не використовуються спеціальні кінцеві символи, що визначають тип змінної, їй буде присвоєний тип **Variant**.



Кінцеві символи або суфікси в імені змінної вказують на її тип. Види суфіксів та деякі приклади їх застосування наведено в таблиці 2.2.

Таблиця 2.2

Приклади застосування суфіксів в імені змінної

Тип	Суфікс	Приклади застосування
Integer	%	<i>ЧислоВоронНаГілиціДуба%</i>
Long	&	<i>Колір даху будинку&</i>
Single	!	<i>X238!</i>
Double	#	<i>Пі#</i>
String	\$	<i>My_name\$</i>
Currency	@	<i>price@</i>

Якщо тип змінної не визначений, то це може призвести до зниження ефективності роботи програми, нерационального використання пам'яті, а іноді – до помилок у роботі додатка. Тому “гарним тоном” у програмуванні вважається **обов'язкове** оголошення типу кожної змінної. Для встановлення такого режиму трансляції програми, при якому допускається тільки явне оголошення змінних, необхідно в початок модуля вставити оператор **Option Explicit** (явне оголошення).

Явне оголошення змінних здійснюється операторами **Dim**, **Private**, **Static**, **Public**, що мають такий синтаксис:

Dim Ім'язмінної [As ТипДаних]

Private Ім'язмінної [As ТипДаних]

Static Ім'язмінної [As ТипДаних]

Public Ім'язмінної [As ТипДаних]

Тут ТипДаних – один з перерахованих вище типів (**Integer**, **Long** тощо). У квадратних дужках [] вказана синтаксично необов'язкова конструкція.

Оператори **Dim**, **Private**, **Static**, **Public** визначають зону дії змінної. За допомогою одного оператора можна оголошувати декілька змінних, розділяючи їх комами.

У випадку, коли змінна оголошується за допомогою ключового слова **Dim**, вона буде локальною змінною. Зоною її застосування



буде тільки та процедура, в якій змінна була оголошена. Після завершення роботи процедури значення змінної буде втрачене.

Якщо ж змінна оголошується з ключовим словом **Static**, вона теж буде локальною, проте її значення не втратиться після завершення процедури.

Якщо змінна оголошується з ключовим словом **Private**, вона буде видимою у декількох процедурах одного модуля. От тільки описувати її слід у розділі **Declarations**.

Якщо змінна повинна бути видимою усім процедурам усіх модулів, її оголошують із ключовим словом **Public**. Така змінна має назву глобальної.

Приклад 1. Оголошення змінних My_name, My_book.

```
Dim My_name As String  
Dim My_book
```

Приклад 2. В одному операторі **Dim** можна через кому записати кілька конструкцій “*Ім'яЗмінної As Тип Даних*”.

```
Dim X As Single, I As Integer, Pi As Double  
Dim Франція, Німеччина, Норвегія
```

Значенням змінної типу **String** може бути рядок *довільної* чи *фіксованої* довжини. При оголошенні змінної типу **String** фіксованої довжини після ключового слова **String** у рядку оператора через символ “*” вказується довжина рядка:

Dim Ім'яЗмінної As String* Довжина рядка

Приклад 3.

```
Dim НомерТелефону As String* 12
```

Значенням змінної *НомерТелефону* є текстовий рядок фіксованої довжини (12 символів), наприклад, (095)9004567. До рядка з меншою кількістю символів Visual Basic автоматично припише необхідну кількість пропусків праворуч.

Константи – це дані, значення яких не змінюються в процесі виконання програми. Як і змінні, константи оголошуються на початку тексту програмного коду. Для оголошення констант



використовується ключове слово **Const**. Одночасно з оголошенням констант їм присвоюються значення.

Оголошення константи має такий синтаксис:

Const <Ім'яКонстанти> [**As Тип**] = <ЗначенняКонстанти>

Visual Basic містить велику кількість вбудованих констант майже для всіх випадків, які зустрічаються у практиці програмування коду: константи кольору, повідомлень тощо. Усі вбудовані константи мають префікс **vb**. Наприклад, константи *кольору*: **vbRed** – значення червоного кольору, **vbWhite** – значення білого кольору і т.д.

Оператор присвоєння. Приклади застосування

Для надання змінній деякого значення використовується оператор присвоєння.

Оператор присвоєння – один з основних операторів будь-якої мови програмування. *Синтаксичне правило* запису оператора:

[**Let**] <Ім'яЗмінної> =< Вираз >

Конструкція <Вираз> праворуч від знака рівності – це або значення, що міститься в діапазоні даного типу змінної, або *вираз*, значення якого повинно бути обчислене.

При *виконанні* оператора присвоєння обчислюється значення виразу, що розміщений праворуч від знака рівності. Це значення присвоюється змінній, ім'я якої зазначено ліворуч від знака рівності. Типи змінної та значення виразу повинні бути сумісними.

Наступні два фрагменти програмних кодів демонструють роботу оператора присвоєння в середовищі програмування Visual Basic.

Приклад 4. Оператори присвоєння записані правильно і будуть виконані:

```
Dim Var1 As Integer, Var2 As Long
Dim Var3 As Single, Var4 As Double
Var1 = 32000
Var2 = -2000000000
Var3 = 9E-30
Var4 = Var3
```

Приклад 5. Перший з операторів присвоєння буде виконаний, а три наступних – ні, тому що використані значення виходять за межі допустимого діапазону:

```
Dim Var1 As Integer, Var2 As Long
Dim Var3 As Single, Var4 As Double
Var4 = 9E-300
Var1 = 33000
Var2 = -3000000000
Var3 = Var4
```

Арифметичні операції та вирази

Найчастіше в операторі присвоєння праворуч від знака “=” розміщений так званий арифметичний вираз.

Арифметичний вираз – це послідовність констант, змінних, функцій, з’єднаних знаками арифметичних операцій з можливим використанням круглих дужок. Змінні, що входять до виразів, повинні отримати числові значення до їх використання в програмі.

Значення арифметичних виразів *обчислюються* за правилами, що є загальновідомими, тому ми не будемо на них детально зупинятися. Перелік *арифметичних операцій*, що використовуються у мові Visual Basic, наведено в таблиці 2.3.

Таблиця 2.3

Арифметичні операції Visual Basic

Операція	Опис операції
A^B	Піднесення A до степені B
$-A$	Зміна знака A
$A * B$	Множення A на B
A/B	Ділення A на B
$A \setminus B$	Цілочисельне ділення A на B
$A \text{ Mod } B$	Залишок від цілочисельного ділення A на B
$A + B$	Додавання A та B
$A - B$	Віднімання B від A

Приклади використання математичних операцій:

- $(3 + 2)/(6 - 4)$; значення виразу дорівнює 2.5;
- $3 * (- 3)$; значення виразу дорівнює -9;
- 345; значення виразу дорівнює 6;
- $34 \bmod 5$ дорівнює 4;
- $2^2 + 1$; значення виразу дорівнює 5.

У таблиці 2.3 арифметичні операції перераховані в порядку *спадання* їхнього пріоритету, однак дві пари операцій мають *однаковий* пріоритет – це множення і ділення ($*$ і $/$) та додавання і віднімання ($+$ і $-$). Порядок виконання операцій у виразах можна змінити за допомогою круглих дужок.

Розглянемо фрагмент програми з операторами оголошення змінних і констант та декількома операторами присвоєння:

```
Dim r As Single, s As Single ' Оголошення змінних
Const pi = 3.14159 ' Оголошення константи –
' число "Пі"
Let r = 10 ' Присвоєння значення змінній r
s = pi * r ^ 2 ' Обчислення площі круга
r = 2 * r ' Присвоєння значення змінній r
```

У фрагменті використані **коментарі** для роз'яснення змісту програми. Коментар може бути як окремим рядком, так і міститись у рядку програмного коду після символу ' (апостроф).

Крім демонстрації обчислення арифметичних виразів, цей фрагмент вказує на те, що *одна* змінна на різних етапах роботи програми може набувати *різних* значень.

Як правило, при написанні програм кожен оператор розміщують в окремому рядку. У тих випадках, коли оператор має велику довжину, його можна розбити на декілька рядків, використовуючи символи продовження рядка: пропуск, за яким іде символ продовження ”_”. Якщо оператори мають невелику довжину, Visual Basic дозволяє їх розмістити в одному рядку, розділивши двокрапкою.

Вбудовані функції мови Visual Basic

Функція – це оператор, що виконує певні дії (наприклад, запит користувача на введення інформації або обчислення за формулою), а потім повертає результат своєї роботи у програму.

Стандартними, стосовно Visual Basic, називаються такі функції, обчислення яких є складовими компонентами VB. До них належать математичні, рядкові, фінансові та інші функції.

Наявність вбудованих функцій значно спрощує процес програмування, оскільки без створення підпрограм дозволяє обчислення багатьох функцій. До будь-якої із вбудованих функцій можна звернутись за ім'ям, передавши їй значення необхідних аргументів, і здобути результат – значення, що повертається. Найуживаніші вбудовані функції наведені в таблиці 2.4.

Таблиця 2.4

Основні вбудовані функції Visual Basic

Функція	Дія функції
Abs (x)	Повертає абсолютне значення x
Atn (x)	Повертає арктангенс x
Cos (x)	Повертає косинус кута x, заданого в радіанах
Exp (x)	Повертає експоненту x
Fix (x)	Повертає ціле число, рівне x без дробової частини
Int (x)	Повертає найбільше ціле число, що не перевищує x
Log (x)	Повертає натуральний логарифм x
Sin (x)	Повертає синус кута x, заданого в радіанах
Sgn (x)	Повертає -1, якщо $x < 0$, 0, якщо $x = 0$, 1, якщо $x > 0$
Sqr (x)	Повертає квадратний корінь з x
Rnd (x)	Генерує випадкове число між 0 та 1
Tan (x)	Повертає тангенс кута x, заданого в радіанах
Функції роботи з рядками	
Val (x)	Перетворює символний рядок x на числове значення
Str (x)	Перетворює числове значення x в символний рядок

Завдання для виконання лабораторної роботи № 2

I. Завантажити середовище Visual Basic для створення нового проекту.

II. Розробити форму із заголовком "Лінійні обчислювальні процеси", записати алгоритм та ввести програмний код для обчислення наступних виразів:

$$1. \quad y = \frac{x^2 - z^2}{\lg|x-7|}, \quad x = \frac{\sin^2 a^3 - \sin b}{\ln|a-b|+1}, \quad z = \sqrt{\left| \frac{a+b}{ab} \right|} + \pi,$$

$$a = 2.5, b = -2.1$$

$$2. \quad z = \left| \frac{x\sqrt{|x|} + 3}{\operatorname{arctg} y + 2.4} \right|, \quad y = |a^2 + ab + 0.5|, \quad x = \ln b + \frac{b - b^2}{b} e^{-a},$$

$$a = -0.5, b = 7$$

$$3. \quad y = |\ln|x| - \lg|z| + 0.3|, \quad x = \frac{e^{-2.5} + \sin^2 a^3}{2\lg|ab|}, \quad z = \frac{\sin^2(a-b) + \sqrt[3]{ab}}{1 + \log_a b},$$

$$a = 0.6, b = 3.12$$

$$4. \quad t = \ln|m-y| + \cos^3 m, \quad m = \sqrt[3]{|x+a|} + \lg \frac{\pi}{3}, \quad y = a^3 \sqrt{\sin^4 x^2} + 12,$$

$$x = 0.7, a = -4$$

$$5. \quad d = \operatorname{ctg} \frac{x+1}{y-2} + \lg|kx|, \quad x = \sqrt{|m+n|}^3 + mn, \quad y = \sqrt[3]{|kn-3|} + \frac{\pi}{6},$$

$$m = 3, n = -2, k = 0.8$$

$$6. \quad j = \log_{\pi}|x| + \left| \frac{\pi}{5} - y \right|, \quad x = \operatorname{arctg} \frac{5.4}{m} + mn, \quad y = \sqrt{|m-n|} + \ln n^2,$$

$$m = -2, n = 3.87$$



$$7. f = \frac{x^e - e^{-x} + 0.23}{\sqrt{|\sin \pi - 1|}}, \quad x = e^{-a} + \pi^{-e}, \quad y = \lg a^3 - \operatorname{arctg} a$$

$$a = 3.45$$

$$8. b = |c - z|^{-e} + \sqrt[3]{|z| + 1}, \quad c = e^{k-5.1} + \lg|k + x|, \quad z = \ln^3|2^x - 1|,$$

$$x = 0.03, k = 4$$

$$9. a = \sqrt{|\pi - y|} + \sin^2 \pi x, \quad y = \operatorname{tg}^4 \pi - 1, \quad x = \operatorname{ctg} \frac{d-1}{e} + 2 \frac{d+1}{2},$$

$$d = 4.4, b = 1.87$$

$$10. y = ax^{-3.1} + e^{a/x}, \quad x = \operatorname{tg} \frac{z}{a} + \operatorname{ctg} \sqrt{|z|}, \quad z = \sqrt[3]{\ln a + \ln a^2},$$

$$a = 2.77$$

$$11. t = \frac{x^2 - y^2}{e^{-x+y}}, \quad x = \sqrt{6 + e^y + |y|}, \quad y = \lg e + \sqrt[5]{|\cos e - 2|}$$

$$12. z = \frac{x^3 - y^5}{e^{-x+y}}, \quad x = \sqrt{5e^y + |y + 2|}, \quad y = \sqrt[3]{\lg e} + \sqrt[5]{|\cos e|}$$

$$13. x = \cos \frac{\pi - z}{3} + e, \quad z = \sqrt[3]{|y + \sin^2 y|}, \quad y = 0.3 \log_5 e^{-2.3}$$

$$14. x = \frac{\cos \frac{z - \pi}{3}}{e}, \quad z = \sqrt{|y| + e^y}, \quad y = \lg e + \cos^2 e$$

$$15. g = e^{|z|} + \lg z^4, \quad z = \sqrt[5]{|x| + 8.1}, \quad x = \lg e + \cos \frac{\pi + e}{2}$$



III. Зберегти проект у папці Lab2_z1. Продемонструвати створений проект викладачеві. Записати отримані результати.

IV. Розробити проект, форму та ввести програмний код для розв'язання наступних задач:

1. Обчислити вартість товару та вартість товару з врахуванням торгової надбавки, якщо задано ціну одиниці товару (грн.), кількість товару (шт.) та торгіву надбавку (%).
2. Обчислити залишок на банківському рахунку після проведення банківської операції, якщо відомий залишок на рахунку до операції, надходження та зняття коштів з рахунку.
3. Обчислити суму оплати виконаних будівельних робіт, якщо відомий відсоток виконання запланованого обсягу робіт та загальна вартість будівельного проекту.
4. Обчислити відсоток студентів, що склали іспити на "відмінно" та "добре", якщо відома загальна кількість студентів та кількість студентів, що склали іспити без задовільних оцінок.
5. Обчислити відсоток студентів, що не склали іспити, якщо відомо загальну кількість студентів; кількість студентів, що склали іспити без задовільних оцінок; кількість студентів, що склали іспити із задовільними оцінками.
6. Обчислити кількість мешканців населеного пункту на кінець року, якщо відомо кількість мешканців на початку року, кількість прибулих, вибулих, народжених та померлих.
7. Обчислити рівень виконання навчального навантаження (у відсотках) вчителями школи, якщо відомо заплановану та фактично виконану кількість годин.
8. Обчислити вартість партії товару з двох видів, якщо задано ціну одиниці товару та кількість товару кожного виду.
9. Обчислити вартість товару та вартість товару з врахуванням торгової знижки, якщо задана ціна одиниці товару (грн.), кількість товару (шт.) та торгова знижка (%).



10. Обчислити суму для оплати комунальних послуг, якщо задано місячну вартість комунальних послуг та кількість місяців, за які потрібно провести розрахунок.
 11. Обчислити урожайність (ц/га) зернових фермерського господарства, якщо задано площу посівів (га) та валовий збір (т).
 12. Обчислити суму для оплати за водопостачання за звітний період, якщо задано початкові та поточні покази лічильника, а також вартість 1 м^3 води.
 13. Обчислити суму оплати за електроенергію за звітний період, якщо задано початкові і поточні покази лічильника та вартість 1 кВт-год .
 14. Обчислити вартість пального (грн), якщо задано ціну 1 л (грн) та витрати пального (л).
 15. Обчислити суму оплати за опалення, якщо задано вартість опалення 1 м^2 та опалювальна площа.
- VI. Зберегти проект у папці Lab2_z2. Продемонструвати створений проект викладачеві. Записати отримані результати.

VII. Захистити виконану роботу.

Контрольні запитання

1. Що таке змінна? Які правила задання змінних?
2. Назвіть і охарактеризуйте типи даних у Visual Basic.
3. Які способи оголошення змінних у Visual Basic?
4. Як здійснюється оголошення змінної за допомогою суфікса?
5. Назвіть типи змінних у Visual Basic.
6. Що таке арифметичний вираз?
7. Які пріоритети виконання операцій?
8. Чи змінюється значення константи?
9. Яку функцію виконує коментар?
10. Функції та їх використання у Visual Basic.
11. Вбудовані функції VB.



Лабораторна робота № 3. Реалізація проектів, що містять обчислювальні процеси з розгалуженнями

Мета: Навчитись використовувати засоби VB для розробки проектів, що реалізують обчислювальні процеси з розгалуженнями.

Розглянемо конструкції Visual Basic, які дозволяють керувати послідовністю виконання дій у програмі. Ці конструкції називають **операторами управління**.

З алгоритмічної точки зору оператори управління поділяються на **”оператори розгалуження”** та **”оператори циклу”**. Застосування операторів розгалуження дозволяє виконувати певні дії в залежності від умов, що виникають у програмі. За допомогою циклів можна виконувати дії, що повторюються.

Зазвичай, конструкції виконуються в тій послідовності, в якій вони записані в програмі. Однак досить часто виникає необхідність змінити порядок виконання команд залежно від результату виконання деякої умови. У Visual Basic існують оператори, призначені для управління порядком виконання команд. До основних операторів управління належать: **умовний оператор (If)** та оператор вибору (**Select Case**). В свою чергу, оператори **If** бувають двох видів: **If...Then** та **If...Then...Else**.

Конструкція **If...Then** застосовується тоді, коли необхідно виконати певні дії залежно від деякої умови. Конструкція **If...Then...Else** використовується у тому випадку, коли необхідно виконати різні дії залежно від умови.

Логічні вирази та їх використання

Основою для прийняття рішень у конструкціях управління є **логічні вирази**. Вони повертають одне з двох значень: **True** (істинне) або **False** (хибне). В логічних виразах застосовують оператори порівняння (табл. 3.1).



Оператори порівняння VB

Оператор	Зміст
>	Більше
>=	Більше або дорівнює
<	Менше
<=	Менше або дорівнює
=	Дорівнює
<>	Не дорівнює

Розглянемо приклади використання операторів порівняння.

Приклад 1.

Логічний вираз $2.9994 < 2.9995$ має значення **True**;

Логічний вираз $3.1416 \leq 3.1416$ має значення **True**;

Логічний вираз $1.3E-18 < 2.3E-19$ має значення **False**;

Логічний вираз $Y^2 < Abs(Y)$ має значення **True** для $Abs(Y) < 1$;

Розглянемо деякі логічні операції.

Логічне множення And (i) – повертає значення True, якщо всі вирази, які беруть участь в конструкції, мають значення True, в інших випадках повертається значення False.

Операцію логічного множення можна запам'ятати за допомогою такої фрази: **”Значення логічного добутку істинне тільки тоді, коли значення всіх логічних співмножників істинні”**. Результати логічного множення наведені в таблиці 3.2.

Таблиця 3.2

Таблиця істинності логічного множення

A	B	A And B
True	True	True
True	False	False
False	True	False
False	False	False

”Істинне і Істинне є Істинне, в інших випадках – Хибне”



Логічне додавання Or (або) – повертає значення True, якщо хоч один з виразів, які беруть участь у конструкції, має значення True. У випадку, коли всі вирази мають значення False, повертається значення False.

Операцію логічного додавання можна запам'ятати за допомогою фрази: *"Значення логічної суми хибне тільки тоді, коли значення всіх логічних доданків хибні"*. Результати логічного додавання наведені в таблиці 3.3.

Таблиця 3.3

Таблиця істинності логічного додавання

A	B	A Or B
True	True	True
True	False	True
False	True	True
False	False	False

"Хибне або Хибне є Хибне, в інших випадках – Істинне"

Логічне заперечення Not (ні) – повертає протилежне значення до значення заданого виразу.

Операцію логічного заперечення можна запам'ятати так: *"Значення логічного заперечення хибне, якщо значення виразу істинне і навпаки"*. Результати логічного заперечення наведені в таблиці 3.4.

Таблиця 3.4

Таблиця істинності логічного заперечення

A	Not A
True	False
False	True

"Не Істинне є Хибне, Не Хибне є Істинне"



Приклад 2.

Логічний вираз $(X^2 > 0)$ **Or** $(X^2 = 0)$ має значення **True**;

Логічний вираз $(Y^2 \geq 0)$ **And** $(\text{Abs}(Z) \geq 0)$ має значення **True** (передбачається, що змінні X , Y і Z мають числовий тип);

Логічний вираз **Not** $(\text{Len}("abc") > 0)$ має значення **False**;

Логічний вираз $(\text{Len}("abc") < 3)$ **Or** $(\text{Len}("abc") > 3)$ має значення **False**;

Логічний вираз $(\text{Len}(S1 + S2) \geq \text{Len}(S1))$ **And** $(\text{Len}(S1 + S2) \geq \text{Len}(S2))$ має значення **True** (передбачається, що змінні $S1$ і $S2$ мають тип **String**).

Приклад 3.

Розглянемо спрощене юридичне правило:

"Громадянину (громадянці) призначається пенсія, якщо це чоловік і йому більше 60 років або якщо це жінка і їй більше 55 років, або якщо стаж роботи на шкідливому виробництві перевищив 25 років". Зведемо це правило до логічного виразу, істинність якого означає, що дана особа пенсійного віку.

Визначимо наступні змінні:

Змінна	Значення змінної
<i>Пенсіонер</i>	значення True , якщо людина пенсійного віку, і False , якщо людина не пенсійного віку
<i>Стать\$</i>	"чоловік" або "жінка"
<i>Вік!</i>	вік людини
<i>СтажЗаВислугою!</i>	стаж роботи на шкідливому виробництві (за вислугою років)

Використання логічного виразу для наведеного вище юридичного правила можна записати так:

Пенсіонер = $(\text{Стать\$} = \text{"чоловік"} \text{ And Вік!} \geq 60)$ **Or** $(\text{Стать\$} = \text{"жінка"} \text{ And Вік!} \geq 55)$ **Or** $\text{СтажЗаВислугою!} \geq 25$



Умовний оператор

Умовний оператор може бути записаний в однорядковому чи багаторядковому вигляді. Однорядковий оператор має такий синтаксис:

```
If <Логічний вираз> Then <Оператор1> [Else <Оператор2> ]
```

В цій конструкції після ключових слів **Then** та **Else** записується лише один оператор. Якщо результатом обчислення логічного виразу є значення **True**, то виконується <Оператор1>, який записаний за ключовим словом **Then**. Якщо логічний вираз набуває значення **False**, то виконується <Оператор2>, записаний за ключовим словом **Else**, або наступний за **If** оператор, якщо елемент конструкції **Else** <Оператор2> відсутній.

Синтаксис багаторядкового оператора наступний:



```
If <Логічний вираз> Then  
  <ПослідовністьОператорів1>  
  [Else  
  <ПослідовністьОператорів2>]  
End If
```

Запис *ПослідовністьОператорів1* та *ПослідовністьОператорів2* строго позиційований:

✓ в рядку за ключовим словом **Then** не можна записувати жодного оператора;

✓ в рядку за ключовим словом **Else** можна записати інший оператор, але йому повинна передувати двокрапка.

Ключові слова **End If** означають кінець багаторядкового оператора. Їх наявність у цьому випадку обов'язкова.

Якщо результатом обчислення логічного виразу є значення **True**, то виконується <ПослідовністьОператорів1>, які записані за ключовим словом **Then**, а якщо логічний вираз набуває значення **False**, то – <ПослідовністьОператорів2>, записаних за ключовим словом **Else**, або наступний за **If...End If** оператор, якщо елемент конструкції **Else** <ПослідовністьОператорів2> відсутній.

В операторі **If** перевіряється на істинність тільки одна умова. Якщо ж є потреба виконання операторів чи послідовності



операторів залежно від результатів перевірки декількох умов, то це можна зробити за допомогою наступної конструкції:

```
If <Логічний вираз1> Then  
    <ПослідовністьОператорів1>  
[ ElseIf <Логічний вираз2> Then  
    <ПослідовністьОператорів2>  
...]  
[ Else <ПослідовністьОператорівN> ]  
End If
```

Приклад 4.

Приклад алгоритму обчислення максимального з двох чисел ($Число1 \neq Число2$) з використанням конструкції **If...Then**:

```
МаксЧисло=Число1  
If Число2 >Число1 Then МаксЧисло=Число2
```

Приклад 5.

Розглянемо попередній приклад з використанням конструкції **If...Then...Else...**:

```
If Число1 >Число2 Then МаксЧисло=Число1 Else _  
МаксЧисло=Число2
```

Приклад 6.

Розглянемо юридичне правило визначення пенсійного стажу: "Громадянину (громадянці) призначається пенсія, якщо це чоловік і йому більше 60 років або якщо це жінка і їй більше 55 років, або якщо стаж роботи на шкідливому виробництві перевищив 25 років".

Наприклад, якщо жінці 58 років, з яких вона 37 років пропрацювала на шкідливому виробництві, то пенсійний стаж буде визначатися як максимальне з двох чисел $58-55=3$ і $37-25=12$ і буде дорівнювати 12 рокам.

Якщо жінці 70 років, з яких вона 10 років пропрацювала на шкідливому виробництві, то її пенсійний стаж за віком ($70-55=15$) буде перевищувати її пенсійний стаж за вислугою років ($10 \leq 25$,



стаж роботи на шкідливому виробництві не перевищує 25 років, пенсія не призначається) і дорівнюватиме 15 рокам.

Програмний код можна записати наступним чином:

```
Private Sub Command1_Click()  
Dim Stat, Penciya As String, _  
Vik, Stag_vusluga As Single, _  
P_stag, P_stag_vik, P_stag_vusluga As Integer  
Stat = Text1.Text  
Vik = Val(Text2.Text)  
Stag_vusluga = Val(Text3.Text)  
If (Stat = "Чоловік") And (Vik >= 60) Or _  
(Stat = "Жінка" And Vik >= 55) Or _  
(Stag_vusluga >= 25) Then  
Penciya = "Призначається"  
Else  
Penciya = "Не призначається"  
End If  
Text4.Text = Penciya  
If (Vik >= 60 And Stat = "Чоловік") Or _  
(Stag_vusluga >= 25) Then  
P_stag_vik = Vik - 60  
P_stag_vusluga = Stag_vusluga - 25  
End If  
If (Vik >= 55 And Stat = "Жінка") Or _  
(Stag_vusluga >= 25) Then  
P_stag_vik = Vik - 55  
P_stag_vusluga = Stag_vusluga - 25  
End If  
P_stag = P_stag_vik  
If P_stag_vusluga > P_stag Then P_stag = P_stag_vusluga  
Text5.Text = Str(P_stag)  
End Sub
```

Результат виконання програмного коду показано на рисунку 3.1.

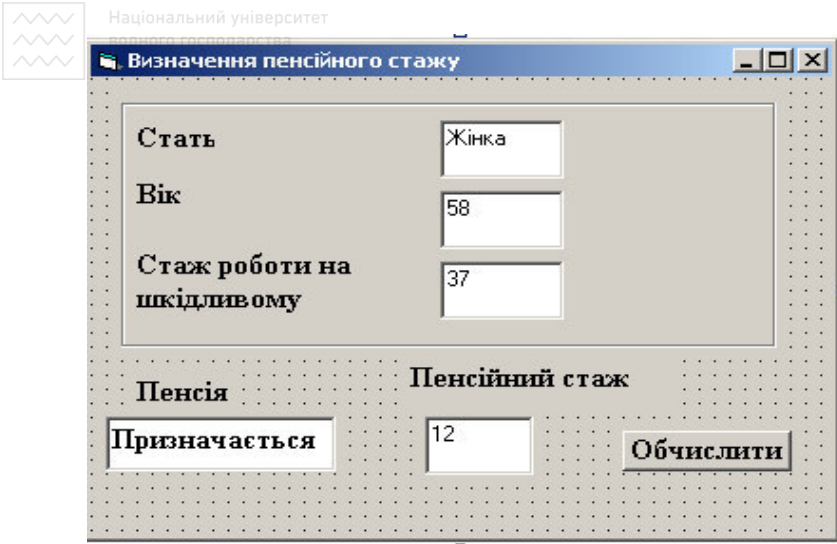


Рис. 3.1. Визначення пенсійного стажу

У мові Visual Basic є оператор вибору, який дозволяє опрацювати в програмі декілька умов.

Синтаксис оператора вибору:

Select Case <Вираз>

Case <ДіапазонЗначень1>

<ПослідовністьОператорів1>

[**Case** <ДіапазонЗначень2>

<ПослідовністьОператорів2>

...]

[**Case Else**

<ПослідовністьОператорівN>]

End Select

Працює ця конструкція так. Спочатку обчислюється значення заданого виразу, потім виконується послідовність операторів, приписаних тому оператору **Case**, в діапазон значень якого потрапляє значення виразу. Діапазоном значень може бути одне значення виразу, перелік значень розділених комою, діапазон значень. Якщо значення не належить жодному діапазону, то виконується послідовність операторів, записаних після слів **Case Else** (якщо цей елемент присутній). Після завершення виконання



конструкції керування буде передане операторам, що розміщені після ключових слів **End Select**.

Зауважимо, що конструкція **Select Case** може виконати не більше, ніж одну з послідовностей операторів, які містяться в ній. Після того, як одне із значень перевірки збігається із значенням виразу, буде виконано відповідну послідовність команд і оператор закінчить свою роботу. Решта умов перевірятися не буде.

Завдання для виконання лабораторної роботи № 3

I. Завантажити середовище візуального програмування Visual Basic.

II. Розглянути зразки побудови алгоритму, форми та програмного коду (кнопка **Результат**) для обчислення значення функції:

$$Y = \begin{cases} -100 + X, & \text{якщо } X = -10 \\ 100 + X, & \text{якщо } X = 10 \\ \frac{e^X}{2(X + 0.45)}, & \text{якщо } X = 0 \end{cases}$$

Розмістити об'єкти на формі згідно зразка (рис. 3.2).

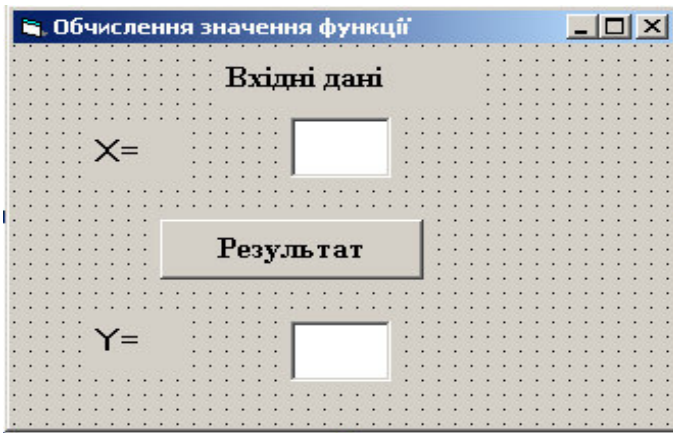


Рис. 3.2. Зразок розташування об'єктів на формі



Приклад побудови алгоритму розв'язку показано на рис. 3.3.

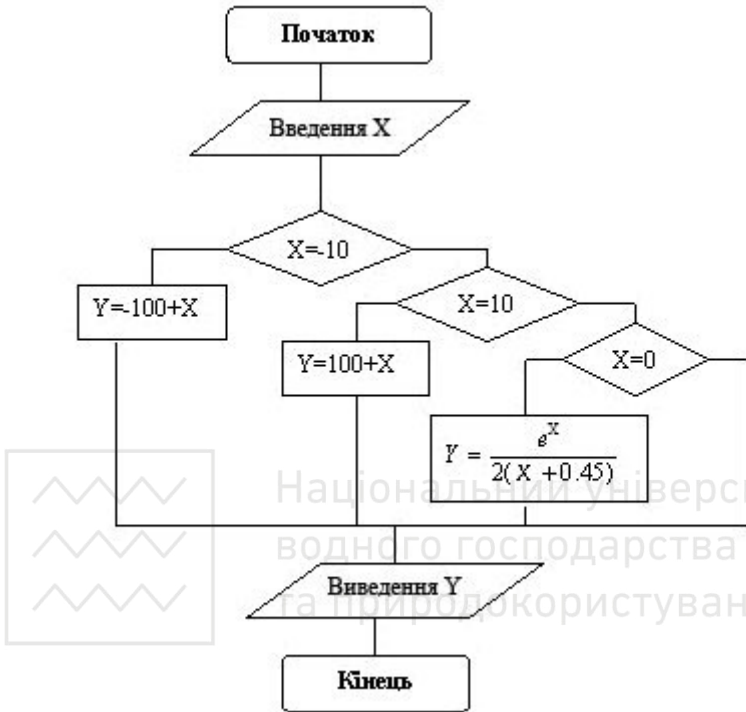


Рис. 3.3. Блок-схема алгоритму

Програмний код для обчислення значень функції Y з використанням умовного оператора може мати такий вигляд:

```
Private Sub Command1_Click()
X = Val(Text1.Text)
If X = -10 Then Y = -100 + X
If X = 10 Then Y = 100 + X
If X = 0 Then Y = Exp(X) / 2 / (X + 0.45)
Text2.Text = Str(Y)
End Sub
```



III. Використовуючи наведені вище зразки, виконати модифікацію форми та коду обробки події для командної кнопки відповідно до завдання свого варіанта:

$$1. y = \begin{cases} x - |x|, & x \geq 1 \\ x^3 - e^x, & |x| < 1 \\ \frac{1}{2(x^4 + 0.45)}, & x \leq -1 \end{cases}$$

$$2. y = \begin{cases} e^{|x|}, & x < 6 \\ 4\cos^3 x - 5, & x > 6 \\ \sqrt{|tgx|}, & x = 6 \end{cases}$$

$$3. y = \begin{cases} \arctg(-x^2 - 4), & x < -3 \\ \frac{1}{x}, & x \neq 0 \\ \sqrt[4]{x^5}, & x > 15 \end{cases}$$

$$4. z = \begin{cases} x^3 + y^2, & x = y \\ tg \frac{x}{2y}, & x \neq y \end{cases}$$

$$5. y = \begin{cases} 5x \sin x^2, & |x| = 4 \\ \frac{x \ln(2x+1)}{2x}, & x \neq 0 \\ \sqrt{|tgx|}, & x > 2 \end{cases}$$



$$6. y = \begin{cases} \cos^3 x^2 + x, & x > 10 \\ \ln^2 |x^3| + 0.6, & x = -1.3 \\ \frac{e^{3x}}{2(x+0.6)}, & x = 10 \end{cases}$$

$$7. y = \begin{cases} \frac{\sin^3 x + e^t}{2.56}, & x + 2 > t \\ \ln |tx|, & x + 2 \leq t \end{cases}$$

$$8. y = \begin{cases} \frac{23e^t}{tz}, & t^2 + z^2 > 56 \\ \sqrt[3]{|\cos z|^5}, & t^2 + z^2 = 56 \\ \lg |2 + zt|, & t^2 + z^2 < 56 \end{cases}$$

$$9. y = \begin{cases} \sqrt{z + 12.34}, & z > 0 \\ \operatorname{ctg} |\ln |z||, & z \leq 0 \end{cases}$$

$$10. z = \begin{cases} y^2 - 2\operatorname{tg} \frac{y-1}{2y}, & y > 2 \\ \sqrt{|y^3 - 4|}, & y < 2 \\ \operatorname{ctg} y^3, & y = 2 \end{cases}$$



$$11. t = \begin{cases} e^x + 2x|x|, & x < -2 \\ \sqrt{x}, & x > -2 \\ 2x + 34, & x = 2 \\ \lg|x + 6.8|, & x = 2 \end{cases}$$

$$12. p = \begin{cases} \sqrt{|tgz|}, & 1 < z < 2 \\ z^3 - 0.97, & 2 \leq z < 5 \\ \frac{z + 3.4}{5.4e^z}, & z \geq 5 \end{cases}$$

$$13. s = \begin{cases} ctg^3 x^2, & x \leq \pi \\ 3.6\pi x, & x > \pi \end{cases}$$

$$14. y = \begin{cases} \sqrt{|9 - x \sin x|}, & x^3 + 1 \geq 25 \\ e + \sqrt{e + 2}, & x^2 - e < 1 \\ 12.3 \frac{\lg x}{2x}, & x = 12.3 \end{cases}$$

$$15. y = \begin{cases} 1 - \frac{e}{2\cos\sqrt{\pi}}, & \pi x \leq 3.45 \\ \lg|tg x| + \pi^2, & \pi x > 3.45 \end{cases}$$

IV. Зберегти проект у папці Lab3_z1.

V. Продемонструвати створений проект викладачеві. Записати в зошит отримані результати.



VI. Розглянути зразок форми та програмного коду (**команда Результат**) для визначення належності точки $M(x,y)$ фігури (рис. 3.4).

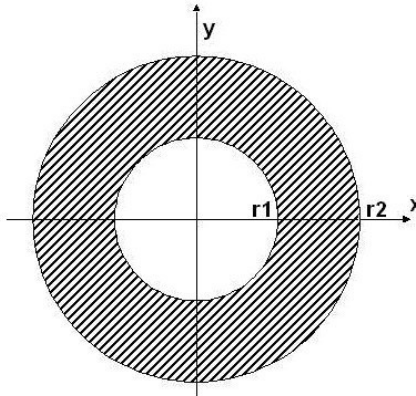


Рис. 3.4. Задана геометрична фігура

На рисунку 3.5 показано розташування об'єктів на формі.

Рис. 3.5. Зразок розташування об'єктів на формі

Програмний код для визначення належності точки геометричній фігури можна записати так:



```
Private Sub Command1_Click()
```

```
Const r1 = 1
```

```
Const r2 = 2
```

```
x = Val(Text1.Text)
```

```
y = Val(Text2.Text)
```

```
If (x^2 + y^2 <= r2^2) And (x^2 + y^2 >= r1^2) Then
```

```
Text3.Text = "т. М належить фігурі"
```

```
Else
```

```
Text3.Text = "т. М не належить фігурі"
```

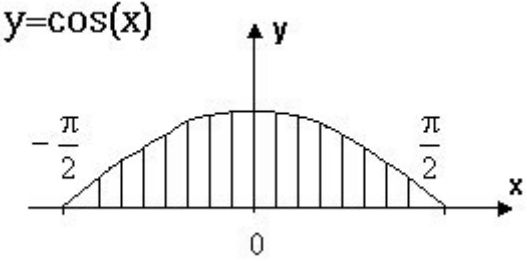
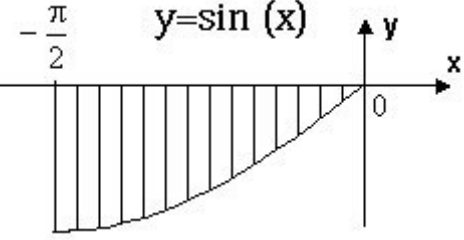
```
End If
```

```
End Sub
```

VII. Використовуючи наведені вище зразки форми та програмного коду задачі про визначення належності точки $M(x,y)$ фігурі Φ , виконати модифікацію форми та коду обробки події для командної кнопки відповідно до завдання свого варіанта. Координати точки $M(x,y)$ та вид фігури Φ обрати з таблиці 3.5.

Таблиця 3.5

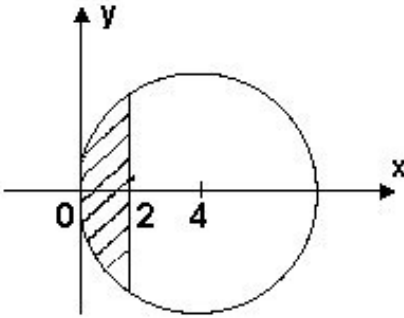
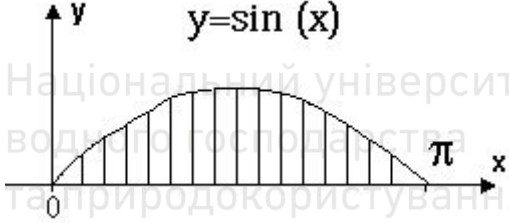
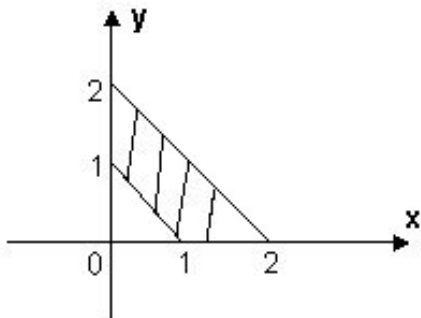
Координати точки $M(x,y)$ та вид фігури Φ

№	Координати точки	Вид фігури Φ
1	$M\left(-\frac{\pi}{3}; \frac{\pi}{3}\right)$	
2	$M\left(-\frac{\pi}{3}; \frac{\pi}{3}\right)$	

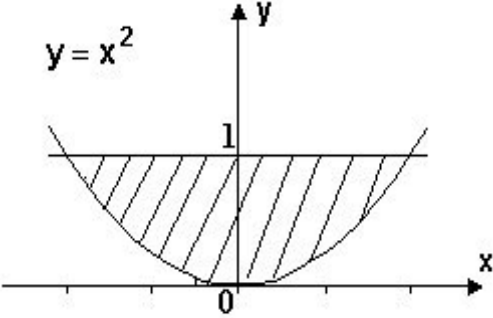
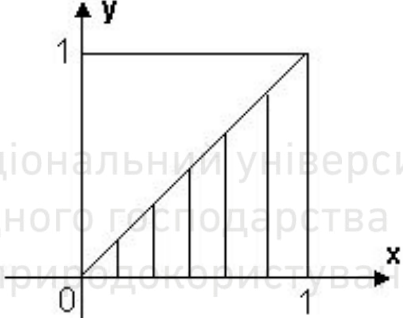
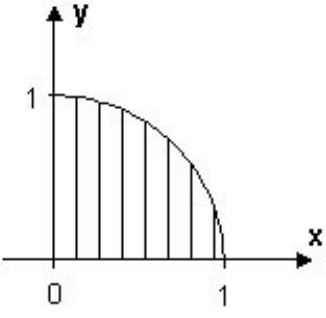


3	$M(1.5; 2.3)$	
4	$M\left(\frac{\pi}{8}; \frac{2}{6}\pi\right)$	
5	$M(-2.5; 0.3)$	

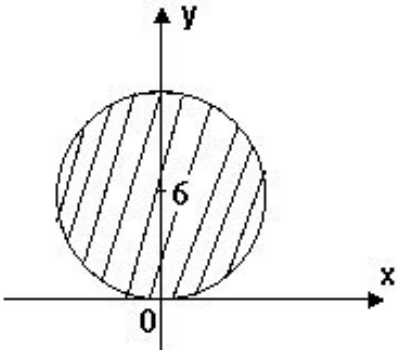
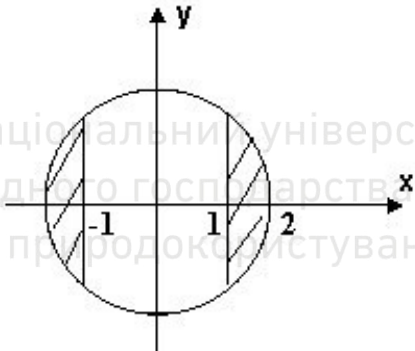
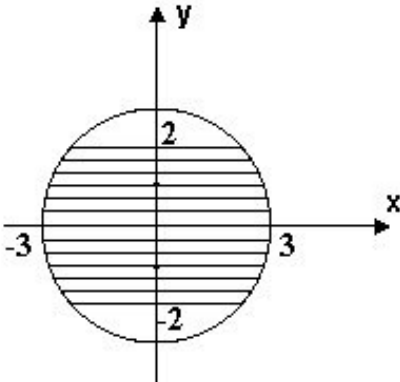


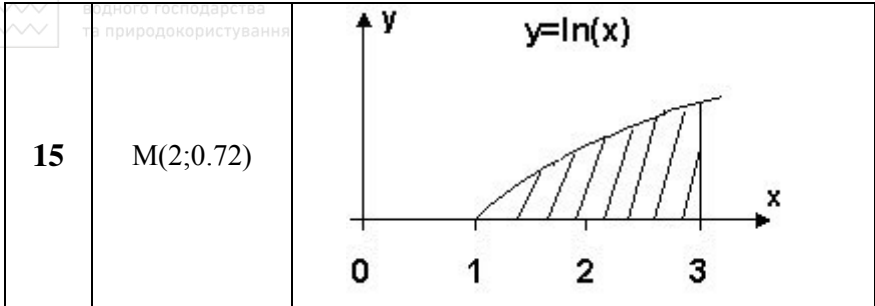
6	M(1.8;1.7)	
7	$M\left(\frac{3\pi}{4}; \frac{\pi}{12}\right)$	 <p>$y = \sin(x)$</p>
8	M(0.8;1.4)	



9	$M(-0.3;0.5)$	 <p>$y = x^2$</p>
10	$M(0.4;0.5)$	
11	$M(0.8;0.6)$	



12	M(5.3;6.4)	
13	M(1.7;1.5)	
14	M(1.8;1.3)	



VIII. Зберегти проект у папці Lab3_z2. Продемонструвати створений проект викладачеві. Записати отримані результати.

IX. Розглянути зразок форми (рис. 3.6) та програмного коду (команда **Обчислити**) для визначення суми виплат S при заданому розмірі початкового вкладу V , заданому терміні вкладу T і річній відсотковій ставці P , що обчислюється за формулою:

$$S = V \left(1 + \frac{P}{100} \right)^T, P = \begin{cases} 12\%, & \text{якщо } 0 < V < 4000 \\ 13\%, & \text{якщо } V \geq 4000 \end{cases}$$

Рис. 3.6. Зразок розташування об'єктів на формі



Програмний код для визначення суми виплат S можна записати так:

```
Private Sub Command1_Click()  
Dim P, T, V As Single  
Dim S As Currency  
V = Val(Text2.Text)  
T = Val(Text3.Text)  
If V >= 4000 Then P = 13 Else P = 12  
S = V * (1 + P / 100) ^ T  
Text4.Text = Str(S)  
End Sub
```

X. Використовуючи наведені вище зразки, розробити форму із заголовком "Обчислювальні процеси з розгалуженнями", записати алгоритм та ввести програмний код для розв'язування наступних задач:

1. Обчислити вартість товару та вартість товару із врахуванням торгової надбавки, якщо задано кількість товару (шт.), торгову надбавку (%), а ціна одиниці товару (грн.) визначається наступним чином:

Кількість товару (шт.)	Ціна одиниці товару (грн.)
≥ 2000	45
< 2000	65

2. Обчислити залишок на банківському рахунку після проведення банківської операції, якщо відомо залишок на рахунку до операції, зняття коштів з рахунку, а надходження визначається наступним чином:

Операція	Надходження (грн.)
№1	2000
№2	1300

3. Обчислити суму оплати виконаних будівельних робіт, якщо відомо відсоток виконання запланованого обсягу робіт та вартість запланованого обсягу робіт, а штрафні санкції визначаються наступним чином:

Виконання обсягу робіт (%)	Штрафні санкції
<50	0,1% вартості запланованого обсягу
≥ 50	відсутні

4. Обчислити вартість товару та вартість товару із врахуванням торгової надбавки, якщо задано кількість товару (шт.), ціну одиниці товару (грн.), а торгова надбавка (%) визначається наступним чином:

Кількість товару (шт.)	Торгова надбавка (%)
≥ 5000	2
< 5000	15

5. Обчислити вартість товару та вартість товару із врахуванням торгової знижки, якщо задано кількість товару (шт.) та ціну одиниці товару (грн.), а торгова знижка (%) визначається наступним чином:

Кількість товару (шт.)	Торгова знижка (%)
≥ 1000	5
< 1000	0.5

6. Обчислити кількість мешканців населеного пункту на кінець року, якщо відомо кількість прибулих, вибулих, народжених, померлих та кількість мешканців на початку року, що визначається наступним чином:

Населений пункт	Кількість мешканців на початку року
Старожуків	2000
Обарів	1500



7. Обчислити рівень виконання навчального навантаження (у відсотках) працівником школи, якщо відомо фактично виконану кількість годин та заплановану, що визначається наступним чином:

Прізвище та ініціали	Запланована кількість годин
Шевчук О.П.	750
Лавренчук Т.А.	650

8. Обчислити вартість партії товару, якщо задано ціну одиниці товару (грн.) та кількість товару (шт.), що визначається наступним чином:

№ партії	Кількість товару (шт.)
№1	320
№2	565

9. Обчислити вартість товару та вартість товару із врахуванням торгової знижки, якщо задано торгову знижку (%), кількість товару (шт.) та ціну одиниці товару (грн.), що визначається наступним чином:

Кількість товару (шт.)	Ціна одиниці товару (грн.)
≥ 1000	15
< 1000	35

10. Обчислити суму для оплати комунальних послуг навчальним закладом, якщо задано кількість місяців, за які потрібно провести розрахунок та місячну вартість комунальних послуг, що визначається наступним чином:

ЗОШ	Місячна вартість (грн.)
№10	12000
№22	50000



11. Обчислити урожайність (ц/га) зернових фермерського господарства, якщо задано валовий збір (т) та площу посівів (га), що визначається наступним чином:

Фермерське господарство	Площа посівів (га)
”Світанок”	35
”Свобода”	120

12. Обчислити суму оплати за водопостачання за звітний період, якщо задано початкові та поточні покази лічильника та вартість 1 м³ води, що визначається наступним чином:

Витрати води (м ³)	Вартість 1 м ³ (грн.)
≤ 10 м ³	4.15
> 10 м ³	6

13. Обчислити суму оплати за електроенергію за звітний період, якщо задано початкові та поточні покази лічильника та вартість 100 кВт-год, що визначається наступним чином:

Витрати електрики	Вартість 100 кВт (грн.)
≤ 300	24
> 300	32

14. Обчислити вартість пального (грн), якщо задано об’єм закупленого пального (л) та ціну 1 л (грн), що визначається наступним чином:

Об’єм пального (л)	Ціна 1 л (грн.)
>2000	15.95
≤1000	12.05

15. Обчислити суму оплати за опалення, якщо задано вартість опалення 1 м² та опалювальна площа, що визначається наступним чином:

Опалювальна площа (м ²)	Вартість опалення 1 м ² (грн.)
≤250	5.95
>250	12.5

- XI. Зберегти проект у папці Lab3_z3. Продемонструвати створений проект викладачеві.



XII. Записати отримані результати.

XIII. Захистити виконану роботу.

Контрольні запитання

1. Які мовні конструкції для реалізації розгалужень використовують у Visual Basic?
2. Які операції відношення Ви знаєте?
3. Дати визначення простого і складного логічного виразу.
4. Логічне додавання. Правило логічного додавання.
5. Логічне множення. Правило логічного множення.
6. Логічне заперечення. Правило логічного заперечення.
7. Різновиди однорядкової форми умовного оператора та їх використання.
8. Різновиди багаторядкової форми умовного оператора та їх використання.
9. Конструкція Select Case та її використання.



Лабораторна робота №4. Розробка проектів, що реалізують циклічні обчислювальні процеси

Мета: Навчитись використовувати засоби VB для розробки проектів, що реалізують циклічні обчислювальні процеси.

Циклічні обчислювальні процеси

Циклічним обчислювальним процесом називають обчислювальний процес, в якому певна послідовність дій повторюється скінченну кількість разів.

Впорядковану послідовність дій, що повторюються, називають **циклом**. Перелік дій циклу називають **тілом циклу**.

Для організації циклічного обчислювального процесу необхідно задати початкове значення параметра циклу, закон його зміни та умову виходу із повторень. Часто реалізація циклічного обчислювального процесу вимагає розрахунку числа повторень циклу.

Розглянемо приклади.

Приклад 1.

Необхідно обчислити і надрукувати квадрати чисел натурального ряду: 1, 2, 3, ..., 100 (одержати ряд 1, 4, 9, ..., 10000). Алгоритм розв'язування задачі має вигляд:

1. Змінній *Число* присвоїмо значення 0.
2. Збільшимо її значення на одиницю і результат знову присвоїмо змінній *Число*.
3. Піднесемо нове значення до квадрата.
4. Надрукуємо результат.

Багаторазово (100 разів) повторимо виконання пунктів 2..4.

Приклад 2.

Нехай маємо пачку аркушів із списками студентів усіх груп факультету. Необхідно визначити, скільки *в середньому студентів* навчається в одній групі. Очевидно, що для цього необхідно просумувати кількість студентів усіх груп і отриману суму поділити на кількість груп.

Можна запропонувати такий алгоритм розв'язування цієї задачі:

Нехай значенням змінної *Номер* буде номер групи.

1. Змінним *Номер* та *Сума* присвоїмо значення 0.
2. Збільшимо значення змінної *Номер* на одиницю і присвоїмо змінній *Номер* це нове значення.
3. Візьмемо черговий листок із списком студентів групи. Додамо до значення змінної *Сума* число, що дорівнює кількості студентів цієї групи, і присвоїмо їй отримане значення.
4. 2-е і 3-є розпорядження будемо повторювати доти, поки не закінчатся аркуші в пачці.
5. Остаточне значення змінної *Сума* розділимо на остаточне значення змінної *Номер*.
6. Надрукуємо результат – середнє число студентів в одній групі.

В прикладі 2 *число повторюваних пунктів* всього два. Але алгоритм складніший від попереднього, тому що заздалегідь не встановлено *число повторень* – воно дорівнює числу аркушів із списками груп, які попередньо не перераховані. Якщо доручити виконання цього алгоритму комп'ютеру, він самостійно порахує, скільки було аркушів у пачці – це і буде число груп (остаточне значення змінної *Номер*).

Головна відмінність цих прикладів полягає в тому, що в першому випадку число повторень заздалегідь відоме (**100**). В другому випадку число повторень залежить від умови, записаної в 4-му пункті алгоритму.

Ці підходи не є такими вже складними, але використання великої кількості операторів присвоювання – рутинна процедура. Застосування операторів циклу дозволяє значно скоротити обсяг програми.

Програмування циклів з лічильником

Для організації циклічних обчислювальних процесів, в яких кількість повторень відома наперед, використовують *оператор циклу* **For ... Next**. Він має таку структуру:



For < Ім'язмінної > = < ПочатковеЗначення > **To** _
< КінцевеЗначення > [**Step** < Крок >]
< ПослідовністьОператорів (Тіло циклу) >
Next [< Ім'язмінної >]

Тут <Ім'язмінної> – лічильник циклу, для якого задають його початкове значення, крок зміни і кінцеве значення. За допомогою лічильника програма відстежує, скільки разів виконалось тіло циклу (кількість повторень). Параметр < ПочатковеЗначення > вказує значення, яке присвоюється лічильнику перед першим виконанням циклу. Цикл виконується доти, поки значення лічильника не перевищить кінцевого значення <КінцевеЗначення >, вказаного після ключового слова **To**. Після кожного виконання циклу значення лічильника змінюється на величину < Крок >, вказану за ключовим словом **Step**. За його відсутності (за замовчуванням) кроком циклу приймається одиниця. Ключове слово **Next** означає кінець тіла циклу. < ПослідовністьОператорів (Тіло циклу)>– це та частина програми, число повторень якої визначається законом зміни значень лічильника.

При виконанні оператора циклу з лічильником насамперед цикл перевіряється на *несуперечність*. Якщо кінцеве значення лічильника більше початкового, то крок циклу повинен бути додатним. Якщо кінцеве значення лічильника менше початкового, то крок буде від'ємним.

Якщо буде виявлено суперечність, робота циклу буде негайно припинена і почнуть виконуватися оператори, записані за оператором циклу. В лічильнику циклу залишиться його початкове значення.

Якщо ж суперечності не буде виявлено, то оператори тіла циклу будуть виконані при початковому значенні лічильника. Потім значення лічильника буде змінено на величину кроку. Після цього (у випадку циклу з додатнім кроком) буде зроблена перевірка, чи не перевищує значення лічильника кінцевого значення. Якщо це так, робота оператора циклу буде припинена. Якщо ні, то оператори тіла циклу будуть виконані при новому значенні лічильника, після чого воно буде знову змінено на величину кроку і т.д. Аналогічно дії виконуються і при від'ємному значенні кроку зміни лічильника циклу.



В цій лабораторній роботі буде продемонстровано виведення результатів обчислень безпосередньо у форму за допомогою методу **Print**, що має наступний синтаксис:

[<Ім'я об'єкта>] **Print** [<список виведення>]

Одноразовим викликом методу **Print** можна вивести декілька значень елементів із списку виведення. В цьому випадку їх необхідно відокремити або крапкою, або крапкою з комою. Якщо елементи відокремлюються комою, то кожен наступний елемент виводиться з початку наступного стовпчика (ширина кожного стовпчика дорівнює 14 символам), а якщо крапкою з комою, то елементи виводяться через пропуск. Кожен наступний метод **Print** здійснює виведення інформації з нового рядка. Якщо попередній метод **Print** завершився крапкою чи крапкою з комою, то наступний метод **Print** продовжить виведення в поточному рядку. Коли потрібно здійснити перехід на новий рядок, застосовується метод **Print** без параметрів.

У методі **Print** для виведення значення із заданої позиції поточного рядка використовують функцію **Tab** (<номер позиції>), а для виведення заданої кількості пропусків – функцію **Spс** (<кількість>). Для виведення інформації у потрібному форматі використовується функція **Format** (<вираз>, <формат>). Другим параметром найчастіше служать символи, які наведені в таблиці 4.1.

Таблиця 4.1

Параметри функції **Format**

Символ	Призначення
0	виведення символів, у випадку їх відсутності – нулів
#	виведення цифр, у випадку їх відсутності – пропусків
.	виведення десяткової крапки
E-, E+, e-, e+	виведення у експонентній формі



Приклад 3.

Розглянемо програму, яка реалізує циклічний алгоритм виведення чисел від 0 до 100 з кроком 20 з використанням оператора циклу з лічильником:

```
Private Sub Command1_Click()  
For i=0 to 100 Step 20 ' Перший рядок циклу  
Print i  
Next i ' Останній рядок циклу  
End Sub
```

Тут змінна i –лічильник циклу, 0 і 100 – початкове і кінцеве значення лічильника, 20 – крок, а послідовність операторів складається тільки з одного оператора: **Print** i – виведення значення лічильника на екранну форму.

Результатом роботи даної програми буде стовпчик чисел: $0, 20, 40, 60, 80$ і 100 . Після закінчення роботи циклу значення лічильника буде рівним 120 . Це число перевищує кінцеве значення лічильника.

Приклад 4.

Роботу циклу з лічильником можна продемонструвати і на прикладі програми, за допомогою якої відображаються графічні елементи, що повторюються, наприклад, пронумеровані прямокутники (рис.4.1).

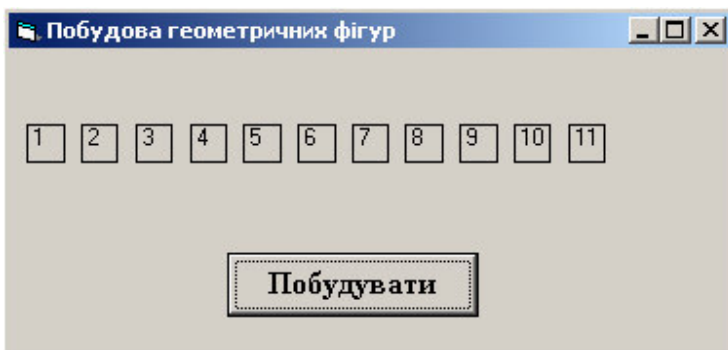


Рис. 4.1. Результат роботи програми виведення графічних об'єктів



Програмний код може мати вигляд:

```
Private Sub Command1_Click()  
Cls  
Scale (0, 4) - (40, -4)  
For i = 0 to 10  
Line(3 + i * 3, 1) - (1 + i * 3, 3), vbBlack, B  
Print i + 1  
Next i  
End Sub
```

Цикл з лічильником зручно використовувати для побудови графіка функції. Графік функції $y=f(x)$ будується на довільному відрізку осі абсцис на основі точок, коли для кожного значення аргументу x з цього відрізка комп'ютер може легко знайти значення функції y і розмістити на площині креслення точку з координатами (x,y) . Якщо точок буде достатньо багато, то вони зливатимуться в суцільну лінію – графік функції.

Приклад 5.

На рис. 4.2 показано екранну форму з розміщенням на ній графічним вікном. В ньому за допомогою методів **Pset i Line** з використанням циклу з лічильником побудований графік функції $y = ax^2 + bx + c$ ($a = 1, b = -2, c = -3$).

Код програми має вигляд:

```
Private Sub Command1_Click()  
Picture1.Scale (-3, 5)-(5, -5)  
  
' Побудова осі абсцис  
Picture1.Line (-3, 0)-(5, 0), vbCyan  
For i = -3 To 4  
Picture1.PSet (i, 0), vbCyan  
Picture1.Print i  
Next i  
  
' Побудова осі ординат  
Picture1.Line (0, -5)-(0, 5), vbCyan
```



Національний університет

```
For i = -4 To 5  
Picture1.PSet (0, i), vbCyan  
Picture1.Print i  
Next i
```

' Побудова графіка

```
For x = -2 To 4 Step 0.01  
y = x ^ 2 - 2 * x - 3  
Picture1.PSet (x, y), vbGreen  
Next x  
End Sub
```

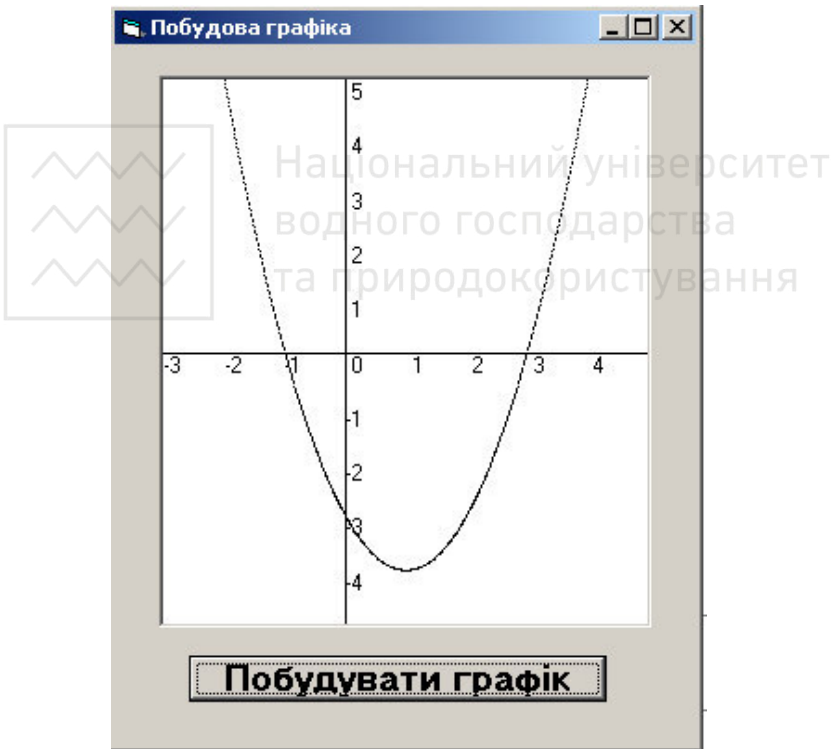


Рис. 4.2. Побудова графіка функції



Оператори циклу з умовою

Як було відмічено раніше, оператор циклу виду **For...Next** застосовується у випадках, коли кількість повторень тіла циклу відома.

Для програмування циклічних процесів із заздалегідь невідомим числом повторень використовують різні форми оператора циклу з умовою.

Оператор циклу з умовою – це багаторядковий оператор, перший рядок якого починається з ключового слова **Do** (виконати), а останній – з ключового слова **Loop** (цикл).

Нехай, наприклад, в задачі прикладу 1 (ст. 58) потрібно не тільки *знаходити* квадрати натуральних чисел від 1 до 100, але й додавати їх. Результат на кожному кроці циклу будемо присвоювати змінній *Сума* типу **Integer**. Чи обчислиться сума всіх 100 доданків?

При встановленому числі повторень, рівному 100, на одному із кроків циклу виникне помилка, викликана спробою присвоїти змінній *Сума* значення, що перевищує число 32767, яке є максимальним значенням змінної типу **Integer**. Ця спроба викличе появу на екрані попередження, зображеного на рис. 4.3.

Чи можна уникнути такої ситуації? Один із способів полягає у використанні *операторів циклу з умовою*. Чотири можливих варіанти використання таких операторів циклу наведено в таблиці 4.1.

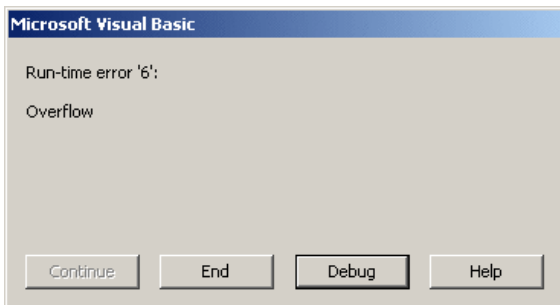


Рис. 4.3. Повідомлення про помилку Overflow



Розглянемо більш детально роботу циклів з умовою.

При використанні *конструкції типу While* тіло циклу виконується (**випадок 1**) або повторюється виконання тіла циклу (**випадок 3**), якщо значення <ЛогічнийВираз> дорівнює **True** (*Істинне*). В іншому випадку відбувається завершення циклу.

При використанні *конструкції типу Until* тіло циклу виконується (**випадок 2**) або повторюється виконання тіла циклу (**випадок 4**), якщо значення <ЛогічнийВираз> дорівнює **False** (*Хибне*). В іншому випадку цикл завершується.

Таблиця 4.1

Оператори циклу з умовою

1	2
Do While <ЛогічнийВираз> <Тіло циклу> Loop	Do Until <ЛогічнийВираз> <Тіло циклу> Loop
3	4
Do <Тіло циклу> Loop While <ЛогічнийВираз>	Do <Тіло циклу> Loop Until <ЛогічнийВираз>

Якщо умова записана після **Loop While(Until)** (**випадки 3 і 4**), тіло циклу виконуватиметься хоча б один раз.

Якщо ж умова записана після **Do While(Until)** (**випадки 1 і 2**), то тіло циклу може і не виконатись жодного разу.

Очевидно, що завжди можна замінити умову **While** умовою **Until** і навпаки, використавши логічне заперечення логічного виразу.

При програмуванні коду інколи виникає ситуація, коли необхідно перервати виконання циклу до його повного завершення. Це можна зробити за допомогою команди **Exit**. Ця команда завершує виконання циклу і передає управління наступній за циклом конструкції. Синтаксис цього оператора всередині циклу **For: Exit For**, а всередині циклу **Do – Exit Do**.

Приклад 6.

Умова $X > 100$ **Or** $Time > 600$ завершення циклу рівнозначна умові **Not** $(X > 100$ **Or** $Time > 600)$ продовження циклу або $X \leq 100$ **And** $Time \leq 600$.



Приклад 7.

Розглянемо наступну задачу. Петро Петрович купив мішок цукру масою 50 кг і вирішив кожного разу брати з цього мішка десяту частину його вмісту доти, поки в мішку залишиться менше 30 кг цукру.

Один з варіантів програмного коду, що визначає номер звертання Петра Петровича до мішка і масу цукру, що залишається в мішку, може бути таким:

```
Private Sub Command1_Click()
Dim Maca As Single, i As integer
Maca = 50: i = 0
Print "Номер:", "Залишок:"
Do
    ' Перший рядок циклу
    i = i + 1
    Maca = Maca - Maca / 10
    Print i, Maca
Loop Until Maca < 30
' Останній рядок циклу
End Sub
```

Очевидно, що замість **Until** $Maca < 30$ можна написати: **While** $Maca \geq 30$.

У випадку, якщо початкова маса цукру буде не 50, а 25 кг, то результатом виконання програмного коду буде:

Номер:	Залишок:
1	22.5

Незважаючи на те, що маса цукру менше 30 кг, Петро Петрович одного разу все-таки відсипле собі 2.5 кг цукру.

Щоб програма працювала коректно, потрібно записати таку конструкцію оператора циклу:

Do Until $Maca < 30$... **Loop**

В цьому випадку результат роботи програми матиме вигляд:

Номер	Залишок
1	45
2	40.5
3	36.45
...	...



Приклад 8.

Розглянемо програмний код задачі обчислення суми квадратів натуральних чисел від 1 до 100 до тих пір, поки ця сума буде меншою від деякого заданого числа.

```
' Максимальне значення змінної типу Integer  
Const Max As Integer=32767
```

```
Private Sub Command1_Click()  
Dim i As Integer, Sum As Integer  
i=0: Sum=0  
Do ' Перший рядок циклу  
i=i+1  
Sum=Sum+i^2  
' Останній рядок циклу  
Loop While Sum < Max - i^2  
Print "i :", "Sum :"  
Print i, Sum  
End Sub
```

Обчислення суми буде продовжуватися до тих пір, поки значення змінної *Sum* не досягне встановленого нами значення.

Програмування циклічних обчислювальних процесів неможливе без організації вікон для введення інформації, що можна зробити за допомогою функції **InputBox**.

Функція **InputBox** забезпечує формування вікна для введення інформації з відображенням його заголовка і запрошенням користувача до введення інформації.

Функція виклику вікна **InputBox** має наступний синтаксис з відповідними іменованими аргументами:

```
InputBox(<повідомлення> [,<заголовок вікна>] [,<текст за  
замовчуванням>] [,<Xпоз>][,<Yпоз>] [,<файл підказки>]  
[,<контекст>])
```

<повідомлення> – визначає текст, що відображається в діалоговому вікні як запрошення;

<заголовок вікна> – текст заголовка діалогового вікна; якщо цей параметр не вказано, то відображається назва додатка;



<текст за замовчуванням> – визначає значення, що відображається в рядку введення;

<Хпоз>, <Үпоз> – параметри, що вказують координати верхнього лівого кута вікна; за замовчуванням вікно відображається посередині екрана;

<файл підказки>, <контекст> – два необов’язкові параметри, що дають змогу відкривати певні файли довідкової системи.

Приклад 9.

Розглянемо програмний код з використанням функції InputBox для формування вікна введення:

```
Private Sub Command1_Click()  
Dim strPWD As String  
strPWD = InputBox ("Введіть пароль", _  
"Перевірка доступу", "*****")  
End Sub
```

Результатом роботи програми буде вікно введення (рис. 4.5):

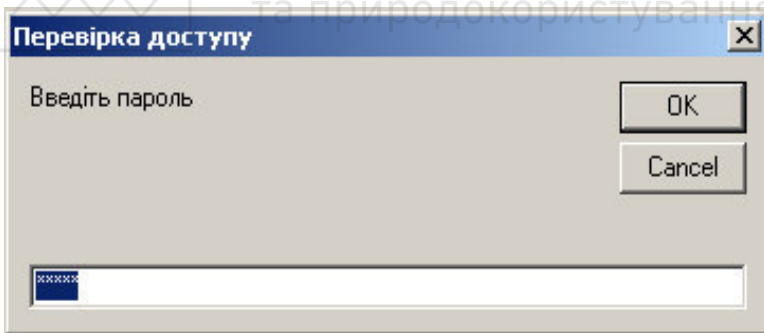


Рис. 4.5. Діалогове вікно

Завдання для виконання лабораторної роботи № 4

I. Завантажити середовище Visual Basic для створення нового проекту.

II. Розглянути зразки форм та програмних кодів :

а) задачі табулювання функції $y = x^2 + 10$, $2.6 \leq x \leq 8.8$, $\Delta x = 0.2$, з використанням оператора **FOR...NEXT**. У поданих зразках табулювання функцій (рис. 4.6, 4.7, 4.8, 4.9, 4.10) результати виводяться у багаторядкове поле редагування. Для цього властивості **MultiLine** потрібно задати значення **True** - дозвіл об'єкту працювати більше, ніж з одним рядком, а властивості **ScrollBars** необхідно задати значення **2 - Vertical** - наявність вертикальної смуги прокручування.

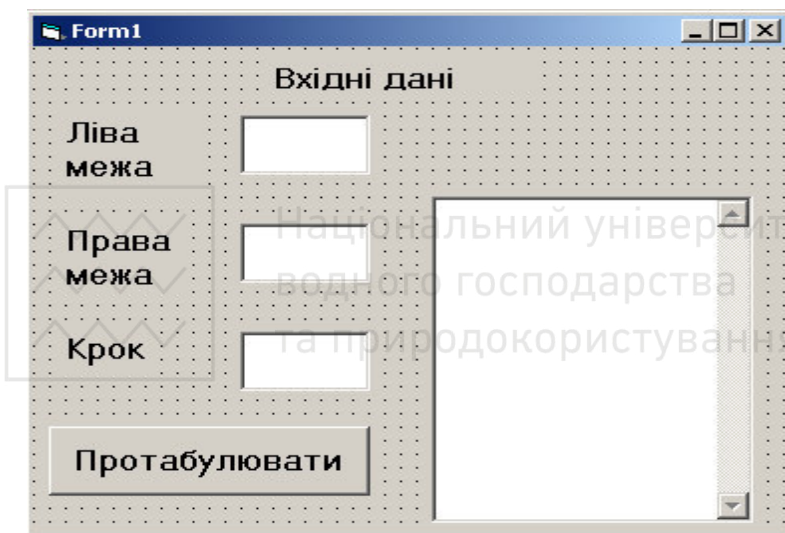


Рис. 4.6. Зразок №1 розташування об'єктів на формі

```
Private Sub Command1_Click()
Dim newline, space As String
Dim x, xp, xk, dx As Single
xp = Val(Text1.Text)
xk = Val(Text2.Text)
dx = Val(Text3.Text)
newline = Chr(13) + Chr(10)
space = Chr(9)
Text4.Text = " x" + space + " y" + newline
For x = xp To xk Step dx
```



$$y = x^2 + 10$$

```
Text4.Text = Text4.Text + Str(Format$(x, "0.000")) _  
+ space + Str(Format$(y, "0.0000")) + newline _  
Next  
End Sub
```

б) $y = x^2 + 10, x \geq 2.6, n=8, \Delta x=0.1$

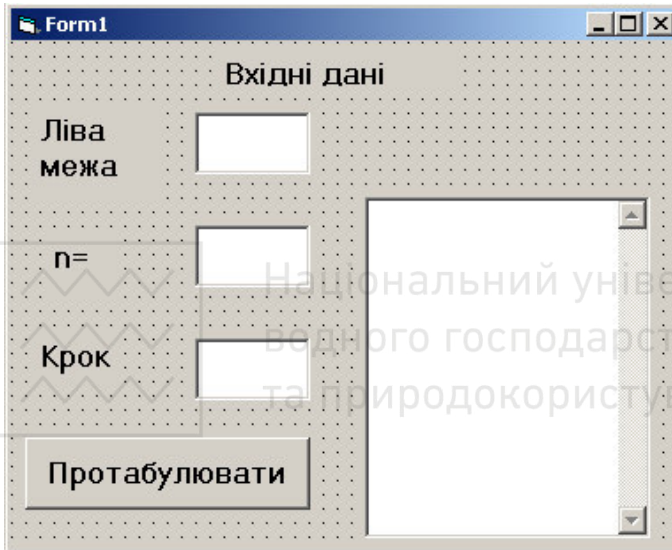


Рис. 4.7. Зразок №2 розташування об'єктів на формі

```
Private Sub Command1_Click()  
Dim newline, space As String  
Dim n, i As Integer  
xp = Val(Text1.Text)  
n = Val(Text2.Text)  
dx = Val(Text3.Text)  
newline = Chr(13) + Chr(10)  
space = Chr(9)  
Text4.Text = " x" + space + " y" + newline  
For i = 1 To n  
x = xp + (i - 1) * dx
```



Національний університет

водного господарства
та природокористування

$$y = x^2 + 10$$

```
Text4.Text = Text4.Text + Str(Format$(x, "0.000")) _  
+ space + Str(Format$(y, "0.0000")) + newline  
Next i  
End Sub
```

в) задачі табулювання функції $y = e^x + 10$ з використанням оператора **DO WHILE...LOOP**

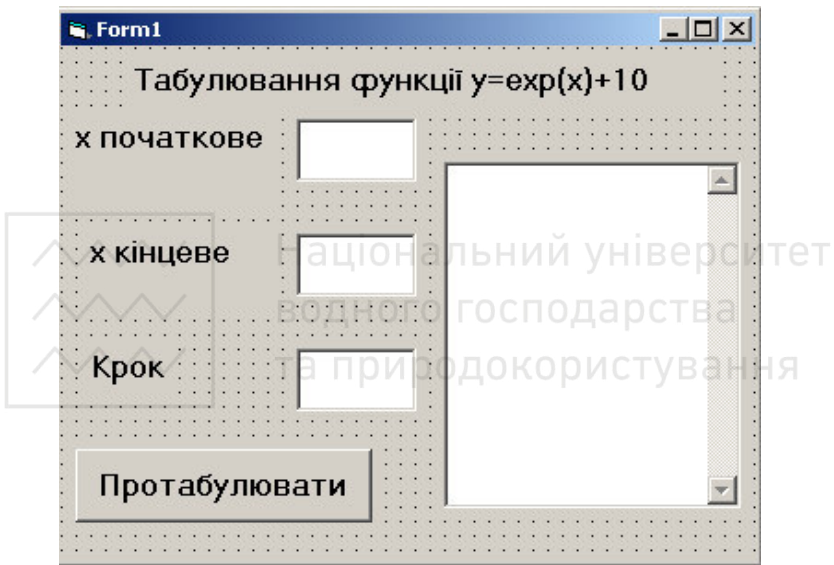


Рис. 4.8. Зразок №3 розташування об'єктів на формі

```
Private Sub Command1_Click()  
Dim newline, space As String  
xp = Val(Text1.Text)  
xk = Val(Text2.Text)  
dx = Val(Text3.Text)  
newline = Chr(13) + Chr(10)  
space = Chr(9)  
Text4.Text = " x" + space + " y" + newline  
x = xp
```




Національний університет

Do While x <= xk

y = Exp(x) + 10

Text4.Text = Text4.Text + Str(Format\$(x, "0.000"))

+ space + Str(Format\$(y, "0.0000")) + newline

x = x + dx

Loop

End Sub

г) задачі обчислення добутку

$$P = \prod_{m=mp}^{mk} m^2$$

Рис. 4.9. Зразок №4 розташування об'єктів на формі

```
Private Sub Command1_Click()
```

```
mp = Val(Text1.Text)
```

```
mk = Val(Text2.Text)
```

```
p = 1
```

```
For m = mp To mk
```

```
p = p * m ^ 2
```

```
Next m
```



Національний університет

```
Text3.Text = Str(p)
```

```
End Sub
```

д) задачі обчислення суми

$$S = \sum_{i=ip}^{ik} i^2$$

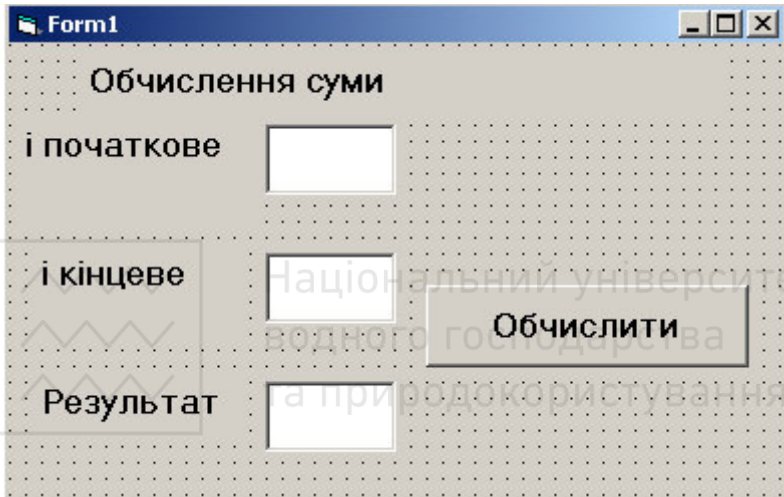


Рис. 4.10. Зразок №5 розташування об'єктів на формі

```
Private Sub Command1_Click()  
ip = Val(Text1.Text)  
ik = Val(Text2.Text)  
s = 0  
For i = ip To ik  
s = s + i ^ 2  
Next i  
Text3.Text = Str(s)  
End Sub
```

III. Використовуючи подані вище зразки форм та програмних кодів, розробити форми із заголовком “Циклічні процеси”,



виконавши модифікацію форми та коду обробки події для командної кнопки для виконання наступних завдань:

<p>1.</p>	$y = \frac{\cos^2 x}{x^2 + 1},$ <p>а) $3,8 \leq x \leq 7,6$, $\Delta x = 0,6$; б) $0,5 \leq x$, $\Delta x = 0,1$, $n = 9$.</p>	$y = \sum_{i=5}^{13} \frac{i^2 + 1}{i^3 + 2},$ $f = \prod_{k=5}^{13} \frac{k+3}{k+5} \frac{k+6}{k+6}.$
<p>2.</p>	$y = \frac{\operatorname{tg} 0,5x}{x^3 + 7,5},$ <p>а) $0,1 \leq x \leq 1,2$, $\Delta x = 0,1$; б) $0,5 \leq x$, $\Delta x = 0,05$, $n = 8$.</p>	$y = \sum_{i=1}^{10} \frac{i+2}{i+3} \frac{i+6}{i+6},$ $p = \prod_{l=8}^{17} \frac{l^2 - 2}{l+3}.$
<p>3.</p>	$y = \frac{e^{2x} - 8}{x+3},$ <p>а) $-1 \leq x \leq 2,3$, $\Delta x = 0,7$; б) $1,5 \leq x$, $\Delta x = 0,3$, $n = 6$.</p>	$S = \sum_{k=6}^{18} \frac{k^4 + 2}{2k^2 - 1},$ $V = \prod_{k=3}^7 \frac{k+3}{k^2 + 10k + 24}.$
<p>4.</p>	$y = \frac{x + \cos 2x}{3x},$ <p>а) $2,3 \leq x \leq 5,4$, $\Delta x = 0,8$; б) $x \geq 1,2$, $\Delta x = 0,2$, $n = 7$.</p>	$y = \sum_{j=3}^{12} \frac{j+2}{j+4} \frac{j+6}{j+6},$ $p = \prod_{s=5}^{11} \frac{S^2 + 3}{3S + 2}.$
<p>5.</p>	$y = \frac{x + \cos 2x}{x+2},$ <p>а) $0,2 \leq x \leq 10$, $\Delta x = 0,8$; б) $x \geq 0,6$, $\Delta x = 1,5$, $n = 6$.</p>	$S = \sum_{k=3}^{10} \frac{k-1}{k+3} \frac{k-2}{k-2},$ $p = \prod_{l=11}^{17} \frac{l-4}{l^4 + 7}.$
<p>6.</p>	$y = \frac{\cos^3 t^2}{1,5t + 2},$ <p>а) $2,3 \leq t \leq 7,2$, $\Delta t = 0,8$; б) $t \geq 0$, $\Delta t = 0,3$, $n = 5$.</p>	$S = \sum_{i=4}^{16} \frac{i^3 - 2i + 3}{i+4},$ $y = \prod_{n=4}^{12} \frac{n^2 + 2n + 3}{n+3}.$

<p>7.</p>	$z = \frac{x^3 + 2x}{3 \cos \sqrt{x} + 1},$ <p>а) $0 \leq x \leq 2, \Delta x = 0,4;$ б) $x \geq 0,3, \Delta x = 0,8, n = 7.$</p>	$y = \sum_{n=2}^9 \frac{n-2}{n^2+16},$ $p = \prod_{k=2}^{21} \frac{k+3}{k+1}.$
<p>8.</p>	$z = \frac{t + \sin 2t}{t^2 - 3},$ <p>а) $2,4 \leq t \leq 6,9, \Delta t = 0,4;$ б) $t \geq 3,1, \Delta t = 0,8, n = 6.$</p>	$S = \sum_{i=10}^{30} \frac{i^3 + 3i^2 + 7}{3i + 8},$ $z = \prod_{k=5}^{11} \frac{k+3}{k+7} \cdot \frac{k+9}{k+9}.$
<p>9.</p>	$y = \frac{x^3 - 2}{3 \ln x},$ <p>а) $4,5 \leq x \leq 16,4, \Delta x = 2,2;$ б) $x \geq 2, \Delta x = 1,5, n = 5.$</p>	$S = \sum_{k=3}^{10} \frac{k-1}{k+3} \cdot \frac{k-2}{k+3},$ $p = \prod_{n=6}^{14} \frac{n+3}{n-5} \cdot \frac{n}{n-9}.$
<p>10.</p>	$z = \frac{2,3t + 8}{ 2 \cos t + 1},$ <p>а) $0 \leq t \leq 6,5, \Delta t = 1,1;$ б) $t \geq 0,4, \Delta t = 0,9, n = 7.$</p>	$y = \sum_{n=4}^{11} \frac{n}{n^2 + 5n + 6},$ $p = \prod_{l=1}^{17} \frac{3l-4}{l^2+7}.$
<p>11.</p>	$y = \frac{5tg(x+7)}{(x+3)^2},$ <p>а) $1,2 \leq x \leq 6,3, \Delta x = 0,2;$ б) $x \geq 0,2, \Delta x = 0,1, n = 5.$</p>	$S = \sum_{k=6}^{18} \frac{k^4 + 2}{2k^2 - 1},$ $f = \prod_{n=3}^9 \frac{n}{n+2} \cdot \frac{n}{n+5}.$
<p>12.</p>	$y = \frac{1,5t - \ln 2t}{3t + 1},$ <p>а) $2,5 \leq t \leq 9, \Delta t = 0,8;$ б) $t \geq 0,8, \Delta t = 1,2, n = 6.$</p>	$y = \sum_{n=5}^{15} \frac{n+3}{n+5} \cdot \frac{n+7}{n+7},$ $p = \prod_{l=8}^{17} \frac{l^2 - 2}{l + 3}.$
<p>13.</p>	$y = \frac{2,5x^3}{e^{2x} + 2},$	$S = \sum_{i=4}^{16} \frac{i^3 - 2i + 3}{i + 4},$

	<p>a) $0 \leq x \leq 0,5$, $\Delta x = 0,1$; б) $x \geq -0,8$, $\Delta x = 0,25$, $n = 6$.</p>	$w = \prod_{k=1}^9 \frac{k+1}{k+5}$
14.	<p>$y = \frac{5 \lg x}{x^2 - 1}$</p> <p>a) $1,2 \leq x \leq 3,8$, $\Delta x = 0,4$; б) $x \geq 2$, $\Delta x = 1,5$, $n = 8$.</p>	$y = \sum_{n=1}^6 \frac{n+3}{n+5}$ $p = \prod_{l=1}^{14} \frac{l+5}{2l}$
15.	<p>$z = \frac{6x+4}{\sin 3x - x}$</p> <p>a) $2,3 \leq x \leq 7,8$, $\Delta x = 0,9$; б) $x > 2,8$, $\Delta x = 0,3$, $n = 6$.</p>	$S = \sum_{k=5}^{16} \frac{4k-5}{2k+2}$ $y = \prod_{k=2}^8 \frac{k^2}{k+3}$

IV. Кожний із проектів зберегти в окремій папці Lab4_z1, Lab4_z2, Lab4_z3, Lab4_z4. Продемонструвати створені проекти викладачеві. Записати отримані результати.

V. Розглянути зразки форми (рис. 4.11, 4.12) та програмного коду задачі обчислення приросту жителів кожного з N населених пунктів на кінець року, якщо відомо кількість народжених та кількість померлих в населеному пункті на кінець поточного року:

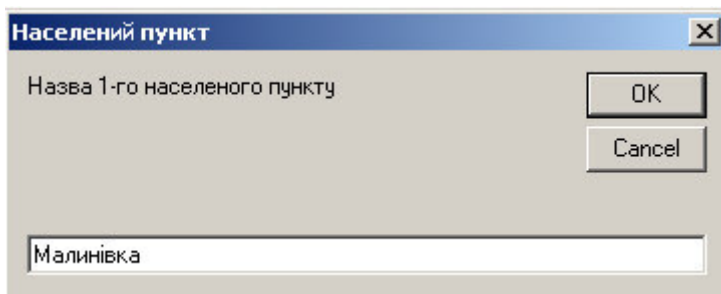


Рис. 4.11. Зразок №6 використання вікон функції InputBox для введення даних



Населений пункт	Кількість народжених	Кількість померлих	Приріст населення
Малинівка	524	45	479
Піддубці	458	34	424
Старожуків	387	24	363

Визначення приросту населення

Рис. 4.12. Зразок №7 форми з результатами задачі

Програмний код задачі можна записати таким чином:

```
Private Sub Command1_Click()  
Const N% = 3  
Dim Nazva As String * 10  
For i% = 1 To 50  
Print " _ ";  
Next i%  
Print  
Print "Населений", "Кількість", "Кількість", "Приріст"  
Print "пункт", "народжених", "померлих", "населення"  
For i% = 1 To 50  
Print " _ ";  
Next i%  
Print  
For i% = 1 To N%  
Title$ = "Населений пункт"  
Prt$ = "Назва " & i% & "-го населеного пункту"  
Nazva = InputBox(Prt$, Title$)  
Title$ = "Кількість народжених"
```



```
Prt$ = "К-ть народжених в " & i% & "-му населеному  
пункті"  
nar = Val(InputBox(Prt$, Title$))  
Title$ = "Кількість померлих"  
Prt$ = "К-ть померлих в " & i% & "-му населеному пункті"  
pom = Val(InputBox(Prt$, Title$))  
r = nar - pom  
Print Format(Nazva, "#####"), nar, pom, r  
Next i%  
For i% = 1 To 50  
Print " _";  
Next i%  
Print  
End Sub
```

VI. Використовуючи наведені вище зразки, виконати модифікацію форми та коду обробки події для командної кнопки відповідно до свого варіанта для розв'язування наступних задач:

1. Обчислити вартість товару та вартість товару із врахуванням торгової надбавки кожного з N видів товару, якщо задано ціну одиниці товару (грн.), кількість товару (шт.) та торгову надбавку (%) кожного виду товару.
2. Обчислити для кожного з N клієнтів залишок на банківському рахунку після проведення банківської операції, якщо відомо залишок на рахунку до операції, надходження та зняття коштів з рахунку кожного клієнта.
3. Обчислити для кожного з N об'єктів суму оплати виконаних будівельних робіт, якщо відомо відсоток виконання запланованого осягу робіт та загальна вартість будівельного проекту кожного об'єкта.
4. Обчислити для кожної з N груп відсоток студентів, що склали іспити, якщо відомо загальну кількість студентів групи; кількість студентів, що склали іспити без задовільних оцінок; кількість



студентів, що склали іспити із задовільними оцінками в кожній групі.

5. Обчислити для кожної з N груп відсоток студентів, що не склали іспити, якщо відомо загальну кількість студентів; кількість студентів, що склали іспити без задовільних оцінок; кількість студентів, що склали іспити із задовільними оцінками в кожній групі.

6. Обчислити кількість жителів кожного з N населених пунктів на кінець року, якщо відомо кількість жителів на початку року, кількість прибулих, вибулих, народжених та померлих кожного населеного пункту.

7. Обчислити рівень виконання навчального навантаження (у відсотках) кожним з N вчителів школи, якщо відомо заплановану та фактично виконану кількість годин кожного вчителя.

8. Обчислити вартість товару кожного з N видів, якщо задано ціну одиниці товару та кількість товару кожного виду.

9. Обчислити вартість товару та вартість товару із врахуванням торгової знижки кожного з N видів товару, якщо задано ціну одиниці товару (грн.), кількість товару (шт.) та торгова знижка (%) кожного виду товару.

10. Обчислити суму оплати комунальних послуг кожного з N об'єктів, якщо задано місячну вартість комунальних послуг та кількість місяців, за які потрібно провести розрахунок для кожного об'єкта.

11. Обчислити урожайність (ц/га) зернових кожного з N фермерських господарств, якщо задано площу посівів (га) та валовий збір (т) кожного фермерського господарства.

12. Обчислити для кожного з N об'єктів суму оплати за водопостачання за звітний період, якщо задано вартість 1 м^3 води та початкові і поточні покази лічильника кожного об'єкта.



13. Обчислити для кожного з N клієнтів суму оплати за електроенергію за звітний період, якщо задано вартість 1 кВт-год та початкові і поточні покази лічильника кожного клієнта.

14. Обчислити для кожного з N клієнтів вартість пального (грн), якщо задано ціну 1 л (грн) та витрати пального (л) кожним клієнтом.

15. Обчислити для кожного з N об'єктів суму оплати за опалення, якщо задано вартість опалення 1 м² та опалювальна площа кожного об'єкта.

VII. Зберегти проект у папці Lab4_z5. Продемонструвати створений проект викладачеві. Записати отримані результати.

VIII. Захистити виконану роботу.

Контрольні запитання

1. Дайте визначення циклу.
2. Що необхідно знати для організації циклу?
3. Синтаксис оператора **FOR...NEXT** та його використання у Visual Basic.
4. Дати визначення "початкового значення", "кінцевого значення", "кроку" параметра циклу.
5. Синтаксис операторів **DO WHILE...LOOP** та **DO...LOOP WHILE** і їх використання у Visual Basic.
6. Синтаксис операторів **DO UNTIL...LOOP** та **DO...LOOP UNTIL** і їх використання у Visual Basic.
7. Вкажіть відмінності між циклом з умовою і циклом з параметром.
8. Призначення команди **EXIT**.
9. Метод **Print**, його синтаксис та використання.
10. Функція **Tab** та її застосування.
11. Функція **Spс** та її застосування.
12. Функція **Format**, її синтаксис та застосування.



Лабораторна робота №5. Розв'язування задач економічного змісту з використанням одновимірних масивів

Мета: Навчитись використовувати засоби Visual Basic для розв'язування задач економічного характеру з використанням одновимірних масивів.

Поняття масиву. Правила запису індексованої змінної

У попередніх лабораторних роботах при розв'язуванні задач ми обмежувались використанням простих типів даних. Тут кожна змінна являла собою одне значення. Використання лише простих даних суттєво обмежує можливості програмування. Тому у Visual Basic, як і в інших алгоритмічних мовах, введені структуровані типи змінних, які можуть набувати не одне значення, а сукупність значень. Найпоширенішим структурованим типом даних є масив.

Масивом називають впорядковану скінченну сукупність однотипних елементів з одним іменем. Наприклад, рядок тексту можна подати у вигляді масиву символів, вектор – у вигляді масиву чисел, а матрицю – у вигляді масиву векторів.

За визначенням усі елементи масиву мають однаковий тип. Однак бувають винятки. Зокрема, коли тип даних масиву Variant, то окремі елементи можуть містити дані різних типів (об'єкти, рядки, числа).

Кількість різних індексів визначає *вимірність* масиву. Масиви можуть бути одновимірними (вектори), двовимірними (матриці), тривимірними і т.д.

Нижні та верхні межі зміни індексу масиву визначають кількість його елементів і називаються **граничною парою**. Елементи масиву послідовно розташовуються в заданих межах.

У мові Visual Basic розрізняють статичні (з фіксованими межами) та динамічні масиви.

Межі *статичного* масиву встановлюються на етапі розроблення і можуть змінюватися тільки у новій версії програми.

Динамічні масиви змінюють свої межі в ході виконання програми. Розмір масиву можна задавати відповідно до конкретних умов. Однак потрібно враховувати, що робота з динамічними масивами потребує додаткових засобів програмування.

Для звертання до окремих елементів масиву використовується індексована змінна – змінна, значенням якої є елемент масиву.

Правила запису індексованої змінної у мові Visual Basic

Синтаксис запису *індексованої змінної* для одновимірною масиву:

Ім'яЗмінної (<індекс>)

Ім'я змінної формується за правилами, встановленими для імені простої змінної.

Індекс – це ціле невід'ємне число або змінна, значенням якої також має бути ціле невід'ємне число. Воно вказує на номер елемента масиву в списку (нумерація починається з нуля).

Для *багатовимірних* масивів запис *індексованої змінної* має вигляд:

Ім'яЗмінної (<список індексів>)

Список індексів – це кілька індексів, розділених комою. У випадку двовимірною масиву перший індекс – це номер рядка таблиці, а другий – номер її стовпця. У тривимірному масиві, крім перерахованих двох індексів, додається третій індекс – номер самої таблиці.

Масив фіксованої довжини оголошується так само, як і змінна. Єдина відмінність – використання круглих дужок після імені масиву, в яких вказуються діапазони зміни значень індексів.

Наприклад:

Dim A (1 to 15) As Integer

Dim B (11 to 20) As Double

Dim C (1 to 10, 1 to 10, 1 to 10) As Integer

Індекс масиву **A** змінюється від 1 до 15, масиву **B** – від 11 до 20, а всі три індекси масиву **C** змінюються від 1 до 10.



Часто замість діапазону в дужках записується ціле число, яке вказує на максимальне значення індексу.

Наприклад:

```
Dim Counters (19) As Integer '20 елементів
```

Для встановлення нумерації індексів за кожним виміром з 1 на початку процедури (або в стандартному модулі) використовують оператор **Option Base 1**, який змінює початок нумерації індексів кожного виміру з 0 на 1.

Наприклад:

```
(General) (Declaration)  
Option Base 1  
Dim A (4) As String
```

Завдання для виконання лабораторної роботи № 5

I. Завантажити середовище візуального програмування Visual Basic.

II. а) Розглянути зразок програмного коду (кнопка **ВИКОНАТИ**) та форми (рис. 5.1) для задачі визначення витрат на ремонт приміщення протягом 6 місяців, якщо задані витрати на ремонт приміщення кожного місяця.

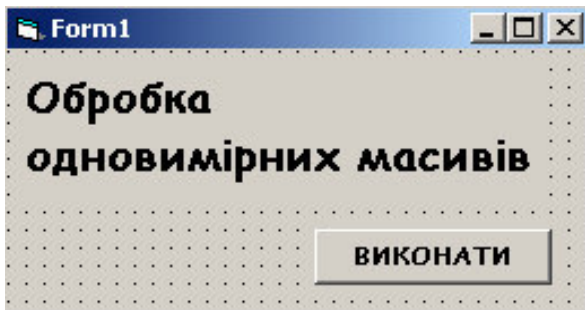
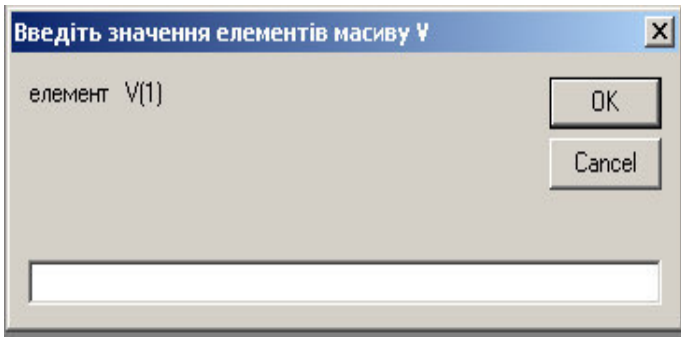


Рис. 5.1. Зразок розташування об'єктів на формі

Використання функції **InputBox** (рис. 5.2) забезпечує формування вікон для введення інформації щомісячних витрат на



ремонт з відображенням заголовка **Введіть значення елементів масиву V** і запрошенням користувача до введення інформації:



.....

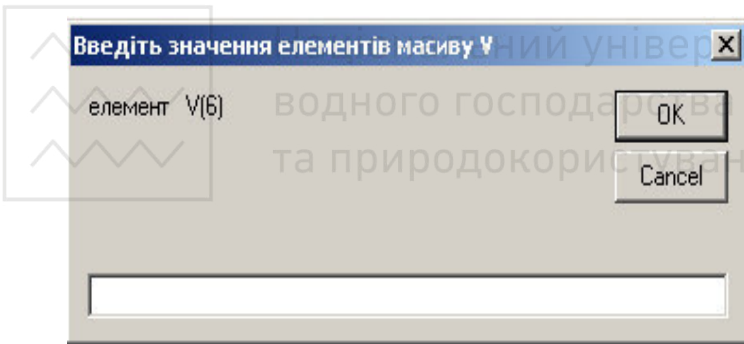


Рис. 5.2. Вікно введення елементів масиву

Один із можливих варіантів програмного коду для визначення витрат на ремонт приміщення можна подати так:

```
Private Sub Command1_Click()  
Const n% = 6  
Dim Sum As Single  
Dim V(1 To n%) As Single  
Debug.Print " Масив витрат"  
For i% = 1 To n%  
Title$ = "Введіть значення елементів масиву V"
```



```
Prt$ = "елемент V(" & i% & ")"  
V(i%) = Val(InputBox(Prt$, Title$))  
Debug.Print V(i%); " ";  
Next i%  
Debug.Print  
Sum = 0  
For i% = 1 To n%  
Sum = Sum + V(i%)  
Next i%  
Debug.Print "Витрати за півріччя="; _  
Format(Sum, "####.000")  
End Sub
```

Результати виконання наведеного програмного коду мають наступний вигляд (рис. 5.3):

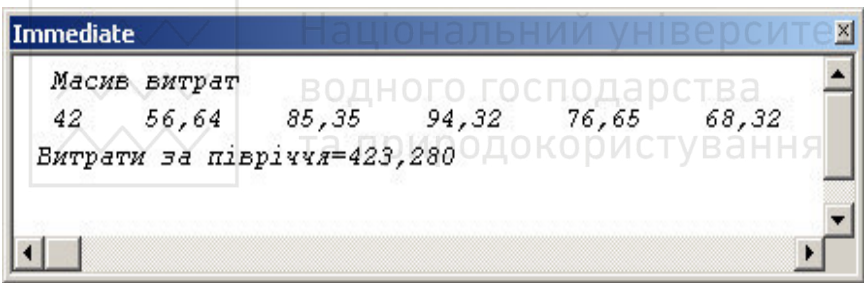


Рис. 5.3. Результати роботи програми

б) Розглянути зразок форми програмного коду (**кнопка ВИКОНАТИ**) на рис. 5.4 для задачі виведення спочатку додатних, а потім від'ємних елементів масиву C(9).

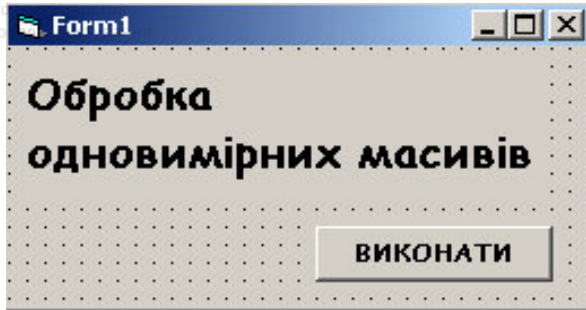


Рис. 5.4. Зразок розташування об'єктів на формі

Можна запропонувати наступний варіант програмного коду для виведення додатних, а потім від'ємних елементів масиву C(9):

```
Private Sub Command1_Click()  
Const n% = 9  
Dim Sum As Single  
Dim C(1 To n%) As Single  
Debug.Print " Заданий масив C"  
For i% = 1 To n%  
Title$ = "Введіть значення елементів масиву C"  
Prt$ = "елемент C(" & i% & ")"  
C(i%) = Val(InputBox(Prt$, Title$))  
Debug.Print C(i%); " ";  
Next i%  
Debug.Print  
Debug.Print " Додатні елементи масиву C"  
For i% = 1 To n%  
If C(i%) > 0 Then Debug.Print C(i%); " ";  
Next i%  
Debug.Print  
Debug.Print " Від'ємні елементи масиву C"  
For i% = 1 To n%  
If C(i%) < 0 Then Debug.Print C(i%); " ";  
Next i%  
Debug.Print  
End Sub
```



На рис. 5.5. проілюстровано результат роботи програми.

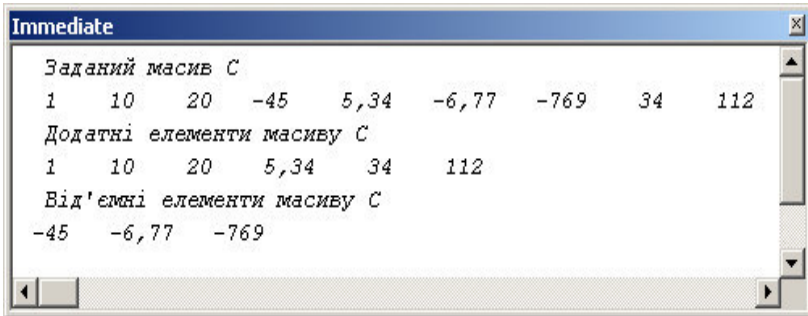


Рис. 5.5. Результати роботи програми

в) Розглянути зразок форми програмного коду (**кнопка ВИЗНАЧИТИ ПЕРЕМОЖЦЯ**) на рис. 5.4 та результатів задачі на рис. 5.5 визначення команди-переможця на кожному з етапів конкурсу “Ерудит” і на конкурсі загалом, якщо в масиві $K1(7)=(8, 9, 10, 8, 7, 9, 8)$ записані бали команди Диваки, а в масиві $K2(7)=(7, 9, 8, 10, 10, 9, 7)$ записані бали команди Говоруни.

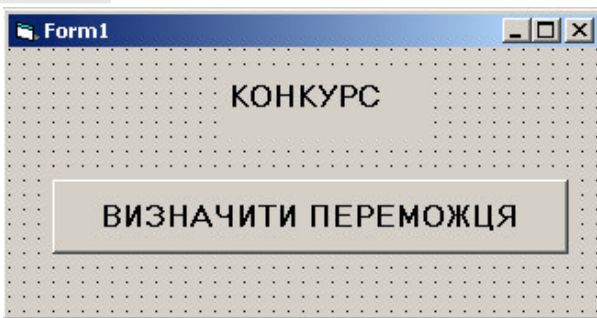


Рис. 5.6. Зразок форми

```
Private Sub Command1_Click()
Const n% = 7
Dim SumK1, SumK2 As Single
Dim K1(1 To n%), K2(1 To n%) As Single
Debug.Print " Бали команди Диваки"
```




```
SumK1 = 0
For i% = 1 To n%
Title$ = "Введіть бали команди Диваки"
Prt$ = "K1(" & i% & ")"
K1(i%) = Val(InputBox(Prt$, Title$))
SumK1 = SumK1 + K1(i%)
Debug.Print K1(i%); " ";
Next i%
Debug.Print
Debug.Print " Бали команди Говоруни"
SumK2 = 0
For i% = 1 To n%
Title$ = "Введіть бали команди Говоруни"
Prt$ = "K2(" & i% & ")"
K2(i%) = Val(InputBox(Prt$, Title$))
SumK2 = SumK2 + K2(i%)
Debug.Print K2(i%); " ";
Next i%
Debug.Print
For i% = 1 To n%
If K1(i%) > K2(i%) Then _
Debug.Print "Переможець "; i%; " -го етапу - Диваки"
If K2(i%) > K1(i%) Then _
Debug.Print "Переможець "; i%; " -го етапу - Говоруни"
If K1(i%) = K2(i%) Then _
Debug.Print "На "; i%; " -му етапі конкурсу - нічия"
Next i%
Debug.Print "Сума балів Диваків = "; _
Format (SumK1, "00.000")
Debug.Print "Сума балів Говорунів = "; _
Format(SumK2, "00.000")
If SumK1 > SumK2 Then _
Debug.Print "Переможець конкурсу - команда Диваки"
If SumK2 > SumK1 Then _
Debug.Print "Переможець конкурсу - команда Говоруни"
If SumK1 = SumK2 Then _
Debug.Print "Результат конкурсу - нічия"
End Sub
```

Immediate

```

Бали команди Диваки
8   9   10  8   7   9   8
Бали команди Говоруни
7   9   8   10  10  9   7
Переможець 1 -го етапу - Диваки
На 2 -му етапі конкурсу - нічия
Переможець 3 -го етапу - Диваки
Переможець 4 -го етапу - Говоруни
Переможець 5 -го етапу - Говоруни
На 6 -му етапі конкурсу - нічия
Переможець 7 -го етапу - Диваки
Сума балів Диваків = 59,000
Сума балів Говорунів = 60,000
Переможець конкурсу - команда Говоруни
    
```

Рис. 5.7. Результати роботи програми

III. Спроекувати форму, записати алгоритм розв'язку та програмний код для виконання наступних завдань:

1. В масиві $A(8)=(345.8, 625.5, 872.2, 824.6, 283.5, 1120.0, 346.1, 746.0)$ записані дані про поквартальні прибутки (тис. грн.) підприємства "Веселка" за два останні роки. Створити масив $B(4)$, в який записати дані за останній рік. Знайти суму прибутків за обидва роки і за кожен рік окремо.
2. В масиві $P(12)=(769.50, 769.50, 782.35, 782.35, 782.35, 782.35, 790.70, 790.70, 812.00, 815.50, 815.50, 820.00)$ записані дані про суми пенсії, яку отримав пенсіонер А. Г. Петренко протягом минулого року. Визначити, в яких місяцях проводили перерахунок пенсії і на яку суму вона збільшувалась.
3. В масиві $PR(12)=(12345, 1800, -350, -200, 740, 4470, 10390, 15200, 6320, -500, -865, 2250)$ записані суми прибутків (збитків)



фірми “Сонечко” за минулий рік. Фірма щомісяця, коли отримувала прибутки, відраховувала 18% до спец. фонду, а в ті місяці, які були збитковими – не перераховувала кошти до спец. фонду. Створити масив $V(12)$, в який записати суми перерахованих фірмою “Сонечко” коштів до спец. фонду. Обчислити загальну суму перерахованих коштів за рік.

4. В масиві $Z(12)=(1800, 2200, 1950, 1800, 1850, 2000, 2320, 2400, 2530, 2120, 1900, 1550)$ записані суми нарахованої заробітної плати муляра С. Іваненка за минулий рік. В масиві $V(12)=(234.20, 315.08, 290.00, 234.20, 240.00, 300.50, 348.70, 360.00, 309.00, 285.00, 245.00, 167.20)$ записані суми відрахувань від його заробітної плати. Створити масив $O(12)$, в який записати суми отриманої С. Іваненком зарплати (різниця між нарахованими і відрахованими сумами). Обчислити також середню заробітну плату муляра С. Іваненка за минулий рік.
5. В масиві $R(6)=(23, 12, 45, 37, 15, 26)$ записані рейтингові бали 6 цехів заводу “АвтоШИК”, які вони набрали на конкурсі “ФАХцех”. Визначити номери цехів, які набрали максимальну і мінімальну кількість балів, а також вивести рейтинг цехів (за набраними балами у порядку спадання).
6. В масиві $P(13)=(10200, 10275, 10360, 10420, 10500, 10575, 10675, 10720, 10795, 10905, 10990, 11030, 11105)$ записані показники лічильника електроенергії сім’ї Степаненків за минулий рік. Створити масив $E(12)$, в який записати кількість електроенергії, яку використовували Степаненки щомісяця. Визначити найбільшу кількість електроенергії, яку використала сім’я, а також загальну кількість електроенергії за рік.
7. Для обчислення суми пенсії потрібно визначити коефіцієнт – відношення заробітної плати людини до середньої заробітної плати. В масиві $Z(12)=(1200, 1200, 1230, 1245, 1245, 1245, 1317, 1317, 1317, 1300, 1300, 1415)$ записані суми заробітної плати пенсіонерки Л. Соколенко. В масиві $S(12)=(940, 945, 960, 980, 1000, 1000, 1007, 1100, 1150, 1200, 1290, 1300)$ записані середні заробітні плати. Створити масив $K(12)$, в який записати



коефіцієнти пані Соколенко. Визначити середній коефіцієнт (K_c) і обчислити суму пенсії за формулою $P=980 \cdot K_c \cdot 1.25 \cdot (CT-20)$, де CT – стаж роботи, який у пенсіонерки Л. Соколенко становить 38 років.

8. В масиві $G(12)=(670, 620, 500, 410, 65, 30, 35, 42, 100, 370, 580, 620)$ записані дані про кількість (куб. м) природного газу, яку використав громадянин О. Дубенко за минулий рік. Створити масив $S(12)$, в якому обчислити суми сплати за використаний газ, якщо тариф становить 45 коп. за 1 куб. м. Створити також масив $SP(12)$, в якому обчислити суми сплати з урахуванням пільги Дитина війни (25%).
9. В масиві $D(12)=(12, 18, 48, 90, 120, 50, 35, 45, 87, 105, 20, 15)$ записані суми (тис. грн.) доходів фонду “Радість” за минулий рік. Фонд щомісяця сплачує податки у розмірі 25% від суми доходу. Створити масив $P(12)$, в який записати суми сплачених податків. Знайти загальну суму податків за рік. Вивести номери місяців, в які сума податку перевищувала 10 тис. грн.
10. В масиві $D(12)=(2300, 2600, 2500, 2700, 2900, 2400, 2300, 2700, 2500, 2500, 3000, 3000)$ записані дані про сукупний дохід сім’ї Лебеденків за минулий рік. В масиві $K(12)=(315, 320, 320, 200, 185, 185, 160, 150, 190, 400, 420, 450)$ записані дані про суми платежів за комунальні послуги цієї сім’ї. Створити масив $P(12)$, в який записати відсотки, які становлять платежі від доходу. Визначити, чи належить сім’ї Лебеденків субсидія і в які місяці. (Субсидія надається, якщо платежі перевищують 20% доходу).
11. В масиві $C(10)=(345.4, 562.6, -437.9, 235.0, -460.3, 239.0, -970.3, -124.5, 548.2, 546.3)$ записані сальдо на рахунках. Вивести спочатку від’ємні сальдо, потім – додатні. Знайти окремо суми від’ємних і додатних сальдо.
12. В масиві $B(15)=(63.5, 72.8, 93.2, 81.2, 75, 64, 68.7, 74, 90, 91.4, 82, 65, 67.5, 78.3, 64.5)$ записані середні бали за сесію групи студентів спеціальності АБС. Створити масив $C(15)$, в який



записати суми стипендій студентів цієї групи. Студенти, в яких середній бал не нижчий, ніж 90, отримують підвищену стипендію 600 грн.; студенти, яких середній бал не нижчий, ніж 74, отримують 530 грн.; решта – не отримують стипендію (0 грн.). Порахувати окремо кількість відмінників і кількість студентів, що навчаються добре.

13. На початок дня в банкоматі було 20 тис. грн. В масиві $X(11)=(100, 200, 50, 500, 2000, 20, 1000, 300, 500, 100, 1500)$ записані суми (грн.), які клієнти взяли з банкомату. Визначити, яка сума залишилась в банкоматі, а також порахувати кількість клієнтів, які взяли суми понад 1000 грн.
14. На початок місяця на банківському рахунку була сума 3257 грн. В масиві $B(10)=(500, -5.50, 120, 165.40, -280, -350, -700, 320, 400.70, -38.25)$ записані суми, які надходили на рахунок (додатні) або були зняті з рахунку (від'ємні). Визначити, яка сума залишилась на рахунку на кінець місяця, а також загальну суму коштів, які надійшли на рахунок і загальну суму коштів, що були зняті з рахунку.
15. В масиві $W(10)=(350, 820, 400, 410, 300, 260, 750, 600, 500, 390)$ записані суми, віднесені на різні статті витрат підприємства “Аеросвіт”. Знайти загальну суму витрат і створити масив $P(10)$, в який записати питому вагу кожної статті витрат (питома вага – це відсоток, який становить стаття витрат у відношенні до всіх витрат).

IV. Зберегти проект у папці Lab5_z1.

V. Продемонструвати створений проект викладачеві.

VI. Записати отримані результати.

VII. Захистити виконану роботу.



Контрольні запитання

1. Які типи даних називають масивами?
2. Охарактеризуйте статичні та динамічні масиви.
3. Яка змінна використовується для звертання до окремих елементів масиву?
4. Як оголошуються масиви?
5. Як визначається вимірність масивів?
6. Як визначити кількість елементів масиву?
7. Призначення функції *InputBox*.
8. Організація введення та виведення елементів масивів.





Лабораторна робота №6. Вкладені цикли. Використання двовимірних масивів при розв'язуванні задач економічного характеру

Мета: Навчитись використовувати засоби Visual Basic для розробки проектів з використанням двовимірних масивів та конструкції "вкладені цикли" при розв'язуванні задач економічного характеру.

Вкладені цикли. Приклади використання вкладених циклів.

В області дії одного циклу можуть бути організовані один або декілька інших циклів. В цьому випадку охоплюючий цикл називається зовнішнім, а вкладені в нього цикли – внутрішніми. Правила організації як зовнішнього, так і внутрішніх циклів аналогічні правилам організації простих циклів.

Параметри зовнішнього і внутрішнього циклів змінюються наступним чином: при одному значенні параметра зовнішнього циклу параметр внутрішнього циклу набуває всіх можливих значень. При організації вкладених циклів необхідно слідкувати за тим, щоб область дії внутрішнього циклу не виходила за область дії зовнішнього циклу.

Обробка елементів двовимірних масивів вимагає, зазвичай, використання вкладених циклів.

Розглянемо детальніше використання двовимірних масивів при розв'язуванні задач економічного характеру.

Приклад 1.

Місячні сальдо M рахунків за перший квартал (дебетові сальдо задані додатними величинами, кредитові – від'ємними) задані у вигляді таблиці 6.1:



Місячні сальдо за перший квартал

Січень	Лютий	Березень	
S11	S12	S13	Місячне сальдо по 1-рахунку
...	
Sm1	Sm2	Sm3	Місячне сальдо по m-рахунку

а) Необхідно визначити та вивести на екран кількість та суму всіх дебетових сальдо.

Можливий варіант розміщення об'єктів на формі показаний на рис. 6.1.

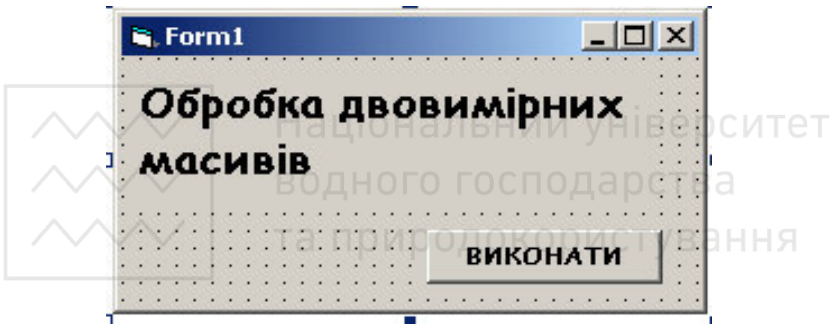


Рис. 6.1. Зразок розташування об'єктів на формі

Пропонується наступний програмний код розв'язку даної задачі:

```
Private Sub Command1_Click()  
Const m% = 4  
Const n% = 3  
Dim Sum As Single  
Dim S(1 To m%, 1 To n%) As Single  
Debug.Print " Масив місячних сальдо S"  
For i% = 1 To m%  
For j% = 1 To n%  
Title$ = "Введіть значення елементів масиву S"  
Prt$ = "елемент S(" & i% & "," & j% & ")"  
S(i%, j%) = Val(InputBox(Prt$, Title$))
```




```
Debug.Print Format(S(i%, j%), "00.00"); " ";  
Next j%  
Debug.Print  
Next i%  
Debug.Print  
Sum = 0: Kil% = 0  
For i% = 1 To m%  
For j% = 1 To n%  
If S(i%, j%) > 0 Then  
Sum = Sum + S(i%, j%)  
Kil% = Kil% + 1  
End If  
Next j%  
Next i%  
Debug.Print "Сума дебетових сальдо= "; _  
Format(Sum, "###.000")  
Debug.Print "Кількість дебетових сальдо= "; _  
Format(Kil%, "###")  
End Sub
```

Результати виконання програмного коду показані на рис. 6.2.

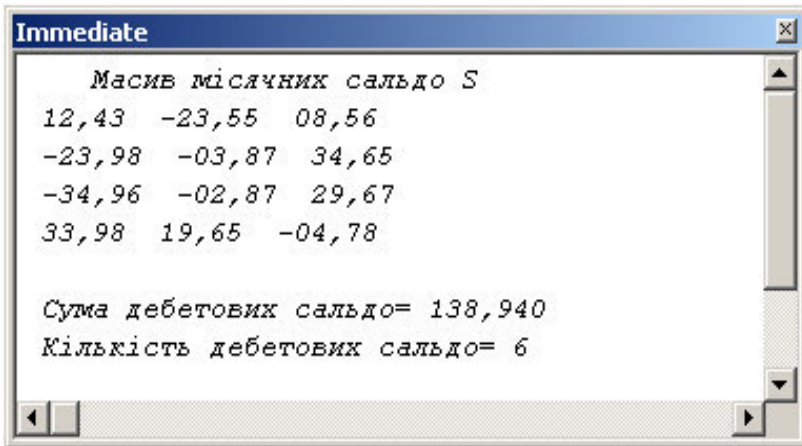


Рис. 6.2. Результати виконання програми



б) Необхідно запам'ятати в одновимірному масиві та вивести на екран всі дебетові сальдо.

Пропонується наступний програмний код розв'язку даної задачі:

```
Private Sub Command1_Click()  
Const m% = 2, n% = 3  
Dim S(1 To m%, 1 To n%) As Single  
Dim SD(1 To m% * n%) As Single  
Title$ = "Введіть елементи масиву S"  
Debug.Print "Масив сальдо S"  
For i% = 1 To m%  
For j% = 1 To n%  
Prt$ = "S(" & i% & ", " & j% & ")"  
S(i%, j%) = Val(InputBox(Prt$, Title$))  
Debug.Print Format(S(i%, j%), "00.00"); " ";  
Next j%  
Debug.Print  
Next i%  
kild% = 0  
For i% = 1 To m%  
For j% = 1 To n%  
If S(i%, j%) > 0 Then  
kild% = kild% + 1  
SD(kild%) = S(i%, j%)  
End If  
Next j%  
Next i%  
Debug.Print "Масив дебетових сальдо"  
For i% = 1 To kild%  
Debug.Print Format(SD(i%), "###.00"); " ";  
Next i%  
Debug.Print  
End Sub
```

Результати виконання програмного коду розміщені на рис. 6.3.

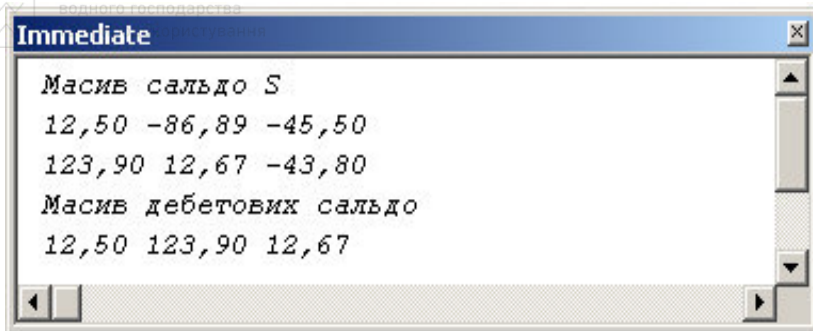


Рис. 6.3. Результати роботи програми

Приклад 2.

Зразок програмного коду задачі виведення результатів екзаменаційної сесії в табличній формі:

```
Private Sub Command1_Click()
Const n% = 2
Const m% = 4
Dim PRZV(1 To n%) As String * 20
Dim BAL(1 To n%, 1 To m%) As Integer
```

'Введення масиву прізвищ та ін.

```
For l% = 1 To n%
Title$ = "Введіть прізвище та ініціали " & l% & "- го студента"
Prt$ = "PRSV(" & l% & ")"
PRZV(l%) = InputBox(Prt$, Title$)
Next l%
```

'Введення балів студентів з іспитів

```
For l% = 1 To n%
For j% = 1 To m%
Title$ = "Введіть бали " & l% & "-го " & "студента"
Prt$ = "з " & j% & "-го іспиту "
BAL(l%, j%) = Val(InputBox(Prt$, Title$))
Next j%
Next l%
```



'Побудова горизонтальної лінії

```
For I% = 1 To 74
Debug.Print "_";
Next I%
Debug.Print
```

'Побудова шапки таблиці

```
Debug.Print "І № І";
Debug.Print " Екзаменаційна оцінка(бал) І"
Debug.Print "І з/п І Прізвище та І";
```

```
Debug.Print Tab(26);
For I% = 1 To 48
Debug.Print "_";
Next I%
Debug.Print "І"
```

```
Debug.Print "І І ініціали І";
Debug.Print "Інформатика І Вища І БЖД І Історія І"
Debug.Print "І І";
Debug.Print " І математика І Україна І"
```

```
For I% = 1 To 74
Debug.Print "_";
Next I%
Debug.Print
```

'Виведення даних таблиці

```
For I% = 1 To n%
Debug.Print Tab(3); I%; Tab(8); Format(PRZV(I%), _
"#####");
Tab(31); Format(BAL(I%, 1), "#####"); Tab(45); _
Format(BAL(I%, 2), "#####");
Tab(56); Format(BAL(I%, 3), "#####"); Tab(67); _
Format(BAL(I%, 4), "#####")
Next I%
```

```
For j% = 1 To 74
```



End Sub

Завдання для виконання лабораторної роботи №6

I. Завантажити середовище візуального програмування Visual Basic.

II. Розглянути зразок форми та програмних кодів (**кнопка ОБЧИСЛИТИ**) прикладу 1 (ст. 96) та прикладу 2 (ст. 100).

III. Використовуючи наведені зразки, побудувати алгоритм розв'язку, спроектувати форму та ввести програмний код розв'язування наведених нижче задач.

В таблиці наведені результати екзаменаційної сесії студентів групи ОіА-17

№ з/п	Прізвище, ініціали	Екзаменаційна оцінка (бал)			
		Інформатика	Вища математика	БЖД	Історія України
1	Афонова П.К.	60	60	60	63
2	Бур Д. Ж.	65	69	60	75
3	Волов К.Т.	76	74	65	81
4	Генок В.Ю.	90	85	82	100
5	Дермін Ф.З.	84	85	87	82
6	Крас К.М.	74	74	74	74
7	Лузь Ц.П.	63	60	60	68
8	Марінюк О.Т	60	64	60	72

Створити двовимірний масив $VAL(8,4)$, в який записати оцінки (бали) студентів, отримані ними на іспитах. Вивести дані з масиву VAL на екран у вигляді таблиці (див. приклад 2). Виконати обчислення згідно варіанта:



1. Знайти мінімальну і максимальну оцінки студентів групи за сесію. Обчислити середній бал кожного студента.
2. Обчислити середній бал другого за списком студента. Знайти його мінімальну і максимальну оцінки.
3. Обчислити середній бал студентів групи з інформатики.
4. Обчислити кількість оцінок “задовільно” за сесію усіх студентів групи. (Оцінка “задовільно” відповідає кількості балів в межах від 60 до 73).
5. Обчислити середній бал студентів групи з вищої математики.
6. Обчислити кількість оцінок “В” за сесію усіх студентів групи. (Оцінка “В” відповідає кількості балів в межах від 82 до 89).
7. Обчислити середній бал студентів групи з БЖД.
8. Обчислити кількість оцінок “С” за сесію усіх студентів групи. (Оцінка “С” відповідає кількості балів в межах від 74 до 81).
9. Обчислити середній бал останнього за списком студента. Знайти його мінімальну і максимальну оцінки.
10. Обчислити середній бал студентів групи з історії України.
11. Обчислити кількість оцінок “Е” за сесію усіх студентів групи. (Оцінка “Е” відповідає кількості балів в межах від 60 до 63).
12. Обчислити середній бал четвертого за списком студента. Знайти його мінімальну і максимальну оцінки.
13. Обчислити кількість оцінок “D” за сесію усіх студентів групи. (Оцінка “D” відповідає кількості балів в межах від 64 до 73).
14. Обчислити середній бал першого за списком студента. Знайти його мінімальну і максимальну оцінки.



15. Обчислити максимальний і мінімальний бали групи студентів за сесію.

IV. Зберегти проект в папці Lab6_z1.

V. Запустити проект на виконання та проаналізувати результати.

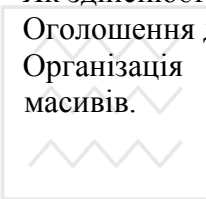
Продемонструвати створений проект викладачеві.

VI. Записати отримані результати.

VI. Захистити виконану роботу.

Контрольні запитання

1. Які цикли називають вкладеними?
2. Назвіть правила організації вкладених циклів.
3. Як здійснюється обробка елементів двовимірних масивів?
4. Оголошення двовимірних масивів.
5. Організація введення-виведення елементів двовимірних масивів.





Тестові завдання

1. Ім'я змінної може містити:
 - a. 235 символів
 - b. 125 символів
 - c. 255 символів
 - d. 155 символів

2. До числових типів даних належать:
 - a. Integer, Long, Object
 - b. Double, Integer, Long, String
 - c. Integer, Long, Date, Double
 - d. Long, Integer, Single, Double

3. Вкажіть правильне оголошення масиву:
 - a. Dim MAS (For i=1 to 10) As Integer
 - b. Dim MAS (1 to 10) As Integer
 - c. Dim MAS (i=1 to 10 step 1) As Integer
 - d. Dim MAS (i=1 , 10) As Integer

4. До математичних операцій належить:
 - a. mod + - * / \ ^
 - b. : + - * ^
 - c. + - * / ^ And
 - d. Or / + - * ^

5. Функція Val(S):
 - a. Перетворює рядок S в числовий вираз
 - b. Перетворює числовий вираз S в рядок
 - c. Повертає число символів у рядку S
 - d. Оголошує змінну S



6. Вкажіть правильний варіант застосування суфіксів:

	Тип	Суфікс
a.	Integer	&
b.	Long	%
c.	Single	\$
d.	Double	#

7. Функція Str(n):

- a. Оголошує змінну n
- b. Повертає число символів у рядку n
- c. Перетворює рядок n в числовий вираз
- d. Перетворює числове значення n в рядок

8. Логічна функція And повертає значення True, якщо:

- a. всі вирази, що беруть участь у конструкції, мають значення True
- b. всі вирази, що беруть участь у конструкції, мають значення False
- c. хоч один з виразів, що беруть участь у конструкції, має значення True
- d. хоч один з виразів, що беруть участь у конструкції, має значення False

9. Вікно функції InputBox має:

- a. 3 кнопки керування
- b. 2 кнопки керування
- c. 4 кнопки керування
- d. 5 кнопок керування



10. У методі Print для виведення інформації у потрібному форматі використовується:

- a. функція Str(n)
- b. функція Format(<вираз>, <формат>)
- c. Spc(<кількість>)
- d. функція Tab(<номер позиції>)

11. Вкажіть правильний варіант запису конструкції For:

- a. For <лічильник> = <поч.значення>То<кін.значення> _
Step <крок>
<конструкції>
Next
- b. For <лічильник > = <поч.значення> Step <крок>
<конструкції>
Next
- c. For <лічильник> = <поч.значення>То<кін.значення> _
Step <крок>
<конструкції>
End
- d. For <лічильник>=<поч.значення>То<кін.значення>
Step <крок>
<конструкції>
End For

12. Вкажіть правильний варіант застосування суфіксів:

- | | Тип | Суфікс |
|----|--------|--------|
| a. | Long | ! |
| b. | Single | # |
| c. | Double | & |
| d. | String | \$ |



13. Вкажіть правильний варіант запису конструкції Do Until:

- a. Do Until <умова>
<конструкції>
Loop
- b. Do Until <умова>
<конструкції>
End Until
- c. Do Until <умова>
<конструкції>
End
- d. Do Until <умова>
<конструкції>
End Do

14. Вказати правильний запис імені змінної:

- a. IName
- b. Ojojoj
- c. Meb1.5
- d. Object

15. Вкажіть правильний варіант застосування суфіксів:

- | | Тип | Суфікс |
|----|---------|--------|
| a. | Integer | @ |
| b. | Long | # |
| c. | Single | ! |
| d. | Double | \$ |

16. Вказати правильне оголошення константи:

- a. Dim Const n%=5
- b. Const n%:=5
- c. Const=5
- d. Const n%=5



17. Для запису декількох операторів у одному рядку використовується символ:

- a. "
- b. :
- c. ;
- d. ,

18. Для продовження запису оператора на наступний рядок використовуються:

- a. _
- b. +
- c. :
- d. ;

19. Вкажіть правильний варіант застосування суфіксів:

- | | Тип | Суфікс |
|----|----------|--------|
| a. | Single | & |
| b. | Double | \$ |
| c. | String | # |
| d. | Currency | @ |

20. Для запису коментарів використовується символ:

- a. '
- b. "
- c. +
- d. :

21. У методі Print для виведення значення із заданої позиції поточного рядка використовують:

- a. функцію Spc(<кількість пропусків >)
- b. функцію Tab(<номер позиції>)
- c. функцію Format(<вираз>, <формат>)
- d. функцію Str(n)



22. Значення True властивості MultiLine :

- a. дозволяє об'єкту працювати більше ніж з одним стовпчиком
- b. дозволяє об'єкту працювати більше ніж з одним рядком
- c. дозволяє об'єкту працювати більше ніж з одним об'єктом
- d. дозволяє об'єкту працювати більше ніж з одним проектом

23. Написи у формі можна розміщувати за допомогою елемента:

- a. Text
- b. Form
- c. CommandButton
- d. Label

24. Вказати команду переривання циклу до його повного завершення:

- a. End
- b. Close
- c. Command
- d. Exit

25. Вкажіть правильний варіант застосування суфіксів:

- | | Тип | Суфікс |
|----|---------|--------|
| a. | Integer | % |
| b. | Long | ! |
| c. | Single | & |
| d. | Double | \$ |



26. У методі Print для виведення заданої кількості пропусків використовують :

- a. функцію Str(n)
- b. функцію Tab(<номер позиції>)
- c. функцію Spc(<кількість>)
- d. функцію Format(<вираз>, <формат>)

27. Логічна функція Or повертає значення False, якщо:

- a. хоч один з виразів, що беруть участь у конструкції, має значення False
- b. хоч один з виразів, що беруть участь у конструкції, має значення True
- c. всі вирази, що беруть участь у конструкції, мають значення False
- d. всі вирази, що беруть участь у конструкції, мають значення True

28. Вкажіть правильний варіант запису конструкції If :

- a. If <умова> Then <послідовність операторів1>
Else
<послідовність операторів2>
End If
- b. If <умова> Then <послідовність операторів1> Else
<послідовність операторів2>
If End
- c. If <умова> Then
<послідовність операторів1>
Else
<послідовність операторів2>
End If
- d. If <умова> Then <послідовність операторів1>
Else
<послідовність операторів2>
EndIf



29. Вкажіть правильний варіант запису конструкції Do While:

- a. Do While <умова>
<конструкції>
Loop
- b. Do While <умова>
<конструкції>
End While
- c. Do While <умова>
<конструкції>
End
- d. Do While <умова>
<конструкції>
End Do

30. Вкажіть правильний варіант застосування суфіксів:

	Тип	Суфікс
a.	Integer	!
b.	Long	&
c.	Single	%
d.	Double	\$



Правильні відповіді до тестових завдань

№	Правильна відповідь
1.	Ім'я змінної може містити 255 символів
2.	Числові типи даних : Long, Integer, Single, Double
3.	Dim MAS (1 to 10) As Integer
4.	Математичні операції: mod, +, -, *, /, \, ^
5.	Перетворює рядок S в числовий вираз
6.	Тип Double (#)
7.	Перетворює числове значення n в рядок
8.	Всі вирази, що беруть участь у конструкції, мають значення True
9.	Вікно функції InputBox має 2 кнопки керування
10.	Функція Format(<вираз>, <формат>)
11.	For<лічильник>=<поч.значення>To<кін.значення>Step<крок> <конструкції> Next
12.	String \$
13.	Do Until <умова> <конструкції> Loop
14.	Правильне ім'я змінної Ojojoj
15.	Тип Single (!)
16.	Const n%=5
17.	Двокрапка (:)
18.	Підкреслення (_)
19.	Тип Currency (@)
20.	Апостроф (')
21.	Функцію Tab(<номер позиції>)
22.	Дозволяє об'єкту працювати більше ніж з одним рядком
23.	Написи у формі можна розміщувати за допомогою елемента Label
24.	Команда переривання циклу до його повного завершення - Exit
25.	Тип Integer (%)
26.	Функцію Spc(<кількість>)



27	Всі вирази, що беруть участь у конструкції, мають значення False
28	If <умова> Then <послідовність операторів1> Else <послідовність операторів2> End If
29	Do While <умова> <конструкції> Loop
30	Тип Long (&)





Література

1. Ананьев Александр. Самоучитель Visual Basic 6.0 / Александр Ананьев, Аркадий Федоров. – СПб : БХВ–Петербург, 2005. – 624 с. : ил. – ISBN 5-8206-0060-6.
2. Баженов В. А. Информатика. Комп'ютерна техніка. Комп'ютерні технології : підручник / Баженов Віктор Андрійович, Венгерський Петро Сергійович, Горлач Віталій Михайлович та ін. – К. : Каравела, 2004. – 464 с. – ISBN 966-8019-05-9.
3. Беспалов В.М. Информатика для економістів : навч. посіб. для студ. ВНЗ / В.М. Беспалов, А.Ю. Вакула, А.М. Гострик та ін. – К. : ЦУЛ, 2003. – 788 с.
4. Браун С. Visual Basic 6: учебный курс: пер. с англ. / С. Браун.– С-Пб.: “Питер”, 1999. – 576 с.
5. Гетц Кен. Программирование на Visual Basic 6 и VBA. Руководство разработчика: пер. с англ. / К. Гетц, М. Гилберт. – К.: Издательская группа BHV, 2001. – 912 с.
6. Глинський Ярослав Миколайович. Практикум з інформатики : навч. посіб. / Я. М. Глинський. – Львів : СПД Глинський, 2007. – 296 с. – ISBN 978-966-2934-00-7.
7. Глушаков Сергей Владимирович. Программирование на Visual Basic 6.0 : учебный курс / С. В. Глушаков, А. С. Сурядний – Харьков : Фолио, 2004. – 497 с. – ISBN 966-03-2437-5.
8. Дженнингс Роджер. Руководство разработчика баз данных на Visual Basic 6: пер. с англ. / Р. Дженингс. – М.: Издательский дом "Вильямс", 1999 – 976 с.
9. Зубик Людмила Володимирівна. Информатика та комп'ютерна техніка у водному господарстві : навч. посіб. / Людмила Володимирівна Зубик, Ярослав Ярославович Зубик, Іван Миколайович Карпович. – Рівне : НУВГП, 2008. – 306 с. – ISBN 978-966-327-103-3.
10. Информатика: Комп'ютерна техніка. Комп'ютерні технології : посібник / Браткевич В. В., Бутов М. В., Золотарьова І. О. та ін. ; за ред. О. І. Пушкаря. – К. : Академія, 2002. – 704 с. – ISBN 966-580-085-X.



11. Кузьменко В. Г. Базы данных в Visual Basic и VBA: Самоучитель. – М. : ООО “Бином-Пресс”, 2004. – 416 с.
12. Леонтьев Виталий Петрович. Новейшая энциклопедия персонального компьютера 2003 / В. П. Леонтьев. – М. : ОЛМА-ПРЕСС, 2003. – 957 с.: ил. – ISBN 5-224-04035-3.
13. Лук'янова Валентина В'ячеславівна. Комп'ютерний аналіз даних : посібник / В. В. Лук'янова. – К. : Академія, 2003. – 344 с. – ISBN 966-580-160-0.
14. Мамченко Світлана Дмитрівна. Економічна інформатика. Практикум : навч. посіб. / С. Д. Мамченко, В. А. Одинець. – К. : Знання, 2008. – 710 с. – ISBN 978-966-346-437-4.
15. Пройдаков Е. М. Англо-український тлумачний словник з обчислювальної техніки, Інтернету і програмування / Е. М. Пройдаков, Л. А. Теплицький. – Київ : СофтПрес, 2006. – 824 с. – ISBN 966-530-070-9.
16. Рзаев Д. Інформатика та комп'ютерна техніка : навчально-методичний посібник для самостійного вивчення дисципліни / Д. Рзаев, О. Шарапов, В. Ігнатенко, Л. Дибкова. К. : КНЕУ, 2006. – 486 с. – ISBN 966-574-458-5.
17. Сайлер Брайан. Использование Visual Basic 6.0. / Издательский Дом “Вильямс”, 1999 г. – 832 с.
18. Толковый словарь по вычислительным системам / Под ред. В. Иллингурта и др. : Пер. с англ. А. К. Белоцкого и др. – М. : Машиностроение, 1991. – 560 с. : ил. – ISBN 5-217-00617-X(СССР).
19. Эпплман Дан. Win32 API и Visual Basic. Для профессионалов: пер. с англ. / Д. Эпплман. – СПб.: “Питер”, 2001. – 1120 с.
20. Методичні вказівки та завдання до виконання лабораторних робіт з розділу “Програмування мовою Паскаль” для студентів спеціальностей “Облік і аудит”, “Економіка підприємства”. 044-103 / О. М. Гладка, Л. Д. Бачишина. – Рівне: РДТУ, 1999. – 28 с.
21. Методичні вказівки для виконання лабораторних та самостійних робіт з дисциплін "Обчислювальна техніка та програмування", "Техніка користування ЕОМ" для студентів 1 курсу напрямків "Будівництво", "Інженерна механіка", "Водні ресурси" денної та заочної форми навчання. 044-145 / Т. В. Лежнюк. – Рівне: НУВГП, 2004. – 32 с.



22. Методичні вказівки та завдання для виконання лабораторних робіт з розділу “VISUAL BASIC 6.0. Частина 1” студентами 1 курсу всіх спеціальностей денної та заочної форм навчання НУВГП. 044-155 / В. О. Савич, Ю. О. Кравчук. – Рівне: НУВГП, 2008. – 32 с.
23. Методичні вказівки та завдання для виконання лабораторних робіт з розділу “VISUAL BASIC 6.0. Частина 2” студентами 1 курсу всіх спеціальностей денної та заочної форм навчання НУВГП. 044-156 / В. О. Савич, Ю. О. Кравчук. – Рівне: НУВГП, 2008. – 36 с.
24. Методичні вказівки та завдання для виконання лабораторних робіт з розділу “Visual Basic 6.0. Частина 3” студентами 1 курсу всіх спеціальностей денної та заочної форм навчання НУВГП. 044-157 / В. О. Савич, О. М. Гладка, Ю. О. Кравчук. – Рівне: НУВГП, 2009. – 24 с.
25. <http://www.knigka.org.ua>
26. <http://www.vbcode.com/>
27. <http://visualbasic-6-0.narod.ru>
28. <http://visualprogs.medyal.ru>
29. <http://www.vbrussian.com/>
30. <http://msdn.microsoft.com>
31. <http://codevb.narod.ru/>
32. <http://vbrus.narod.ru/>
33. <http://vba-help.ru>



Національний університет
водного господарства
та природокористування

Навчальне видання

***Карпович Іван Миколайович
Савич Віра Олексіївна
Шепетько Юлія Олександрівна***

ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ VISUAL BASIC. ПРАКТИКУМ

Навчальний посібник



Друкується в авторській редакції

Національний університет
водного господарства
та природокористування

Підписано до друку _____ Формат 6084
Папір друкарський №1. Гарнітура Times. Друк різнографічний.
Ум. – друк. арк. _____. Обл. – вид. арк. ____
Тираж 150 прим. Зам № _____

*Редакційно-видавничий центр
Національного університету
Водного господарства та природокористування
33028, Рівне, вул. Соборна, 11*

*Свідоцтво про внесення суб'єкта видавничої справи до державного
реєстру видавців, виготівників і розповсюджувачів видавничої
продукції РВ №31 від 26.04.2005*