

Б81.3.06(075)

п 36

Н.В. Пішеніна  
В. О. Пішенін

**ОСНОВИ  
ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
ТА ПРОГРАМУВАННЯ**

Часопис



**Міністерство освіти і науки України  
Вінницький національний технічний університет**

**Н.В. Пішеніна, В.О. Пішенін**

**ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ПРОГРАМУВАННЯ**

**Частина 1**

**ЛАБОРАТОРНИЙ ПРАКТИКУМ**

Затверджено Вченою радою Вінницького національного технічного університету як навчальний посібник для студентів денної та заочної форм навчання напряму підготовки 6.0905 "Енергетика" та спеціальності "Теплоенергетика". Протокол № 2 від 25 вересня 2003 р.

Вінниця ВНТУ 2003

УДК 681.3  
ББК 32.97  
П 36

Р е ц е н з е н т и :

**М.А. Філінюк**, доктор технічних наук, професор  
**С.В. Юхимчук**, доктор технічних наук, професор  
**В.П. Якубович**, кандидат технічних наук, доцент

Рєкомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України

**П 36 Пішеніна Н.В., Пішенін В.О.**

**Основи інформаційних технологій та програмування. Частина 1.**  
Лабораторний практикум. — Вінниця: ВНТУ, 2003. — 92 с.

У лабораторному практикумі розглянуто питання вивчення основ комп'ютерних технологій та набуття навичок роботи з сучасним програмним забезпеченням персональних комп'ютерів, яке використовується в інженерній та науковій практиці.

Зміст практикуму відповідає програмі дисципліни "Основи інформаційних технологій та програмування", основною шляхом якої є поглиблена теоретична підготовка студентів енергетиків з питань використання електронної обчислювальної техніки в інженерних розрахунках під час проектування, налагодження і експлуатації енергетичного обладнання промислових та побутових об'єктів і застосування набутих знань у вивченні курсів дисциплін за фахом.

УДК 681.3  
ББК 32.97

## ЗМІСТ

ВСТУП .....	8
1 ОПЕРАЦІЙНА СИСТЕМА ТА СТАНДАРТНІ ПРОГРАМИ .....	9
1.1 ЛАБОРАТОРНА РОБОТА №1.1 .....	9
1.1.1 Тема: Операційні системи. Найпростіші операції з піктограмами та вікнами в операційній системі Windows 9X або Windows 2000 .....	9
1.1.2 Мета: Познакомитися з операційною системою. Відпрацювати прийоми роботи з манипулятором "Мишка". Вивчити прийоми роботи з об'єктами (робота з файловою структурою у програмі "Мой компьютер").	9
1.1.3 Теоретичні відомості .....	9
1.1.4 Послідовність виконання роботи: .....	12
1.1.5 Контрольні запитання .....	14
1.2 ЛАБОРАТОРНА РОБОТА №1.2 .....	16
1.2.1 Тема: Робота зі стандартними програмами у Windows 9X .....	16
1.2.2 Мета: опанувати основні принципи роботи зі стандартними програмами Windows. Ознайомитися з призначенням та можливостями цих програм. ....	16
1.2.3 Теоретичні відомості .....	16
1.2.4 Хід роботи .....	17
1.2.5 Контрольні запитання .....	19
1.3 ЛАБОРАТОРНА РОБОТА №1.3 .....	20
1.3.1 Тема: текстовий редактор WordPad .....	20
1.3.2 Мета: опанувати основні принципи роботи з стандартним текстовим редактором .....	20
1.3.3 Теоретичні відомості .....	20
1.3.4 Хід роботи .....	20
1.3.5 Контрольні запитання .....	22
1.4 ЛАБОРАТОРНА РОБОТА №1.4 .....	23
1.4.1 Тема: Робота з програмою-архіватором .....	23
1.4.2 Мета: опанувати основні принципи роботи з стандартним архіватором .....	23
1.4.3 Теоретичні відомості .....	23
1.4.4 Хід роботи .....	24
1.4.5 Контрольні запитання .....	25
1.5 ЛАБОРАТОРНА РОБОТА №1.5 .....	27
1.5.1 Тема: OpenOffice Word, графічні можливості .....	27



1.5.2 Мета: навчитися працювати в графічному редакторі OpenOffice Word.....	27
1.5.3 Теоретичні відомості.....	27
1.5.4 Хід роботи.....	29
1.5.5 Контрольні запитання.....	30
<b>1.6 ЛАБОРАТОРНА РОБОТА №1.6.....</b>	<b>31</b>
1.6.1 Тема: OpenOffice. Електронна таблиця Excel.....	31
1.6.2 Мета роботи: навчитись працювати в електронній таблиці, проводити типові математичні обчислення, будувати графіки та діаграми.....	31
1.6.3 Теоретичні відомості.....	31
1.6.4 Хід роботи.....	32
1.6.5 Контрольні запитання.....	34
<b>2 ОСНОВИ ПРОГРАМУВАННЯ.....</b>	<b>35</b>
<b>2.1 ЛАБОРАТОРНА РОБОТА №2.1.....</b>	<b>35</b>
2.1.1 Тема: програмування та розв'язання задач з лінійним обчислювальним процесом.....	35
2.1.2 Мета роботи: набуття практичних навичок програмування мовою Visual Basic та розв'язання лінійних задач з використанням EOM.....	35
2.1.3 Теоретичні відомості.....	35
2.1.4 Хід роботи.....	38
2.1.5 Контрольні запитання.....	39
<b>2.2 ЛАБОРАТОРНА РОБОТА №2.2.....</b>	<b>41</b>
2.2.1 Тема: Програмування та розв'язання задач з розгалуженим обчислювальним процесом.....	41
2.2.2 Мета роботи: набуття практичних навичок програмування мовою Visual Basic та розв'язання на EOM задач з розгалуженим обчислювальним процесом за допомогою оператора умовного переходу.....	41
2.2.3 Теоретичні відомості.....	41
2.2.4 Хід роботи.....	43
2.2.5 Контрольні запитання.....	44
<b>2.3 ЛАБОРАТОРНА РОБОТА №2.3.....</b>	<b>45</b>
2.3.1 Тема: розв'язання задач з розгалуженим обчислювальним процесом з використанням операторів умовного та безумовного переходу.....	45
2.3.2 Мета роботи: набуття практичних навичок програмування мовою Visual Basic та розв'язання на EOM задач з використанням операторів умовного та безумовного переходу.....	45
2.3.3 Теоретичні відомості.....	45

2.3.4 Хід роботи.....	47
2.3.5 Контрольні запитання.....	47
2.4 ЛАБОРАТОРНА РОБОТА №2.4.....	49
2.4.1 Тема: розв'язання задач з циклічним обчислювальним процесом Оператори IF... THEN, FOR... NEXT.....	49
2.4.2 Мета роботи: набуття практичних навичок програмування мовою Visual Basic та розв'язання на EOM задач з циклічним процесом.....	49
2.4.3 Теоретичні відомості.....	49
2.4.4 Хід роботи.....	52
2.4.5 Контрольні запитання.....	53
2.5 ЛАБОРАТОРНА РОБОТА №2.5.....	54
2.5.1 Тема: Розв'язання задач з циклічним обчислювальним процесом Оператори DO... LOOP.....	54
2.5.2 Мета роботи: набуття практичних навичок програмування мовою Visual Basic та розв'язання на EOM задач з циклічним процесом.....	54
2.5.3 Теоретичні відомості.....	54
2.5.4 Хід роботи.....	56
2.5.5 Контрольні запитання.....	56
2.6 ЛАБОРАТОРНА РОБОТА №2.6.....	58
2.6.1 Тема: Розв'язання задач з повторюваними однотипними обчислювальними процедурами.....	58
2.6.2 Мета роботи: набуття практичних навичок програмування мовою Visual Basic зі створенням та використанням підпрограм користувача.....	58
2.6.3 Теоретичні відомості.....	58
2.6.4 Хід роботи.....	59
2.6.5 Контрольні запитання.....	59
2.7 ЛАБОРАТОРНА РОБОТА №2.7.....	61
2.7.1 Тема: Розв'язання задач з повторюваними однотипними обчислювальними процедурами.....	61
2.7.2 Мета роботи: набуття практичних навичок програмування мовою Visual Basic зі створенням та використанням функцій.....	61
2.7.3 Теоретичні відомості.....	61
2.7.4 Хід роботи.....	62
2.7.5 Контрольні запитання.....	62
2.8 ЛАБОРАТОРНА РОБОТА №2.8.....	63
2.8.1 Тема: Розв'язання задач з повторюваними однотипними обчислювальними процедурами.....	63
2.8.2 Мета роботи: набуття практичних навичок програмування мовою Visual Basic зі створенням та використанням підпрограм – процедур.....	63

2.8.3	Теоретичні відомості.....	63
2.8.4	Хід роботи.....	64
2.8.5	Контрольні запитання.....	65
2.9	ЛАБОРАТОРНА РОБОТА №2.9.....	66
2.9.1	Тема: Програмування з виведенням даних у файл.....	66
2.9.2	Мета роботи: набуття практичних навичок програмування мовою Visual Basic з використанням виведення результатів у файл.....	66
2.9.3	Теоретичні відомості.....	66
2.9.4	Хід роботи.....	67
2.9.5	Контрольні запитання.....	67
2.9.6	Приклади програм для роботи в Microsoft Office.....	67
	ЛІТЕРАТУРА.....	72
	ДОДАТОК А.....	73
	Приклади програм до лабораторної роботи №2.1 для виконання в середовищі Microsoft Office.....	73
	ДОДАТОК Б.....	75
	Приклади програм до лабораторної роботи №2.2 для виконання в середовищі Microsoft Office.....	75
	ДОДАТОК В.....	77
	Приклади програм до лабораторної роботи №2.3 для виконання в середовищі Microsoft Office.....	77
	ДОДАТОК Г.....	79
	Приклади програм до лабораторної роботи №2.4 для виконання в середовищі Microsoft Office.....	79
	ДОДАТОК Д.....	82
	Приклади програм до лабораторної роботи №2.5 для виконання в середовищі Microsoft Office.....	82
	ДОДАТОК Е.....	85
	Приклади програм до лабораторної роботи №2.6 для виконання в середовищі Microsoft Office.....	85
	ДОДАТОК Ж.....	87
	Приклади для виконання програм до лабораторної роботи №2.7 в Microsoft Office.....	87

<b>ДОДАТОК И</b> .....	<b>89</b>
<b>Приклади для виконання програм до лабораторної роботи №2.8 в Microsoft Office</b> .....	<b>89</b>

## ВСТУП

Даний лабораторний практикум призначений для методичного забезпечення дисципліни "Основи інформаційних технологій та програмування", що викладається студентам спеціальності "Теплоенергетика". Основним результатом успішного виконання запропонованих лабораторних робіт у першій частині практикуму є: закріплення теоретичного матеріалу відповідних розділів; набуття практичних навичок роботи з персональним комп'ютером; опанування основами роботи з сучасними операційними системами та стандартними програмами, що входять до їх складу; опанування основами програмування типових структур обчислювальних алгоритмів у середовищі Visual Basic Application.

На перших лекційних заняттях студенти отримують перелік тем лабораторних занять та список рекомендованої літератури. На першому лабораторному занятті студенти знайомляться з вимогами техніки безпеки та правилами протипожежної безпеки під час роботи з обладнанням комп'ютерних центрів університету і зобов'язаннями не порушувати встановлені правила, що підтверджуються підписами у журналі з інструктажу відповідних комп'ютерних центрів.

Студент допускається до виконання лабораторної роботи при знанні мети, змісту роботи та методики її виконання. Якщо студент недостатньо підготовлений до виконання лабораторної роботи, то він повинен підготуватися протягом 10 ... 15 хвилин безпосередньо в класі, отримуючи вказівки від викладача.

Захист лабораторної роботи відбувається протягом 15 хвилин групами по декілька студентів шляхом пред'явлення звіту з лабораторної роботи та відповідей на питання викладача безпосередньо на робочих місцях (за комп'ютерами) або поза робочими місцями за рішенням викладача.

Звіти з лабораторних робіт попередньо оформлюються в електронному варіанті, а потім за рішенням викладача роздруковуються для набуття навичок у оформленні документації з використанням комп'ютерних технологій (там, де це потрібно).

Звіт з лабораторної роботи виконується на аркушах формату А4 (210×297мм) на одному боці аркуша з дотриманням вимог стандарту ДСТУ 3008–95.

При оформленні звіту необхідно дотримуватися такої послідовності викладення:

- назва лабораторної роботи;
- мета лабораторної роботи;
- завдання роботи;
- теоретична частина;
- описання ходу виконання та отриманих результатів при цьому;
- висновки.

# 1 ОПЕРАЦІЙНА СИСТЕМА ТА СТАНДАРТНІ ПРОГРАМИ

## 1.1 ЛАБОРАТОРНА РОБОТА №1.1

**1.1.1 Тема:** Операційні системи. Найпростіші операції з піктограмами та вікнами в операційній системі Windows 9X або Windows 2000

**1.1.2 Мета:** Познайомитися з операційною системою. Відпрацювати прийоми роботи з маніпулятором "Мишка". Вивчити прийоми роботи з об'єктами (робота з файловою структурою у програмі "Мой комп'ютер").

### 1.1.3 Теоретичні відомості

"Мишка" – маніпулятор з двома чи трьома кнопками. При переміщенні її по спеціальній поверхні (килимком) на екрані комп'ютера відповідно рухається покажчик миші – стрілочка.

При необхідності виконання будь-якої команди, оператор натискає кнопку миші. **Клацнути мишкою** означає встановити стрілочку (покажчик) на будь-який об'єкт і натиснути кнопку миші, частіше всього ліву. При натисканні правої кнопки на екрані відображається контекстне меню.

**Клавіатура.** Клавіатура призначена для введення в комп'ютер інформації від користувача.

#### **Принцип дії клавіатури:**

Кожна клавіша клавіатури являє собою кришку для мініатюрного перемикача(механічного або мембранного).

Встановлений в клавіатуру мікропроцесор контролює стан цих перемикачів. При натисненні чи відтискуванні клавіші перемикач посилає в комп'ютер відповідний сигнал (переривання), який оброблюється операційною системою (або іншою програмою).

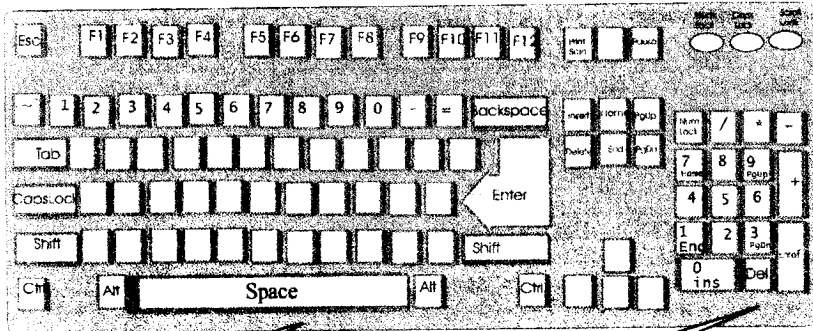
#### **Призначення клавіш:**

**Алфавітно-цифрові клавіші** (рис. 1.1) використовуються для введення алфавітно-цифрових символів. Тип символу залежить від встановленої мови та розкладки клавіатури.

**Цифрова клавіатура**— блок клавіш у правій частині клавіатури; використовується для двох цілей. В режимі блокування цифр(Num Lock) цей блок зручний для введення числової інформації та знаків арифметичних дій. Якщо режим Num Lock відключений, то ці клавіші дублюють клавіші управління курсором, а також клавіші Ins, Delete. Ввімкнення та вимкнення режиму блокування цифр здійснюється натискуванням **клавіші Num Lock.**

**Клавіша Tab** — (табулювання) при редагуванні тексту використовується для переходу до наступної позиції табулювання. В інших програмах її значення може бути іншим: переключення між полями запиту та ін.





Алфавітно-цифрові клавіші

Цифрова клавіатура

Рисунок 1.1 — Схема клавіатури

\* **Клавіші Ctrl, Alt, Shift** — (Ctrl—"керування", Alt—"додатково")призначені для зміни значень інших клавіш. При натискуванні клавіші Shift разом з алфавітно-цифровою клавішею дозволяє ввести прописну літеру або символ верхнього регістру. Багато програм дозволяють використовувати ці клавіші в комбінації з іншими("гарячі клавіші").

\* **Функціональні F1-F12** —порядок використання цих клавіш визначається конкретною програмою і операційною системою (напр.: F1 часто викликає вбудовану довідку програми).

\* **Клавіша Esc** —(Escape—тікати, спасатись)викликає відміну, переривання команди, що виконується.

\* **Клавіші управління курсором: "стрілочки", Home (початок), End (кінець), PgUp (сторінка вверх), PgDn (сторінка донизу)** — натискування на них викликає переміщення курсора (покажчика поточного місця в документі) у відповідному напрямку, або "перегортання" екранного зображення.

\* **Insert** —вставка. Використовується для вставки символів або для переключення між режимами введення символів з розсуванням (вставка) та введення з заміною існуючих символів(заміна).

\* **Delete**—видалення. Використовується для видалення вибраного фрагменту або символу справа від курсора.

\* **Enter** використовується для закінчення введення того чи іншого об'єкту. В текстових редакторах натискування цієї клавіші означає кінець введення абзацу.

\* **Backspace** —"крок назад", видаляє символ зліва від курсора.

\* **Pause/Break** використовується для зупинення (чи призупинення) роботи деяких програм.

■ **Print Scrn**—друкування екрана. Використовується для друку вмісту екрана (в деяких операційних системах) або копіювання його в буфер обміну Windows та ін.

■ **Клавiша Caps Lock** призначена для фіксації режиму великих літер.

■ **Клавiша Num Lock** призначена для ввімкнення і вимкнення режиму блокування цифр на цифровій клавіатурі.

■ **Клавiша Scroll Lock** призначена для ввімкнення і вимкнення режиму блокування прокручування (в деяких операційних системах).

■ **Індикатори режимів** вказують режим роботи клавіатури (вмикаються і вимикаються клавiшами Num Lock, Caps Lock, Scroll Lock)

На клавіатурі 104 клавiші. Основна клавіатура - з літерами, цифрами, розділовими знаками. Чорною фарбою написані літери латинського алфавіту, червоною – українського, російського.

**Файл** – це – одна програма, картинка, сторінка тексту – сукупність будь-яких даних одного типу, що зберігаються на диску окремо від інших. У нього є назва.

**Назва файлу не повинна містити таких знаків:** крапки, літери і (української), коми, "+", "=", "+", "/", "\", "?".

**Робочий стіл** – зовнішній вигляд екрана Windows. Внизу екрана можна побачити так звану панель задач, де знаходиться кнопка "Пуск", перемикач мови, годинник, та інші елементи.

**Головні програми** подані у вигляді значків на Робочому столі з надписами, що пояснюють їх призначення.

**Головне меню** відчиняється кнопкою "Пуск" і містить набір головних команд Windows. При виборі команд, відзначених стрілкою, відчиняється вкладене меню. Для виконання команди потрібно клацнути мишкою на піктограмі вибраної вами команди.

**Команда Програми** дає доступ до програм і додатків, встановлених на комп'ютері.

**Команда Документи** виводить на екран документи, які змінювалися останнім часом. Для відкриття документа варто клацнути на його значку.

**Команда Налаштування** дозволяє змінити склад установлених на комп'ютері програм.

**Команда Пошук** дозволяє робити пошук файлу, папки, знаючи назву, розмір файлу, дату останньої модифікації, основну фразу.

**Команда Виконати** дозволяє запустити будь-яку програму або відкрити якусь папку.

Для відкриття вікна або запуску програми необхідно двічі підряд клацнути на відповідному значку (або виділити мишкою та натиснути кла-

вішу **Enter**) на робочому столі. Можна також вибрати об'єкт з Головного меню, користуючись кнопкою **Пуск**.

Інтерфейс вікна містить:

- **рядок заголовка з іменем вікна** – він вказує на тип вікна або програми;

- **віконний значок** – дозволяє викликати меню вікна та контекстне меню вікна, також виконувати дії над вікном;

- **кнопки управління видом вікна** призначені для згортання вікна, перетворення його в повноекранний або віконний тип, також для закриття вікна;

- **рядок меню програми** містить всі команди даної програми. Команди, що недоступні для використання в даний момент (неактивні) забілені сірим кольором;

- **панель інструментів** містить команди що найчастіше використовуються у вигляді піктограм (кнопок), натискання на які приводить до виконання вибраної команди;

- **лінійки прокручування** (праворуч і знизу) дозволяють здійснювати переміщення по вікну вверх – вниз, вліво – вправо;

- **адресний рядок** вказує на розміщення в пам'яті комп'ютера об'єкта (будь-якого файлу), над яким відбувається дія;

- всі дії у вікні можна здійснювати за допомогою миші або клавіатури.

Завдання: Виконати операції перетворення піктограм у вікна і навпаки та операції переміщення піктограм

#### 1.1.4 Послідовність виконання роботи:

1.1.4.1 Якщо комп'ютер не увімкнено запросити викладача до робочого місця і в його присутності та з його дозволу увімкнути комп'ютер.

На екрані має з'явитись вікно (робочий стіл), на якому розміщені піктограми, а внизу — панель задач із кнопкою **Пуск**. У вікні обов'язково повинна бути наявна піктограма **Мой комп'ютер**, **Корзина** та інші (в залежності від налаштування робочого столу).

Зарисуйте для звіту з лабораторної роботи ескіз зовнішнього вигляду робочого столу з піктограмами, які є на ньому.

1.1.4.2 Почергово зробіть активними деякі піктограми (одноразовим натисканням на них лівої кнопки миші). Зніміть "активність" із останньої піктограми, клацнувши лівою кнопкою миші на вільному місці робочого столу. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.3 Перемістіть піктограму **Мой комп'ютер** в інше місце екрана, натиснувши лівою кнопкою миші на піктограмі і, не відпускаючи її рухаючи мишу (у потрібному місці робочого столу відпустіть кнопку миші). По-

верніть переміщену піктограму на попереднє місце. Виконайте аналогічні дії з двома – трьома іншими піктограмами. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.4 Перетворіть піктограму **Мой компьютер** у вікно, двічі клацнувши на ній лівою кнопкою миші (зверніть увагу, що після цього на панелі задач поруч із кнопкою Пуск з'явиться кнопка **Мой компьютер**). Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.5 У вікні **Мой компьютер** знайдіть значок D: і перетворіть його у вікно. У двох відкритих вікнах знайдіть значки папок (□), значки програм (□), значки дисків (Ⓜ) та значки текстових документів (□). Якщо деякі із значків відрізняються зовнішнім виглядом, відмітьте це у звіті. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.6 Перетворіть одну з піктограм у вікні (D:) у нове вікно. Це вікно будемо називати як <останнє>. Використовуючи смугу прокручування, перегляньте зміст папки. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.7 В останньому вікні виберіть будь-який файл (який дозволить викладач!) і натисніть праву кнопку миші. Виберіть команду **Свойства**, у якому відображається інформація про обраний файл. Зверніть увагу, що на панелі задач розміщені кнопки з назвами **Мой компьютер**, (D:) та інші, які Ви вже відкривали. Кнопка **Свойства** відсутня, а у вікні **Свойства** немає кнопки перетворення вікна у піктограму (-). Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.8 Використайте кнопку одержання допомоги ?, яка розміщена у вікні **Свойства**, (натиснувши на неї), та отримайте підказку з будь-якого об'єкта вікна (підвівши до нього покажчик у вигляді знака ? та натиснувши на ліву кнопку миші). Зніміть з екрана підказку, натиснувши ліву кнопку миші у будь-якому місці вікна **Свойства**. Отримайте підказку з двох-трьох інших об'єктів. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.9 Почергово зробіть активними різні вікна, клацнувши мишкою у будь-якому місці цих вікон. Переконайтеся в тому, що на передньому плані завжди знаходиться активне вікно. Змініть розміри відкритих вікон таким чином, щоб усі вікна було видно на екрані (підведіть покажчик миші до рамки того чи іншого вікна, або до його кутка, і, коли покажчик набуде вигляду двоспрямованої стрілки, натисніть ліву кнопку миші і не відпускаючи переміщуйте мишу). Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.10 Повідомте викладача про виконану роботу.

1.1.4.11 Закрийте вікно **Свойства**, використовуючи кнопку закриття (✖) цього вікна.

1.1.4.12 Використовуючи кнопку перетворення вікна у піктограму (-), перетворіть відкриті вікна в піктограми в такій послідовності: <останнє>

зікно, яке Ви відкривали, (D:), **Мой компьютер** (зверніть увагу на те, що після цього на панелі задач зберігаються назви кнопок <остання>, (D:), **Мой компьютер**). Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.13 Використовуючи кнопки **Мой компьютер**, (D:) та <остання>, які розміщені на панелі задач, відкрийте відповідні вікна. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.14 Використовуючи кнопку закриття вікна, закрийте відкриті вікна в такій послідовності: <остання>, (D:), **Мой компьютер** (зверніть увагу, що на відміну від п. 12 у даному випадку на панелі задач відповідних кнопок немає). Після виконання дій головне вікно матиме такий самий вигляд, який воно мало після виконання п. 1. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.15 Перетворіть піктограму **Мой компьютер** у вікно і відкрийте контекстне меню диска **D:** (не забудьте при цьому використати праву кнопку миші). Зверніть увагу, що доступними у відкритому контекстному меню є лише ті команди, які зображені не бляклим шрифтом. Відкрийте контекстне меню для двох-трьох інших об'єктів. Зверніть також увагу на те, що відкритим може бути контекстне меню тільки для одного об'єкта. Закрийте вікно **Мой компьютер**. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.16 Повідомте викладача про завершення роботи. Після одержання дозволу на вимкнення комп'ютера у присутності викладача та з його дозволу! натисніть кнопку **Пуск**. На екрані має з'явитися меню. Виберіть команду **Завершение работы**. Після цього з'явиться вікно **Завершение работы с Windows**, яке містить три команди: **Приостановить работу**, **Выключить компьютер**, **Перезагрузить компьютер**, **Перезагрузить компьютер в режиме MS-DOS**. Якщо виставити прапореч (крапку) на **Выключить компьютер** та натиснути на кнопку **Да**, комп'ютери останніх модифікацій вимикаються. Потрібно після цього, за необхідності, вимкнути електроживлення. Для комп'ютерів більш ранніх випусків після появи на екрані повідомлення **Теперь питание компьютера можно отключить** натиснути кнопку вимикання комп'ютера на системному блоці. Опишіть у звіті Ваші дії та реакцію на них комп'ютера.

1.1.4.17 В кінці заняття пред'явіть викладачеві зроблені Вами записи та рисунки для підпису та отримання попередньої оцінки за виконану роботу.

1.1.4.18 Звіт з роботи оформити до наступного заняття.

## 1.1.5 Контрольні запитання

1.1.5.1 Що таке файл, якою повинна бути назва файла?

1.1.5.2 Що таке **Главное меню** і як його знайти? Продемонструйте.

- 1.1.5.3 Поясніть призначення спеціальних клавіш.
- 1.1.5.4 Для чого призначена клавіатура?
- 1.1.5.5 Що містить інтерфейс вікна?
- 1.1.5.6 Де розташовані кнопки управління видом вікна? Що вони здійснюють? Продемонструйте.
- 1.1.5.7 Що містить панель **Меню програми**?
- 1.1.5.8 Де знаходиться панель інструментів, що вона містить і для чого вона?
- 1.1.5.9 Що таке **Вложенное меню**? Продемонструйте.
- 1.1.5.10 Як перетворити піктограму будь-якої програми у вікно? Продемонструйте це, використовуючи піктограму **Мой компьютер**.



## 1.2 ЛАБОРАТОРНА РОБОТА №1.2

**1.2.1 Тема:** Робота зі стандартними програмами у Windows 9X

**1.2.2 Мета:** опанувати основні принципи роботи зі стандартними програмами Windows. Ознайомитися з призначенням та можливостями цих програм.

### 1.2.3 Теоретичні відомості

1.2.3.1 У групі програм "Стандартные" розташовані такі програми:

**WordPad** – текстовий редактор

**Paint** – графічний редактор

**Блокнот** – для створення заміток, записок

**Калькулятор** – для обчислень різних математичних функцій

**Таблиця символів** – служить для вставки символів в будь-які документи, які не завжди можна знайти в списку доступних шрифтів.

2.2.3.2 Для запуску потрібної вам програми треба клацнути мишкою на значку програми.

2.2.3.3 Панель інструментів графічного редактора **Paint** містить:

— **Олівець** – призначений для рисування ліній. Для рисування лінії слід встановити курсор на початок лінії і при натиснутій кнопці миші протягнути до кінця лінії.

— **Пензель** – призначений для рисування довільних фігур. Після вибору цього інструмента курсор миші вибирає форми пензля (прямокутна форма, кругла і т.і.). Для рисування слід встановити курсор в початок лінії і при натиснутій кнопці миші протягнути до кінця лінії.

— **Розпилювач** - призначений для створення тонових зображень.

— **Гумка** - призначена для стирання частини рисунка до кольору фону. Після активізації цієї кнопки, відкривається вікно (під панеллю інструментів), у якому можна вибрати розміри гумки.

**Після вибору вищеназваних інструментів, курсор набуває форму цього інструмента. Якщо, рисуючи, натиснути клавішу Shift, то курсор буде переміщуватись тільки горизонтально, вертикально або під кутом 45°.**

— Кнопка **Заливка** призначена для заповнення обмежених областей рисунка вибраним вами кольором. Для цього слід перемістити курсор у

будь-яку точку замкненої області і натиснути ліву (заповнення вибраним вами кольором) або праву (заповнення кольором фону) кнопку миші.

— **Лінія** – цей інструмент призначений для побудови прямих ліній. У вікні, що розташовано нижче можна вибрати товщину лінії. Рисувати слід при натиснутій лівій кнопці миші. Якщо утримувати при рисуванні натиснутою клавішу **Shift**, то кут лінії буде кратний 45°.

— **Крива лінія** – призначено для рисування дуг. Для побудови кривої слід нарисувати спочатку пряму лінію, вставити курсор у вершину дуги і переміщенням курсора змінити кривизну.

— **Прямокутник** – цей інструмент призначений для побудови прямокутників. Для цього треба встановити курсор на одну з його вершин і при натиснутій кнопці миші перемістити курсор в протилежну вершину. Якщо утримувати при рисуванні натиснутою клавішу **Shift**, то будуватиметься квадрат. Відпускання кнопки миші фіксує побудову.

— **Округлений прямокутник** – інструмент призначений для побудови округленого прямокутника. Побудова здійснюється аналогічно інструменту **Прямокутник**.

— **Еліпс** – призначений для побудови еліпса. Побудова здійснюється аналогічно інструменту **Прямокутник**. Якщо утримувати при рисуванні натиснутою клавішу **Shift**, то будуватиметься коло.

— **Багатокутник** – призначений для побудови багатокутників. Для побудови слід перемістити курсор миші за контуром багатокутника, натискаючи кнопку миші в кожній із вершин. В останній вершині необхідно натиснути кнопку миші двічі. Якщо утримувати при рисуванні натиснутою клавішу **Shift**, то багатокутник буде мати кути тільки 45 і 90 градусів.

— **Текст** – інструмент призначено для введення текстових надписів на рисунках.

Кнопки **Прямокутне виділення** і **Довільне виділення** призначені для вирізання (виділення, копіювання) відповідно прямокутної і довільної області рисунку.

## 1.2.4 Хід роботи

1.2.4.1. Користуючись кнопкою **Пуск**, знайдіть в Стандартних програмах **Таблицю символів**. Відкрийте її.

1.2.4.2 Змінюючи шрифт продивіться які бувають символи, значки. Якщо натиснути на символ, можна продивитися його у збільшеному вигляді.

1.2.4.3 Відкрийте вкладку **Специальные символы**, дослідіть її. Також вивчіть всі кнопки цього вікна, дослідіть як вони працюють. Законспектуйте це у звіті. Після цього закрийте Таблицю символів.

1.2.4.4 В Стандартних програмах знайдіть і відкрийте програму **Калькулятор**, дослідіть які бувають види Калькулятора, підрахуйте такі приклади:

$$380*5-(35+(12-9))=;$$

$$\log: 67, 20, 5, 15;$$

$$x^y: 12^2; 32^5; 50^{13}; 4^{60};$$

$$1/x: 20; 5; 0.8; 87;$$

$$\text{tg}: 45; 60; 89; 300;$$

та запишіть відповіді у звіті.

1.2.4.5 Відкрийте програму **Блокнот**. Вивчіть Меню програми. Законспектуйте це у звіті. Надрукуйте текст у Блокноті:

На котельні **Вінницького ПТУ – 44** знаходиться два парових котли Е 1/9, які переконструйовані на водогрійні. На даний час здійснюється реконструкція котельні. З цією метою необхідно виконати такі роботи:

- гідравлічні випробування вузла хімводоочистки;
- *придбати тепловий лічильник;*
- *виконати ізоляцію теплових мереж та газоходів.*

*На реалізацію проекту необхідно 25 тис. грн.*

**Завершення реконструкції котельні дозволить:**

зеконотити природний газ (на 1,5 тис. м<sup>3</sup>/рік), електроенергію (на 4,5 тис.кВт год/рік); збільшити ККД котла до 91%; підвищити надійність

1.2.4.6 Збережіть файл в своїй папці під назвою ЛР2 (лабораторна робота 2). Після цього закрийте **Блокнот**.

1.2.4.7 У папці вашої групи **ТЕ-021** знайдіть файл **1.bmp**, відкрийте його. У цьому файлі міститься рисунок, що демонструє всі можливості графічного редактора **Paint**. Проаналізуйте, за допомогою яких інструментів виконані ці рисунки.

1.2.4.8 Створіть новий документ і, використовуючи інструкції що подані в теоретичній частині, створіть рисунок – вітання. зробіть надписи.

Спробуйте копіювати частини рисунку, видалити, перемістити. Встановіть

**Макет сторінки** користуючись кнопкою меню **Файл**.

1.2.4.9 Збережіть цей файл у своїй папці з будь-якою назвою.

2.2.4.10 Повідомте викладача про виконану вами роботу.

### **1.2.5 Контрольні запитання**

1.2.5.1 Які програми входять до Стандартних?

1.2.5.2 Для чого призначена програма Таблиця символів?

1.2.5.3 Для чого призначена програма Блокнот?

1.2.5.4 Як працювати з графічним редактором? Продемонструйте.

1.2.5.5 Як працювати з калькулятором?

1.2.5.6 Як ввести текст у рисунок в графічному редакторі?

1.2.5.7 Як можна скопіювати, перемістити, видалити рисунки в графічному редакторі?

## 1.3 ЛАБОРАТОРНА РОБОТА №1.3

### 1.3.1 Тема: текстовий редактор WordPad

**1.3.2 Мета:** опанувати основні принципи роботи з стандартним текстовим редактором

### 1.3.3 Теоретичні відомості

**Буфер обміну** – комірка пам'яті, куди можна тимчасово записати якусь інформацію. Якщо виділити, наприклад, файл, ділянку тексту чи картинку і вибрати з меню "Правка" команду "Копировать" або "Вырезать", то цей текст чи картинка буде знаходитись в **буфері обміну**, звідки можна командою "Вставка" вставити виділений об'єкт в інше місце тексту або зовсім окремий файл. При наступному копіюванні, попередня інформація в **буфері обміну** стирається.

### 1.3.4 Хід роботи

1.3.4.1. Через кнопку "Пуск" відкрити в **Стандартних** програмах текстовий редактор **WordPad**.

1.3.4.2. Підведіть покажчик миші до кожної з кнопок панелі інструментів і, затримуючи його дочекайтесь віконця повідомлення про призначення кнопки так, як показано на рисунку.

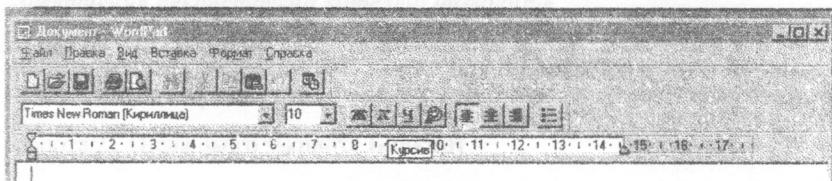


Рисунок 1.2 — Панель інструментів текстового редактора **WordPad**

1.3.4.3. Зарисуйте ескіз панелей інструментів, позначте кожну з кнопок і запишіть призначення кожної з них у свій звіт.

1.3.4.4. Зайдіть в меню **Справка**, прочитайте і законспекуйте у звіті такі розділи:

- **робота с документами, создание нового документа**
- **робота с текстом**
- **форматирование текста**

1.3.4.5. Відкрийте новий документ. Виберіть параметри сторінки, користуючись кнопкою меню **Файл – Макет страницы**. Встановіть шрифт **Times New Roman (Кириллица)** і, за допомогою клавіатури наберіть текст:

Серед термічних процесів швейного виробництва найбільшу питому вагу займає волого-теплова обробка (ВТО), яка застосовується в підготовчих, всередині процесних етапах обробки, при кінцевій обробці виробів. Трудомісткість процесів ВТО достатньо велика і складає до 25 – 30 % від загальної трудомісткості виготовлення одєжі.

1.3.4.6. Зробіть копію частини **цього** тексту, виставте орієнтацію - **по левому краю**.

1.3.4.7. Змінійте колір тексту (червоний, синій, зелений тощо).

1.3.4.8. Виділіть частину тексту та змініть характер шрифту: курсив, підкреслений.

1.3.4.9. Спробуйте встановити інший тип шрифту та змінити його розмір (20, 16, 26).

1.3.4.10. Виберіть іншу мову, наприклад англійську. Надрукуйте речення англійською.

1.3.4.11. В меню **Правка** виберіть команду **Найти**. У віконце напроти слова **Образец** введіть літеру а. Користуючись кнопкою **Найти далее**, знайдіть всі букви а. Після цього закрийте це вікно.

1.3.4.12. В меню **Правка** виберіть команду **Заменить**. Поміняйте літери а на в, користуючись кнопками **Найти далее** та **Заменить** або **Заменить все**.

1.3.4.13. Повторіть цю операцію замінюючи, наприклад літеру е на знак !



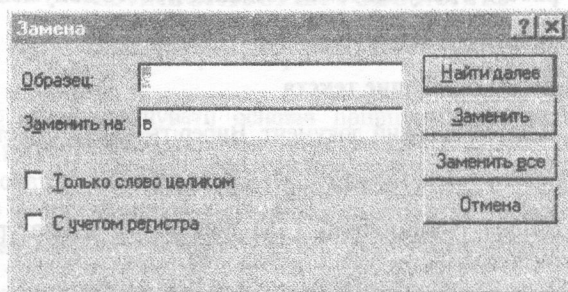


Рисунок 1.3 — Панель "Замена" команды "Правка"

1.3.4.14. Не закриваючи вікно редактора WordPad через кнопку **Пуск** відкрийте стандартну **Таблицю символів**. Вставте в свій документ 2-4 символи, користуючись буфером обміну.

1.3.4.15. Збережіть документ в свою папку, користуючись кнопкою меню **Файл**, командою **Сохранить как**.

1.3.4.16. Повідомте викладача про виконану вами роботу.

### 1.3.5 Контрольні запитання

1.3.5.1 Поясніть призначення кожної кнопки панелі інструментів.

1.3.5.2 Що таке буфер обміну? Як ним користуватись? Що означає команда **Вирізати**? Продемонструйте.

1.3.5.3 Як здійснити копіювання частини тексту?

1.3.5.4 Як змінити колір тексту?

1.3.5.5 Як вибрати параметри сторінки?

1.3.5.6 Поясніть дію кожної команди меню **Правка**?

1.3.5.7 Поясніть дію кожної команди меню **Файл**?

1.3.5.8 . Поясніть дію кожної команди меню **Вид**?

1.3.5.9 Що значить змінити орієнтацію тексту? Продемонструйте.

## 1.4 ЛАБОРАТОРНА РОБОТА №1.4

### 1.4.1 Тема: Робота з програмою-архіватором

**1.4.2 Мета:** опанувати основні принципи роботи з стандартним архіватором.

### 1.4.3 Теоретичні відомості

Архіватори – це програми, що дозволяють зменшити розмір файла для економії місця на дискові. У файлах зустрічаються фрагменти що часто повторюються і немає сенсу тримати їх на диску. Задача архіватора полягає у знаходженні цих фрагментів і записуванні замість них іншої інформації, за допомогою якої можна потім відновити файл. Наприклад, текстові файли стискаються архіватором приблизно в два рази, файли для чорно-білих картинок – в три - чотири рази і т.ін.

Програм-архіваторів існує багато. Розрізняють їх в основному за швидкістю архівування і розархівування, а також і ступенем стиснення. Найвідоміші архіватори – WinZip, WinRar.

Один із недоліків магнітних носіїв полягає у тому, що вони не забезпечують абсолютну збереженість інформації. Ця інформація може бути зруйнована частково або повністю внаслідок фізичного псування носія, внаслідок дії зовнішніх магнітних полів, старіння магнітного покриття та ін. Ці обставини вимагають користувача мати на дискетах архівні копії документів.

Загальні функції архіваторів:

- можливість додавання файлів в архів
- поновлення архіву
- перегляд файлів в архіві
- знищення файлів в архіві
- вилучення файлів з архіву

#### 1.4.4 Хід роботи

1.4.4.1 Знайдіть свою папку. Виділіть її, клацнувши на значку.

1.4.4.2 Наведіть курсор миші на виділену папку, клацніть правою кнопкою.

1.4.4.3 Ви визвали контекстне меню. Законспекуйте, які команди в даному випадку містить контекстне меню.

1.4.4.4 Із переліку команд виберіть та запустіть команду **Добавить в архив**.

1.4.4.5 Перед вами з'явиться робоче вікно параметрів архіватора

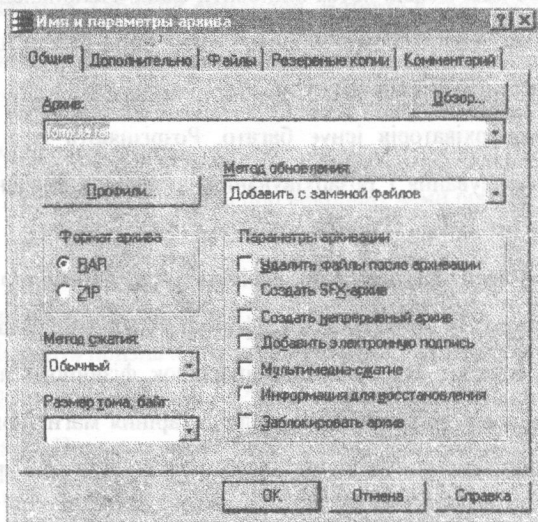


Рисунок 1.4 — Робоче вікно параметрів архіватора WinRAR.

WinRAR.

1.4.4.6 Відкриваючи вкладені меню продивіться і законспекуйте всі методи стиснення файла, можливі розміри тому архіву. На дискету вміщується приблизно 1457 байт. Якщо вибрати розмір тому 1457, то програма розіб'є архів на частини, кожна з яких вміщується повністю на дискету.

1.4.4.7 Продивіться призначення кнопки **Обзор**. Чи можна за допомогою цієї кнопки відшукати файли, що потрібно архівувати?

1.4.4.8 Виберіть параметри архівації – **Создать SFX – архив**. Призначте формат архіву **Rar**. Натисніть кнопку **ОК**. Програма створила в папці вашої групи **архів, що саморозпаковується**, під назвою вашої папки.

1.4.4.9 Знайдіть цей архів. Щоб розпакувати **SFX**-архіви треба двічі натиснути на піктограмі цього архіву.

1.4.4.10 Видаліть шойно створений архів в корзину. Закрийте всі вікна.

1.4.4.11 Користуючись кнопкою **Пуск, Программы**, знайдіть в переліку програм **WinRar**. Запустіть програму.

1.4.4.12 Коротко законспекуйте зміст меню цієї програми.

1.4.4.13 У адресному рядку виберіть **диск D**, папку **Мои документы**, папку вашої групи, потім вашу папку.

1.4.4.14 Заархівуйте її, довільно вибравши параметри і формат архіву.

1.4.4.15 Знайдіть новий архів у робочому вікні програми, продивіться склад архіву, не розпаковуючи його. Для цього достатньо двічі клацнути на значку архіву.

1.4.4.16 Коротко законспекуйте особливості роботи з архіватором.

## **1.4.5 Контрольні запитання**

1.4.5.1 Навіщо архівувати файли?

1.4.5.2 Які загальні функції архіваторів?

1.4.5.3 Які з архіваторів вам відомі?

1.4.5.4 Що таке архіватор?

1.4.5.5 Поясніть принцип роботи програми-архіватора.

1.4.5.6 Поясніть порядок архівування файла. Продемонструйте.

1.4.5.7 Якою програмою-архіватором ви користувались, виконуючи лабораторну роботу? Архіви яких форматів можна створювати за допомогою цієї програми?

1.4.5.8 Заархівуйте файл користуючись контекстним меню.

1.4.5.9 Як визвати контекстне меню?

1.4.5.10 Що таке том архіву? Який він має розмір?

1.4.5.11 Як переглянути заархівовані файли?

1.4.5.12 Чи є різні способи виклику програми-архіватора? Які?

## 1.5 ЛАБОРАТОРНА РОБОТА №1.5

### 1.5.1 Тема: OpenOffice Word, графічні можливості

1.5.2 Мета: навчитися працювати в графічному редакторі OpenOffice Word

### 1.5.3 Теоретичні відомості

Правила рисування в графічному редакторі OpenOffice Word такі ж самі як і в стандартному Paint.

Команди рисування:

- виділення
- лінія
- еліпс
- прямокутник
- крива
- "Рисованная кривая"
- дуга
- сектор еліпса
- сегмент
- текст
- біжучий рядок
- легенда

Кожний об'єкт в цьому редакторі вбудований в умовний прямокутник, який відображається зеленими крапочками при виділенні цього об'єкта.

Графічний об'єкт вважається виділеним тоді, коли на ньому з'являються зелені мітки. Для виділення об'єкта достатньо клацнути на ньому мишкою.

Щоб видалити графічний об'єкт треба виділити його і натиснути кнопку **Delete**.

Зміст контекстного меню графічного редактора:



— **розташування** – можливість переміщення об'єктів на передній план, на задній план; на один об'єкт вперед, назад;

— **вирівнювання** – дозволяє вирівняти один об'єкт відносно іншого. Команда активна, коли є декілька графічних об'єктів;

— **обтікання** – дозволяє визначити положення згрупованого об'єкта в тексті. Команда активна, коли рисунок весь згрупований;

— **прив'язка** – закріплення рисунку за абзацом, сторінкою, можливість перетворення його у знак;

— **групування** – команда активна, коли є декілька графічних об'єктів. Для здійснення групування необхідно виділити всі нарисовані об'єкти. Для цього тримаючи нагиснутою клавішу **Shift**, вибираємо(виділяємо) мишкою всі частини рисунку так, щоб кожна фігура мала зелені мітки. Після цього визиваємо контекстне меню і вибираємо команду **Групувати**;

— **область** – команда встановлення кольорів, їх відтінків, штриховки, типів заливки замкнутих контурів;

— **лінія** – вибір кольору лінії, типу лінії (штрихована, штрих пунктирна і т.д.), характер кінців лінії.

### **Кнопка Положення і розмір:**

**позиція** — дозволяє розташувати базисну точку (початок координат) або в центрі екрана, або в різних кутках екрана;

**розмір** — встановлювання розмірів об'єкта, що виділений. При виборі цієї команди встановлюється ширина і довжина того умовного квадрата, в який вбудований будь-який графічний об'єкт (квадратик відображається при виділенні зеленими крапочками);

**поворот** — повертання виділеного об'єкта на заданий кут;

**наклон/радіус** заокруглення кута — встановлюється значення кута нахилу виділеного об'єкта, радіус заокруглення кута.

## 1.5.4 Хід роботи

1.5.4.1 Завантажити систему OpenOffice Word, дочекатися повного її відкриття.

1.5.4.2 Перевірити чи встановлені потрібні панелі інструментів. Для цього відкриваємо меню Вид – Панелі символів. На вкладеному меню повинні бути відмічені галочками всі панелі: **панель функцій**  
**панель об'єктів**  
**панель інструментів.**

Якщо якась панель не відмічена, то встановити її (клацнути на її назві мишкою).

1.5.4.3 Для ознайомлення з панеллю, встановіть курсор – стрілочку на панель, клацніть правою клавішею миші і виберіть команду – **показати кнопки**. З'являється вкладене меню, де галочками відмічені кнопки, що встановлені на панель.

**Завдання** – встановити всі кнопки, що не встановлені. Для цього на вкладеному меню вибрати піктограму невідміченої кнопки і клацнути на неї.

1.5.4.4 Ознайомитися зі всіма можливостями і функціями графічного редактора, які описані в теоретичних відомостях. Навчитися групувати, виділяти, вибирати колір і тип заливки об'єктів.

1.5.4.5 Виконати конкурсне завдання (на вибір):

створити емблему кафедри теплоенергетики.

створити рекламу або емблему факультету, де ви навчаєтесь.

створити макет Web-сторінки кафедри ТЕ (емблема+реклама).

нарисувати Помічника для цієї програми.

## **1.5.5 Контрольні запитання**

1.5.5.1 Порівняти можливості графічних редакторів Paint і OpenOffice Word.

1.5.5.2 Пояснити кожну команду контекстного меню.

1.5.5.3 Пояснити правила користування кожної команди рисунка.

1.5.5.4 Як вибрати заливку фігури?

1.5.5.5 Як групувати об'єкт?

1.5.5.6 Як видалити об'єкт?

## 1.6 ЛАБОРАТОРНА РОБОТА №1.6

### 1.6.1 Тема: OpenOffice. Електронна таблиця Excel

**1.6.2 Мета роботи:** навчитись працювати в електронній таблиці, проводити типові математичні обчислення, будувати графіки та діаграми.

### 1.6.3 Теоретичні відомості

Робоче поле **Excel** – не чиста сторінка, а табличка, яку треба заповнювати.

**Excel** – дозволяє обчислити суму стовпців та рядків, проценти, вирахувати середнє арифметичне значення, — в ньому можна виконати безліч стандартних функцій – математичних, фінансових, статистичних, логічних.

Кожний файл **Excel** називається Книга. Файл має 16 сторінок – таблиць (листів). Рядки кожної таблиці пронумеровані цифрами, стовпці – літерами. Таким чином у кожній клітинці на робочому листі є своя адреса. Наприклад, перша клітинка зліва має адресу A1.

Якщо клацнути на клітинці мишкою, то вона виділяється рамочкою (стає активною) і можна в цю клітинку ввести будь-що (набрати з клавіатури і натиснути клавішу Enter).

В кожній клітинці таблички може знаходитись число, текст або формула.

#### **Панелі Excel:**

- панель функцій – перша горизонтальна панель;
- панель об'єктів – друга горизонтальна панель;
- панель інструментів – розташована зліва вертикально;
- панель формули – третя горизонтальна панель.

**Панель функцій** містить такі основні кнопки:

"Создать", "Открыть файл", "Сохранить документ", "Сохранить как", "Печать", "Вырезать", "Копировать", "Вставить", "Отменить", "Восстановить" і т.і.

**Панель об'єктів** має такі кнопки:

"Тип шрифта", "Размер шрифта", "Жирный", "Курсив", "Подчеркну-

тый", "По левому краю", "По правому краю", "По центру", "По ширине", "Числовой формат", "Процент", "Обрамление", "Вставить строки", "Вставить колонки", "Удалить строки", "Удалить колонки" і так далі.

**Панель формули** складається з рядка введення функції, "Автосумма" ( $\Sigma$ ) і т.п.

**Панель інструментів** містить такі основні кнопки:

"Вставить объект", "Вставить ячейки", "Функция" "Рисование", "Формуляр", "Правописание", "Исходные данные", "Автопроверка орфография".

### Як швидко скопіювати символ, формулу в Excel?

Введіть в будь-яку клітинку цифру.

Клацніть на клітинці з цифрою. Ви бачите, що вона виділяється рамочкою, в нижньому правому кутку якої є маленький квадратик. Достатньо потягнути за цей квадратик вниз або вправо (виділити стовпець чи рядок) і одразу ж набрана цифра скопіюється в усі виділені клітинки. Таким чином можна копіювати будь-що: формули, літери, числові значення.

## 1.6.4 Хід роботи

1.6.4.1. Запустіть програму OpenOffice Excel.

1.6.4.2. Дослідіть меню програми. Законспектуйте основні команди меню.

1.6.4.3. Наводячи мишкою на кнопки панелі продивіться її назву і призначення. Клацнувши правою кнопкою миші на панелі виберіть команду **Показать кнопки**, встановіть необхідні, на ваш погляд, кнопки.

1.6.4.4. Введіть в кожну клітинку першого рядочка будь-що.

1.6.4.5. Вставте рядок в таблицю. Для цього треба:

— виділити клітинку перед якою треба вставити рядок.

— Зайдіть в меню **Вставка**, виберіть команду **Строка**.

1.6.4.6. Вставте стовпець (аналогічно).

1.6.4.7. Клацніть на кнопочку назви рядочка (1,2,3,4,5,...) або стовпця (A,B,C...) – виділяється весь рядок або стовпець.

1.6.4.8. Ведіть стовпець цифрових значень.

1.6.4.9. Вставте такі функції в клітинки:

косинус, натуральний логарифм, синус, тангенс, квадратний корінь, множення чисел, функція  $X^y$ .

Для цього треба виділити клітинку, зайти в меню **Вставка – Функції**, або вибрати відповідну кнопку на панелі. На екрані з'явиться діалогове вікно, куди треба ввести аргумент (число) для обчислення даної функції.

1.6.4.10. Ввести стовпці чисел  $A_i$  і  $B_i$ .

1.6.4.11. В стовпець C, вставити суму  $A_i+B_i$ . Для цього потрібно.

— Виділити клітинку, куди треба вставити суму (C1)

— Встановити курсор в рядок введення функції

— набрати з клавіатури знак "="

— покажчиком миші клацнути на комірці першого доданку (A1). При цьому в рядочку функції відобразиться "=A1". Потім набрати з клавіатури знак додавання "+" і клацнути мишкою на клітинці з другим доданком (B1). В рядку формул відобразиться "=A1+B1".

— натиснути Enter – для введення функції в клітинку C1.

— В C1 повинна відобразитись сума комірок A і B.

— скопіювати цю формулу в кожен наступну клітинку стовпчика C, користуючись методом швидкого копіювання в **Excel**.

1.6.4.12. Зайдіть в контекстне меню, попередньо виділивши клітинку. Дослідіть його. Законспекуйте.

1.6.4.13. Вставте діаграму.

## 1.6.5 Контрольні запитання

- 1.6.5.1. Для чого призначений **Excel**?
- 1.6.5.2. Які можливості табличного редактора?
- 1.6.5.3. Зміст контекстного меню.
- 1.6.5.4. Як вставити формулу в клітинку? Продемонструйте.
- 1.6.5.5. Поясніть всі команди меню **Вставка**.
- 1.6.5.6. Поясніть всі команди меню **Формат**.
- 1.6.5.7. Які види діаграм бувають в **Excel**?
- 1.6.5.8. Як вставити діаграму?
- 1.6.5.9. Поясніть роботу з рядочком функцій.
- 1.6.5.10 Як швидко скопіювати в **Excel**?

## 2 ОСНОВИ ПРОГРАМУВАННЯ

### 2.1 ЛАБОРАТОРНА РОБОТА №2.1

**2.1.1 Тема:** програмування та розв'язання задач з лінійним обчислювальним процесом

**2.1.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic та розв'язання лінійних задач з використанням ЕОМ

#### 2.1.3 Теоретичні відомості

При програмуванні мовою Visual Basic використовуються:

символи латинського алфавіту (A,...Z, a,...z);	} 3. цих символів складаються назви змінних, функцій, констант і т.д.
символи українського алфавіту (A,...Я, а,...я);	
цифри від 0 до 9	
символ підкреслення_	

Крім цих символів до алфавіту мови Visual Basic входять:

символи, що беруть участь в побудові конструкцій (програмуванні), наприклад +, —, \*, \, /, =, <, >, [, ], (, ), ', &, @ та ін комбіновані символи =>, <=, <>.

**Програма** у Visual Basic являє собою послідовність лексичних одиниць – **лексем** (знаків, літер) які записані згідно із загальноприйнятими синтаксичними правилами програмування і реалізують будь-яку логічну конструкцію.

Для забезпечення зрозумілості програми в тексті програми вставляють коментарі. **Коментар** – це послідовність будь-яких символів (слів), що розміщені в одному рядочку і відокремлені знаком ' або позначені словом **Rem**, наприклад

```
'Коментар  
Rem Коментар
```

**Лексема** – одиниця тексту програми, яка має свій зміст і не може бути поділена в подальшому. У Visual Basic існує шість класів лексем:

- ідентифікатори (назви змінних, програм, операцій, функцій ...);
- службові слова – оператори (спеціальні слова мови програмування);
- константи;
- ряди;
- математичні операції (+ — = / і т.д.);
- розділові знаки (кома, крапка, інтервал і т.д.).



Назви (ідентифікатори) у Visual Basic повинні починатися з літери, не повинні містити розділових знаків (коми, крапки, інтервалу і т.д.), спеціальних знаків. Приклади ідентифікаторів:

**N, I, Number, Номер** — проста назва змінних;  
**Прізвище\_студента** — складна назва.

### Типи змінних у Visual Basic

#### Тип змінної

цілі числа  
логічні значення  
довгі цілі числа  
дійсні числа  
текстовий рядок  
рисунок, будь-який об'єкт

#### Позначення

Integer  
Boolean  
Long integer  
Single  
String  
Object

Кожна програма складається з блоків:

- початку програми (назва);
- оголошення та введення змінних;
- введення початкових даних;
- тіла програми (будь-яких математичних, логічних операцій);
- виведення результату (наприклад на екран);
- кінця програми.

Кожний блок програмується за допомогою спеціальних службових слів – операторів:

початок програми – **Sub ..назва програми..**

кінець програми – **End Sub**

оголошення змінної – **Dim ..назва змінної.. As ..тип змінної..**

де **Dim** (розмір) – службове слово, означає, що резервується область пам'яті для названої змінної; **As** (як) – службове слово, визначає тип змінної наприклад

**Dim X As Single** — змінна типу Single – дійсне число.

#### Інший спосіб оголошення змінної:

**Dim X!** — тут суфіксом ! позначається тип змінної, в даному випадку – Single.

Таким чином, тип змінної можна визначати за допомогою суфіксів (знаків), що добавляються вкінці назви змінної, годі застосовувати слово **As** не треба.

Тип змінної

суфікс

Integer	%
Long integer	&
Single	!
String	\$

### Математичні, логічні операції та вбудовані функції у Visual Basic

A+B додавання; менше;	A^B степінь;	A<B
A—B віднімання;	A=B дорівнює;	A=>B більше або дорівнює;
-A знак числа;	A<>B не дорівнює;	A<=B менше або дорівнює;
A*B множення;	A>B більше;	ABS(X) мо- дуль X;
SIN(X) синус;	SQR корінь квадратний;	LOG(X) ln x;
EXP(X) e <sup>x</sup> ;	COS(X) косинус X;	
ATN(X) арктангенс X;	LOG10(X) lg x;	
TAN(X) тангенс X.		

Крім того, через арктангенс можна визначити такі тригонометричні функції як:

$$\arcsin x = \arctg\left(\frac{x}{\sqrt{1-x^2}}\right) = \text{ATN}(x/\text{SQR}(1-x^2));$$

$$\arccos x = \left| \arctg\left(\frac{\sqrt{1-x^2}}{x}\right) \right| = \text{ABS}(\text{ATN}(\text{SQR}(1-x^2)/x))$$

$$\text{arccctg } x = \left| \arctg\left(\frac{1}{x}\right) \right| = \text{ABS}(\text{ATN}(1/x)).$$

Існує два типи введення початкових даних:

- з використанням оператора присвоєння **Let** наприклад  
**A=10**
- з використанням оператора **Input**, наприклад,  
**A= InputBox("введіть значення для A", "введення початкових даних")**

При цьому для перевірки введених даних рекомендується виводити їх значення на екран, наприклад, таким оператором:

**MsgBox " A= " & A & "; B = " & B & "; X = " & X, vbOKOnly, "Перевірка даних"**

Виведення результату на екран здійснюється за допомогою оператора **Print**, наприклад

Print "Результат обчислень y = "; y

або

**MsgBox "Результат обчислень y = "& y, vBokCancel, "Виведення результатів"**

Приклад програми для наведеної на рисунку 2.1 блок-схеми з вико-

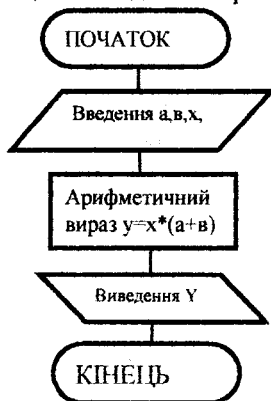


Рисунок 2.1 — структура програми лінійних обчислень (схема алгоритму)

ристання оператора присвоювання даних LET:

```
Rem Програма
Sub LP1_1
    Dim A As single
    Dim B As single
    Dim X As single
    Dim Y As single
    A=1
    B=2
    X=3
    Y=X*(A+B)
    Print "Результати розрахунку Y=", Y
End Sub
```

Приклади програм для виконання у середовищі Microsoft Office наведено у додатку А.

## 2.1.4 Хід роботи

2.1.4.1. Отримавши свій варіант завдання уважно записати свій арифметичний вираз за правилами мови програмування.

2.1.4.2. Скласти блок-схему алгоритму програми

2.1.4.3. Скласти два варіанти тексту програми 1-й із введенням вхідних даних оператором присвоєння Let, 2-й із введенням даних оператором Input.

2.1.4.4. Показати тексти програм викладачеві.

2.1.4.5. Запустити програму OpenOffice Word.

2.1.4.6. Зайти в меню Сервіс—Макрос—Організувати—Новий модуль.

2.1.4.7. Після виконання цих команд з'явиться на екрані вікно, куди треба записати таку назву програми LR2\_1\_1, після цього клацнути мишкою на команді Правка.

2.1.4.8. Перед вами відкрилося вікно Visual Basic в якому ви бачите дві панелі інструментів. В процесі програмування ми працюємо з нижньою панеллю інструментів.

2.1.4.9. Роздивіться уважно призначення кнопок панелі інструментів.

2.1.4.10. На робочому полі редактора Visual Basic ви бачите макет майбутньої програми. Він має вигляд:

```
Rem ***** Program *****
```

```
Sub
```

```
End Sub
```

2.1.4.11. Після введення коментарів (після Rem), введення назви програми (після Sub) набираємо текст 1-го варіанта програми.

2.1.4.12. Після набору програми збережемо створений файл. Для цього на нижній панелі інструментів клацнемо на піктограмі "Сохранить". У вікні, що з'явилося, в адресному рядочку виберемо Диск D—(далі за вказівкою викладача). В рядочку назви напишемо таку назву LP2\_1\_1 — лабораторна робота номер 1, 1-й варіант і натиснемо кнопку "Сохранить".

2.1.4.13. Після цього на нижній панелі інструментів клацнемо мишкою на кнопці запуску програми на виконання.

2.1.4.14. Ви повинні отримати результат.

2.1.4.15. Після цього відкрити нове вікно редактора Visual Basic і ввести другий варіант тексту програми.

2.1.4.16. Результати виконання 1-го і 2-го варіантів програм повинні збігатися.

2.1.4.17. Відкрити програму OpenOffice Excel і зробити контрольний варіант розрахунку вашого рівняння.

## 2.1.5 Контрольні запитання

2.1.5.1 Що таке програма?

2.1.5.2 Для чого вставляють коментар в текст програми? Приклад.

2.1.5.3 За допомогою якого оператора здійснюється виведення результату на екран. Наведіть приклад

2.1.5.4 Наведіть приклад програми з використанням оператора присвоєння LET.

- 2.1.5.5 Що таке коментар?
- 2.1.5.6 Скільки існує видів лексем?
- 2.1.5.7 Які символи входять до алфавіту мови Visual Basic?
- 2.1.5.8 З яких блоків складається програма?
- 2.1.5.9 Наведіть приклад програми з використанням оператора введення INPUT.
- 2.1.5.10 Що таке лексема?
- 2.1.5.11 Скільки типів змінних ви знаєте?
- 2.1.5.12 Як запрограмувати резервування оперативної пам'яті для змінної?
- 2.1.5.13 Правило складання назв ідентифікаторів у Visual Basic.
- 2.1.5.14 Як оформити коментар мовою Visual Basic (два варіанти)?
- 2.1.5.15 Назвіть два способи резервування (оголошення) змінної у Visual Basic-програмі .
- 2.1.5.16 Які існують два типи введення початкових даних?
- 2.1.5.17 Наведіть приклад програми з використанням оператора введення INPUT.

## 2.2 ЛАБОРАТОРНА РОБОТА №2.2

**2.2.1 Тема:** Програмування та розв'язання задач з розгалуженим обчислювальним процесом

**2.2.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic та розв'язання на EOM задач з розгалуженим обчислювальним процесом за допомогою оператора умовного переходу.

### 2.2.3 Теоретичні відомості

Оператор умовного переходу або оператор вибору застосовується тоді, коли потрібно виконати розрахунок, який залежить від будь-якої умови, наприклад: якщо  $x=2$ , то  $y=3+x$ .

Існує два типи операторів умовного переходу — **безальтернативний** і **альтернативний**

Безальтернативний оператор умовного переходу можна записати:

**IF (умова) THEN (вираз)**

В перекладі на українську мову оператор виглядає так:

**ЯКЩО (умова) ТО (вираз)**

**ЯКЩО (умова виконується - дійсна) ТО (виконується вираз)**

Альтернативний оператор умовного переходу:

**IF (умова) THEN (вираз) ELSE (вираз)**

Якщо перекласти українською оператор має вигляд:

**ЯКЩО (умова) ТО (вираз) ІНАКШЕ (вираз)**

**ЯКЩО (умова дійсна) ТО (виконується вираз1) ІНАКШЕ (якщо умова не дійсна) (виконується вираз2)**

Умовою можуть бути:

Співвідношення змінних, математичних виразів тощо,  
наприклад: **IF a=5**

**IF a<=X+B**

**IF a<5**

Знаки співвідношення: =, <, >, =>, <=, <>.

Виразом умовно називається будь-який арифметичний, логічний вираз, що надає програмі напрямок подальшого розрахунку в залежності від поставленої умови. Якщо умова виконується, то далі виконується і вираз, якщо умова не виконується, то програма переходить до виконання другого виразу або наступного рядка програми.

Бувають ситуації, коли у випадку дійсності **умови** необхідно виконати декілька **виразів** (операторів). В цьому випадку застосовується **блоку-**

**вий оператор умовного переходу.** Блоковий оператор також може бути безальтернативний і альтернативний.

Блоковий безальтернативний оператор умовного переходу можна записати у вигляді:

```
IF (умова) THEN
    вираз 1
    вираз 2
    вираз 3
    вираз N
END IF
```

Тут END IF вказує на кінець блоку оператора IF.

Блоковий альтернативний оператор умовного переходу застосовується тоді, коли при виконанні умови необхідно здійснити один набір виразів (операторів), а при невиконанні умови — інший набір операторів. Блоковий альтернативний оператор має вигляд:

```
IF (умова) THEN
    вираз 1
    вираз 2
    вираз 3
    вираз N
ELSE
    вираз 1
    вираз 2
    вираз 3
    вираз N
END IF
```

Оператори умовного переходу можуть бути вкладені один в другий. Наприклад, якщо  $x > n$  і  $n > 0$ :

```
IF (умова1) THEN
    IF (умова2) THEN
        вираз 1
        вираз 2
        вираз 3
        вираз N
    ELSE
        вираз 1
        вираз 2
        вираз 3
        вираз N
    END IF
END IF
```

Приклад програми для наведеної на рисунку 2.2 блок-схеми з використанням оператора присвоювання даних LET:

```

Rem Програма
Sub LR2_1
    Dim A As single
    Dim B As single
    Dim X As single
    Dim Y As single

    A=1
    B=2
    X=3
    If x>0 Then
        Y=X*(A+B)
        Print "Результати розрахунку Y=", Y
    Else
        Y=X+A*B
        Print "Результати розрахунку Y=", Y
    End If
End Sub

```

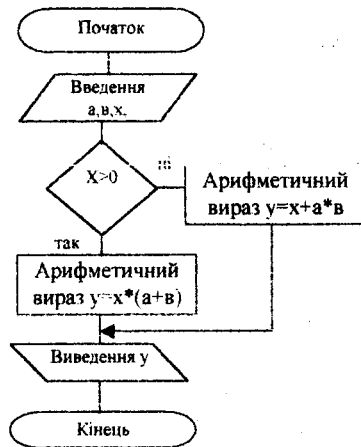


Рисунок 2.2 — Структура програми розгалужених обчислень

Приклади програм для виконання у середовищі Microsoft Office наведено у додатку Б.

### 2.2.4 Хід роботи

2.2.4.1. Отримавши свій варіант завдання уважно записати свій арифметичний вираз за правилами мови програмування.

2.2.4.2. Скласти блок-схему алгоритму програми



2.2.4.3. Скласти два варіанти тексту програми 1-й із введенням вхідних даних оператором присвоєння Let, 2-й із введенням даних оператором Input.

2.2.4.4. Показати тексти програм викладачеві.

2.2.4.5. Запустити програму OpenOffice Word.

2.2.4.6. Зайти в меню Сервіс—Макрос—Організувати—Новий модуль.

2.2.4.7. Після виконання цих команд з'явиться на екрані вікно, куди треба записати таку назву програми LR6\_1, після цього клацнути мишкою на команді Правка.

2.2.4.8 Після введення коментарів (після Rem), введення назви програми (після Sub) набираємо текст 1-го варіанта програми.

2.2.4.9 Після набору програми збережемо створений файл. Для цього на нижній панелі інструментів клацнемо на піктограмі "Сохранить". У вікні, що з'явилося, в адресному рядочку виберемо Диск D—(далі за вказівкою викладача). В рядочку назви напишемо таку назву LP2\_2\_1 — лабораторна робота номер 2\_1, 1-й варіант і натиснемо кнопку "Сохранить".

2.2.4.10 Після цього на нижній панелі інструментів клацнемо мишкою на кнопці запуску програми на виконання.

2.2.4.11 Ви повинні отримати результат.

2.2.4.12 Після цього відкрити нове вікно редактора Visual Basic і ввести другий варіант тексту програми.

2.2.4.13 Результати виконання 1-го і 2-го варіантів програм повинні збігатися.

2.2.4.14 Відкрити програму OpenOffice Excel і зробити контрольний варіант розрахунку вашого завдання.

## 2.2.5 Контрольні запитання

2.2.5.1 Коли застосовують оператор умовного переходу?

2.2.5.2 Два типи операторів умовного переходу.

2.2.5.3 Запишіть безальтернативний оператор умовного переходу. Як він перекладається?

2.2.5.4 Що може бути умовою?

2.2.5.5 Запишіть блоковий альтернативний оператор умовного переходу.

2.2.5.6 Що таке вираз?

2.2.5.7 Коли застосовують блоковий оператор умовного переходу?

2.2.5.8 Запишіть альтернативний оператор умовного переходу. Як він перекладається?

2.2.5.9 Як виглядає структура програми розгалужених обчислень?

2.2.5.10 Запишіть блоковий безальтернативний оператор умовного переходу.

## 2.3 ЛАБОРАТОРНА РОБОТА №2.3

**2.3.1 Тема:** розв'язання задач з розгалуженим обчислювальним процесом з використанням операторів умовного та безумовного переходу

**2.3.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic та розв'язанням на ЕОМ задач з використанням операторів умовного та безумовного переходу.

### 2.3.3 Теоретичні відомості

Оператор безумовного переходу застосовується тоді, коли потрібно виконати перехід програми (направити розрахунок) на виконання будь-якого рядка, відміченого міткою.

Оператор безумовного переходу можна записати так:

```
GOTO m1          (перейти на рядок m1)
m1:              (набір команд)
```

В даному випадку **m1** – це мітка, тобто спеціальне позначення рядка, що має виконуватись. Оператор **GOTO** передає виконання програми на рядок, помічений **m1**. Рядок із міткою **m1** може бути розташований в будь-якій частині програми: на початку, вкінці, всередині. Після виконання набору команд рядка **m1**, виконується наступний рядок програми після оператора **GOTO**.

**Наприклад:**

```
Sub LR2_3 ()
Dim Z!, Y!, B!, X1!, X2!, A!, X!
  X1= InputBox("Введіть значення X1", "введення початкових даних")
  X2= InputBox("Введіть значення X2", "введення початкових даних")
  Y= InputBox("Введіть значення Y", "введення початкових даних")
  Z= InputBox("Введіть значення Z", "введення початкових даних")
Print "X1=", X1, "X2=", X2, "Y=", Y, "Z=", Z
  X=X1
m2:  If X<Y Then
      B=(вираз)
      Print "при X=",X, "B=",B
    Else
      A=(вираз)
```

```
Print "при X=",X, "A=",A
```

```
End If
```

```
If X=X2 Then Goto m1
```

```
X=X2
```

```
Goto m2
```

```
m1: Print "Програма успішно завершила роботу"
```

```
End Sub
```

Структура програми за цим прикладом наведена на рисунку 2.3. В цьому прикладі **m1** і **m2** – мітки, розташовані вкінці і на початку програми. Ця програма запитує у користувача дані (два варіанти значень змінної  $X$  –  $X1$  і  $X2$ , та змінних  $Y$  і  $Z$ ), потім підставляючи спочатку  $X1$  – обчислює вираз  $A$ , далі підставляє  $X2$  – обчислює вираз  $B$ ; виводить результати роз-

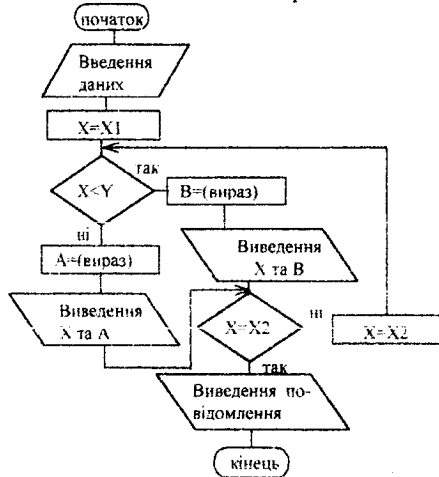


Рисунок 2.3 — Структура програми за лабораторною роботою №2.3

рахунків з відповідним  $X$  та повідомлення про успішне виконання програми.

Зверніть увагу на запис рядка резервування (оголошення) змінних:

```
Dim Z!, Y!, B!, X1!, X2!, A!, X!
```

За допомогою суфікса **!** ми вказали тип змінної (**Single**) і через розділовий знак – кому – перелічили всі змінні, що використовуються у нашій програмі.

Після введення даних  $X1$ ,  $X2$ ,  $Y$ ,  $Z$  ми бачимо рядок:

```
Print "X1=", X1, "X2=", X2, "Y=", Y, "Z=", Z
```

Для **VisualBasic-6** потрібно замість оператора **Print** ввести такий оператор: **MsgBox "X1=" & X1, "X2=" & X2, "Y=" & Y & "Z=" & Z, vbOkCancel,**

## "Перевірка початкових даних"

Цей рядок **треба обов'язково** вставляти в програму, він виводить на екран всі значення щойно введених вхідних даних для перевірки правильності введення.

Як видно із даного прикладу оператор безумовного переходу **GOTO** може застосовуватись окремо і в складі оператора умовного переходу If.  
**наприклад:**

**If X=X2 Then Goto m1**

**якщо виконується умова X=X2, то перейти на рядок m1**

(в даному випадку **m1** – виведення повідомлення про успішне завершення роботи програми).

Вищенаведена програма має таку структуру

Приклади програм для виконання у середовищі Microsoft Office наведено у додатку В.

### 2.3.4 Хід роботи

2.3.4.1 Завдання: Створити програму, яка б виконувала такі операції:

— запитувала у користувача початкові дані;

— виводила рядок перевірки початкових даних;

— підставляючи в розгалужувальні обчислення спочатку значення

X1, а потім X2, обраховувала відповідно вираз А чи В:

— виводила на екран результати обох варіантів обчислень;

— після обчислення А і В виводила повідомлення про закінчення роботи програми.

2.3.4.2 Отримавши свій варіант завдання уважно записати свій арифметичний вираз за правилами мови програмування.

2.3.4.3 Скласти схему алгоритму програми

2.3.4.4 Скласти два варіанти тексту програми 1-й із введенням вхідних даних оператором присвоєння Let, 2-й із введенням даних оператором Input.

2.3.4.5 Показати тексти програм викладачеві.

2.3.4.6 Запустити програму Open Office Word.

2.3.4.7 Зайти в меню Сервіс—Макрос—Організувати—Новий модуль.

2.3.4.8 Програму називаємо LR2\_3, клацнемо мишкою на команді правки.

2.3.4.9 Після введення коментарів (Rem), введення назви програми (Sub), набираємо текст програми.

2.3.4.10 Після набору програми збережемо файл з відповідною назвою.

2.3.4.11 Результат обчислень виразів а та b повинен збігатися з результатами обчислень лабораторної роботи №2.2.

### 2.3.5 Контрольні запитання

2.3.5.1 Чи можуть бути оператори умовного переходу вкладені один в другий? Приклад.

2.3.5.2 Коли застосовують оператор безумовного переходу?

- 2.3.5.3 Як запрограмувати перевірку введених вхідних даних? Наведіть приклад.
- 2.3.5.4 Що таке мітка? Де вона може бути розташована?
- 2.3.5.5 Нарисуйте схему алгоритму програми з оператором безумовного переходу.
- 2.3.5.6 Запишіть загальний вигляд оператора безумовного переходу.
- 2.3.5.7 Чи може оператор безумовного переходу застосовуватись у складі інших операторів? Приклад.
- 2.3.5.8 Наведіть приклад програми з використанням оператора безумовного переходу.
- 2.3.5.9 Коли застосовують оператор безумовного переходу?
- 2.3.5.10 Як запрограмувати перевірку введених вхідних даних? Наведіть приклад.
- 2.3.5.11 Напишіть програму з використанням оператора безумовного переходу у складі оператора IF.
- 2.3.5.12 Запишіть загальний вигляд оператора безумовного переходу.
- 2.3.5.13 Чи може оператор безумовного переходу застосовуватись у складі інших операторів? Приклад.
- 2.3.5.14 Який новий запис рядка резервування змінних ви знаєте? Запишіть.
- 2.3.5.15 Нарисуйте схему алгоритму програми з оператором безумовного переходу.
- 2.3.5.16 Коли застосовують оператор безумовного переходу?
- 2.3.5.17 Як запрограмувати перевірку введених вхідних даних? Наведіть приклад.

## 2.4 ЛАБОРАТОРНА РОБОТА №2.4

**2.4.1 Тема:** розв'язання задач з циклічним обчислювальним процесом Оператори IF... THEN, FOR... NEXT

**2.4.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic та розв'язання на ЕОМ задач з циклічним процесом.

### 2.4.3 Теоретичні відомості

**Цикл** – це оператор або блок операторів, які виконуються програмою багаторазово.

У Visual Basic існують два основних типи циклів – цикли з лічильником (з відомою кількістю повторювань) та цикли з умовою.

Цикли з відомою кількістю повторювань застосовуються тоді, коли необхідно виконати будь-які дії задане число разів.

Цикли з умовою застосовуються тоді, коли будь-які дії в програмі повинні повторюватися до тих пір, доки не буде виконана задана умова.

**Тілом циклу** називають набір команд (операторів) які повинні повторюватись задану кількість разів.

Для організації циклу використовують декілька різних способів.

**ПЕРШИЙ СПОСІБ.** Організація циклу за допомогою оператора умовного переходу If... Then вкінці тіла циклу (схема алгоритму на рис. 2.4).

**Приклад.** Проітабулювати функцію  $y=x+2$  на відрізку [2;10]  $n=10$  разів.

```
Sub LR4_1
```

```
Dim X0!, X1!, X2!, X!, Y!, i As Integer, n As Integer
```

```
X0= InputBox("Введіть значення початку інтервалу табулювання X0", _
```

```
"введення початкових даних")
```

```
n = InputBox("Введіть кількість обчислених значень функції n", _
```

```
"введення початкових даних")
```

```
X2= InputBox("Введіть значення кінця інтервалу табулювання X2", _
```

```
"введення початкових даних")
```

```
Print "X0=", X0, "X2=", X2, "n=", n
```

```
X1=(X2-X0)/(n-1) ' Обчислення кроку табулювання
```

```
X=X0 ' Надання змінній X початкового значення
```

```
i=0 ' Лічильник кількості елементів таблиці
```

```
m1:
```

```
Y=X+2
```

```
i=i+1
```

MsgBox "X = " & X & "; Y = " & Y, vbOKCancel, i & "-й елемент таблиці"

X=X+X1

If i<n Then GOTO m1

MsgBox"Цикл завершено", vbOKOnly,"Інформація про завершення Л.Р.4"

End Sub

В даному прикладі тіло циклу:

m1:

Y=X+2

i=i+1

Print " X=",X, "Y=",Y, "i=",i

X=X+X1

Оператор умовного переходу, **If i<n Then GOTO m1**, за допомогою якого організований циклічний процес, знаходиться вкінці тіла циклу.

**ДРУГИЙ СПОСІБ.** Організація циклу за допомогою оператора умовного переходу **If... Then** на початку тіла циклу (схема алгоритму на рис. 2.4).

**Приклад.** Протабулювати функцію  $y=x+2$  на відрізьку [2;10]  $n=10$  разів.

Sub LR4\_1

Dim X0!, X1!, X2!, X!, Y!, i As Integer, n As Integer

X0= InputBox("Введіть значення початку інтервалу табулювання X0", \_

"введення початкових даних")

n= InputBox("Введіть кількість обчислених значень функції n", \_  
"введення початкових даних")

X2= InputBox("Введіть значення кінця інтервалу табулювання X2", \_

"введення початкових даних")

Print "X0=", X0, "X2=", X2, "n=", n

X1=(X2-X0)/(n-1) ' Обчислення кроку табулювання

X=X0' Надання змінній X початкового значення

i=1 ' Лічильник кількості елементів таблиці

m1:

If i>n Then GOTO m2

Y=X+2

Print " X= ";X, "Y= ";Y, i"-й елемент розрахованої таблиці",

X=X+X1

i=i+1

GOTO m1

m2:

MsgBox"Цикл завершено", vbOKOnly,"Інформація про завершення Л.Р.4"

## End Sub

Цикли з лічильником (з відомою кількістю повторювань) у Visual Basic ще називають циклом For... Next.

Оператор циклу можна записати так:

**For X=(початкове значення) To (кінцеве значення X) Step (значення кроку табулювання)**

(тіло циклу)

**Next X**

Перекласти можна так:

Для X=(початкове значення) До (кінцеве значення) Крок (значення кроку)

(тіло циклу)

Наступне X

Для нашого прикладу оператор циклу виглядає так:

**For X=X0 To X2 Step (X2-X0)/(n-1)**

Приклад програми з використанням оператора циклу (схема алгоритму на рис. 3).

**Sub LR4\_1**

**Dim X0!, X1!, X2!, X!, Y!, i As Integer, n As Integer**

**X0= InputBox("Введіть значення початку інтервалу табулювання X0", \_**

**"введення початкових даних")**

**n= InputBox("Введіть кількість обчислених значень функції n", \_**

**"введення початкових даних")**

**X2= InputBox("Введіть значення кінця інтервалу табулювання X2", \_**

**Print "X0=", X0, "X2=", X2, "n=", n**

**i=0** *Лічильник кількості елементів таблиці*

**For X=X0 To X2 Step (X2-X0)/(n-1)**

**Y=X+2**

**i=i+1**

**Print " X=";X, " Y=";Y, " i=";i**

**Next X**

**MsgBox"Цикл завершено", vbOKOnly,"Інформація про завершення Л.Р.4"**

**End Sub**

Як ми бачимо ті рядки програми з оператором If, що надають змінній початкового значення і визначають крок табулювання замінюються рядком

**For X=X0 To X2 Step (X2-X0)/(n-1),**

а оператори умовного та безумовного переходу, які забезпечували циклічність програми замінюються оператором **Next X** тобто наступне обчислення X з новим значенням.



Приклади програм для виконання у середовищі Microsoft Office наведено у додатку Г.

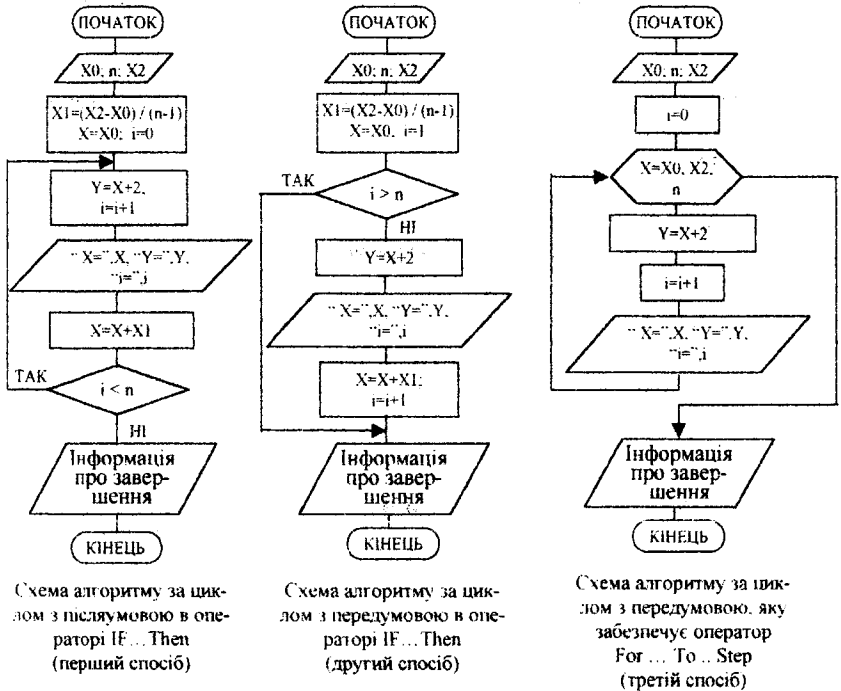


Рисунок 2.4 — Структури програми за лабораторною роботою №2.4

## 2.4.4 Хід роботи

- 2.4.4.1 Завдання: протабулювати функцію  $y=f(x)$  на даному відрізку  $[x_{\min}; x_{\max}]$   $p$  разів
- 2.4.4.2 Отримавши свій варіант завдання уважно записати свій арифметичний вираз за правилами мови програмування.
- 2.4.4.3 Скласти схему алгоритму програми
- 2.4.4.4 Скласти три варіанти тексту програми 1-й з оператором If... Then на початку тіла циклу, 2-й з оператором If...Then вкінці тіла циклу, а 3-й з оператором For... Next.
- 2.4.4.5 Показати тексти програм викладачеві.
- 2.4.4.6 Запустити програму OpenOffice Word (Microsoft Excel, Star Office).
- 2.4.4.7 Зайти в меню Сервіс—Макрос—Організувати—Новий модуль.
- 2.4.4.8 Програму називаємо LR4, клацнемо мишкою на команді правки.

2.4.4.9 Після введення коментарів (Rem), введення назви програми (Sub), набираємо текст програми.

2.4.4.10 Після набору програми збережемо файл з відповідною назвою.

### **2.4.5 Контрольні запитання**

2.4.5.1 Що таке цикл?

2.4.5.2 Два способи організації циклу?

2.4.5.3 Чи можна з допомогою оператора IF організувати циклічні обчислення? Приклад.

2.4.5.4 Два основні типи циклів?

2.4.5.5 Що називається тілом циклу?

2.4.5.6 Загальний вигляд оператора циклу.

2.4.5.7 Коли застосовуються цикли з умовою?

2.4.5.8 Наведіть приклад програми із застосуванням оператора FOR.

2.4.5.9 Що значить протабулювати функцію?

2.4.5.10 Коли застосовують цикли з відомою кількістю повторювань?

2.4.5.11 Чи можна з допомогою оператора IF організувати циклічні обчислення? Приклад.

## 2.5 ЛАБОРАТОРНА РОБОТА №2.5

**2.5.1 Тема:** Розв'язання задач з циклічним обчислювальним процесом Оператори DO...LOOP

**2.5.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic та розв'язання на ЕОМ задач з циклічним процесом.

### 2.5.3 Теоретичні відомості

В циклах з умовою типу **DO...LOOP** група операторів виконується доти, доки виконується умова (або не виконується).

Існує **два типи** циклів з умовою **DO...LOOP**:  
**DO...While LOOP** і **DO...Until LOOP**

Цикл DO...While – виконується до тих пір, поки умова є істинною. Цикли **DO...While** бувають з передумовою (умова стоїть на початку циклу) і післяумовою (умова знаходиться в кінці циклу).

Цикли **DO...While** з передумовою має вигляд:

```
DO While (умова)  
(оператори або група обчислюваних рівнянь)  
LOOP
```

Цикли **DO...While** з післяумовою виглядають так:

```
DO  
(оператори або група обчислюваних рівнянь)  
LOOP While (умова)
```

Цикл DO...Until – виконується до тих пір, поки умова є хибною. Цикли **DO...Until** також бувають з передумовою і післяумовою

Цикли **DO...Until** з передумовою можна записати таким чином:

```
DO Until (умова)  
(оператори або група обчислюваних рівнянь)  
LOOP
```

Цикли **DO...Until** з післяумовою мають вигляд:

**DO**

(оператори або група обчислюваних рівнянь)

**LOOP Until** (умова)

**DO** – службове слово, що вказує на початок циклу.

**While/ Until** – службові слова, вони визначають тип циклу,

**LOOP** – службове слово, що вказує на кінець циклу.

Приклад програми DO While ... LOOP з передумовою.

Sub LR5\_1

Dim X0!, X1!, X2!, X!, Y!, i As Integer, n As Integer

X0= InputBox("Введіть початок інтервалу табулювання X0", \_  
"введення початкових даних")

n= InputBox("Введіть кількість обчислених значень функції n", \_  
"введення початкових даних")

X2= InputBox("Введіть кінець інтервалу табулювання X2", \_  
"введення початкових даних")

Print "X0=", X0, "X2=", X2, "n=", n

X1=(X2-X0)/(n-1)      *Обчислення кроку табулювання*

X=X0      *Надання змінній X початкового значення*

i=1      *Лічильник кількості елементів таблиці*

**DO While** X<=X2

    Y=X+2

    Print " X=",X, "Y=",Y, "i=",i

    i=i+1

    X=X+X1

**LOOP**

MsgBox"Цикл завершено", vbOKOnly, \_

    "Інформація про завершення Л.Р.4"

End Sub

Приклад програми DO Until ... LOOP з післяумовою:

Sub LR5\_4

Dim X0!, X1!, X2!, X!, Y!, i As Integer, n As Integer

X0= InputBox("Введіть початок інтервалу табулювання X0", \_  
"введення початкових даних")

n= InputBox("Введіть кількість обчислених значень функції n", \_  
"введення початкових даних")

X2= InputBox("Введіть значення кінця інтервалу табулювання X2", \_  
"введення початкових даних")

Print "X0=", X0, "X2=", X2, "n=", n

X1=(X2-X0)/(n-1)	<i>'Обчислення кроку табулювання</i>
X=X0	<i>'Надання змінній X початкового значення</i>
i=1	<i>'Лічильник кількості елементів таблиці</i>

DO

Y=X+2

Print " X=",X, " Y=",Y, " i=",i

i=i+1

X=X+X1

LOOP Until X>X2

MsgBox"Цикл завершено", vbOKOnly, \_

"Інформація про завершення лабораторної роботи"

End Sub

Приклади програм для виконання у середовищі Microsoft Office наведено у додатку Д.

### 2.5.4 Хід роботи

2.5.4.1. Завдання: протабулювати функцію  $y=f(x)$  на даному відрізку  $[x_{\min}; x_{\max}]$   $n$  разів.

2.5.4.2. Отримавши свій варіант завдання уважно записати свій арифметичний вираз за правилами мови програмування.

2.5.4.3. Скласти схему алгоритму програми

2.5.4.4. Скласти чотири варіанти тексту програми:

— з оператором **DO While ... LOOP** з передумовою і післяумовою;

— з оператором **DO Until ... LOOP** з передумовою і післяумовою

2.5.4.5. Показати тексти програм викладачеві.

2.5.4.6. Запустити програму OpenOffice Word.

2.5.4.7. Зайти в меню Сервіс—Макрос—Організувати—Новий модуль.

2.5.4.8. Програму називаємо LR5 1, клацнемо мишкою на команді правки.

2.5.4.9. Після введення коментарів (Rem), введення назви програми (Sub), набирасмо текст програми.

2.5.4.10. Після набору програми збережемо файл з відповідною назвою.

**2.5.4.11. РЕЗУЛЬТАТИ ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ №2.5 І №2.4 ПОВИННІ ЗБІГАТИСЯ!**

### 2.5.5 Контрольні запитання

2.5.5.1 Які різновиди операторів циклу **DO...LOOP** є в системі програмування Visual Basic?

2.5.5.2 Поясніть роботу оператора **DO...While LOOP**.

2.5.5.3 Поясніть роботу оператора **DO...Until LOOP**.

2.5.5.4 Цикли **DO...While** з передумовою.

2.5.5.5 Цикли **DO...While** з післяумовою.

2.5.5.6 Цикли **DO... Until** з передумовою.

2.5.5.7 Цикли **DO... Until** з післяумовою.

2.5.5.8 Навести та пояснити приклад програми за циклом **DO While ... LOOP** з передумовою.

2.5.5.9 Навести та пояснити приклад програми за циклом **DO Until ... LOOP** з післяумовою.

2.5.5.10 Організація циклів у Visual Basic-програмах за допомогою операторів **IF...THEN**.

2.5.5.11 Організація циклів у Visual Basic-програмах за допомогою операторів **For ... To .. Step**.

## 2.6 ЛАБОРАТОРНА РОБОТА №2.6

**2.6.1 Тема:** Розв'язання задач з повторюваними однотипними обчислювальними процедурами

**2.6.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic зі створенням та використанням підпрограм користувача.

### 2.6.3 Теоретичні відомості

Підпрограма – це ділянка програми, яка складається з послідовності операторів, багаторазово використовуваних у програмі.

Підпрограма записується у кінці програми.

Для виконання підпрограми досить звернутися до неї за допомогою оператора **GOSUB**. Виведення з підпрограми відбувається при виконанні оператора **RETURN**.

В одній програмі може бути декілька підпрограм.

Загальний вигляд оператора звернення до підпрограми:

**GOSUB m220** (виконати рядок m220)

m220:

(набір команд (операторів))

**RETURN**

Після виконання підпрограми (яка починається з рядочка m220), оператор **RETURN** повертає хід виконання програми до оператора, що розташований після того **GOSUB**, з якого було звернення до підпрограми.

Підпрограма може вмещати в собі звернення до іншої підпрограми і т.д.

Рекурсивне звернення, тобто звернення підпрограми до самої себе, не дозволяється.

Приклад: скласти підпрограму обчислення функції

$$D = \frac{\operatorname{th}(an + b)}{\operatorname{th}^2(a^2 + b^2)} - \frac{\operatorname{th}(a + mb)}{\operatorname{th}^2(a^2 - b^2)}, \text{ при цьому обчислення гіпербо-$$

лічного тангенсу оформити у вигляді підпрограми  $\operatorname{th}(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$ .

**Sub LR6 ()**

**Dim A!, B!, N!, M!, T!, X!, D!, T1!, T2!, T3!, T4!**

**A= InputBox(“Введіть значення”)**

**B= InputBox(“Введіть значення B”)**

**N= InputBox(“Введіть значення N”)**

**M= InputBox(“Введіть значення M”)**

**X=A\*N+B**

**GOSUB m220**

```

T1=T
X=A^2+B^2
GOSUB m220
T2=T
X=A+M*B
GOSUB m220
T3=T
X=A^2-B^2
GOSUB m220
T4=T
D=T1/T2^2-T3/T4
Print "D="D
GOTO EN
***** Підпрограма (початок) *****
M220:
    T=(EXP(2*X)-1)/(EXP(2*X)+1)
RETURN
***** Підпрограма (кінець) *****
En:
End Sub

```

Приклад коду програм до лабораторної роботи №2.6 для виконання в Microsoft Excel наводиться у додатку F.

### 2.6.4 Хід роботи

- 2.6.4.1. Отримати завдання.
- 2.6.4.2. Виділити ту частину, яку доцільно оформити у вигляді окремої підпрограми.
- 2.6.4.3. Скласти схему алгоритму програми
- 2.6.4.4. Показати тексти програм викладачеві.
- 2.6.4.6. Запустити програму OpenOffice Word, (або іншу програму з Office, у тому числі з Microsoft Office)
- 2.6.4.7. Зайти в меню Сервіс—Макрос—Організувати—Новий модуль.
- 2.6.4.8. Програму називаємо LR6, клацнемо мишкою на команді правки.
- 2.6.4.9. Після введення коментарів (Rem або апостроф), введення назви програми (Sub), набираємо текст програми.
- 2.6.4.10. Після набору програми збережемо файл з відповідною назвою.
- 2.6.4.11. Налаштувати програму та показати її роботу викладачеві для зарахування.

### 2.6.5 Контрольні запитання

- 2.6.5.1 Що таке підпрограма у Visual Basic-програми?



- 2.6.5.2 У якому місці записується підпрограма у тексті Visual Basic-програми?
- 2.6.5.3 Як організувати звернення до підпрограми з будь-якого місця Visual Basic-програми?
- 2.6.5.4 Як забезпечується вихід із підпрограми у Visual Basic-програмі?
- 2.6.5.5 Поясніть роботу оператора **GOSUB**.
- 2.6.5.6 Поясніть роботу оператора **RETURN**.
- 2.6.5.7 Чи може підпрограма містити у собі звернення до іншої підпрограми?
- 2.6.5.8 Що таке рекурсивне звернення до підпрограми і чи дозволяється воно у Visual Basic-програмах?

## 2.7 ЛАБОРАТОРНА РОБОТА №2.7

**2.7.1 Тема:** Розв'язання задач з повторюваними однотипними обчислювальними процедурами.

**2.7.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic зі створенням та використанням функцій.

### 2.7.3 Теоретичні відомості

Функція – це блок програми, що виконує певні операції в межах свого блоку і повертає значення.

Загальний вигляд оформлення функції:

**Function Назва функції(перелік параметрів) As Single**  
*Тіло функції (певні операції)*

**End Function**

Наприклад:

**Function T(X) As Single**

$$T=(\text{EXP}(2*X)-1)/(\text{EXP}(2*X)+1)$$

**End Function**

**Function** – ключове слово, воно вказує на те, що цей блок програми є функцією.

**Назвою функції** може бути будь-який ідентифікатор, в дужках вказується перелік змінних, від яких залежить функція (її формальні параметри).

**As Single** – вказується тип значення яке отримуємо в результаті виконання функції, в даному випадку результату розрахунку рівняння T.

**End Function** – службові слова, що вказують на закінчення блоку функції.

Щоб отримати результат виконання функції, необхідно звернутися до неї. Для цього вказується **назва функції(перелік параметрів)**, наприклад:

$$P=T(A*N+B)$$

У цьому випадку змінній **P** надається результат виконання функції T(X), іншими словами – розрахунку рівняння  $T=(\text{EXP}(2*X)-1)/(\text{EXP}(2*X)+1)$ , де замість X програма приймає значення  $A*N+B$ , тобто  $X=A*N+B$

Функція записується після програми, тобто вона може не бути частиною основної програми.

Приклад: скласти програму обчислення функції

$$D = \frac{\text{th}(an + b)}{\text{th}^2(a^2 + b^2)} - \frac{\text{th}(a + mb)}{\text{th}^2(a^2 - b^2)}, \text{ при цьому обчислення гіпербо-$$

лічного тангенсу оформити у вигляді функції  $\text{th}(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$ .

**Sub LR7 ()**

**Dim A!, B!, N!, M!, X!, D!**

```

A= InputBox("Введіть значення")
B= InputBox("Введіть значення B")
N= InputBox("Введіть значення N")
M= InputBox("Введіть значення M")
Print "A=",A,"B=",B"N=",N,"M=",M
D=T(A*N+B)/T(A^2+B^2)^2-T(A+M*B)/T(A^2-B^2)^2
Print "D=",D
End Sub
*****Функція*****
Function T(X) As Single
    T=(EXP(2*X)-1)/(EXP(2*X)+1)
End Function

```

Зверніть увагу на рядок резервування (оголошення) змінних!

Серед переліку (Dim A!, B!, N!, M!, X!, D!) немає ідентифікатора (назви) функції T, тобто назву функції в операторі Dim визначати не потрібно.

Приклад коду програм до лабораторної роботи №2 7 для виконання в Microsoft Excel наводиться у додатку Ж.

## 2.7.4 Хід роботи

- 2.7.4.1 Отримати індивідуальне завдання.
- 2.7.4.2 Виділити ту частину, яку доцільно оформити у вигляді окремої функції.
- 2.7.4.3 Скласти схему алгоритму програми
- 2.7.4.4 Показати тексти програм викладачеві.
- 2.7.4.6 Запустити програму Open Office Word, (або іншу програму з Office, у тому числі з Microsoft Office)
- 2.7.4.7 Зайти в меню Сервіс—Макрос—Організувати—Новий модуль.
- 2.7.4.8 Програму називаємо LR7\_1 , клацнемо мишкою на команді правки.
- 2.7.4.9 Після введення коментарів (Rcm або апостроф), введення назви програми (Sub), набираємо текст програми.
- 2.7.4.10 Після набору програми збережемо файл з відповідною назвою.
- 2.7.4.11 Налаштувати програму та показати її роботу викладачеві для зарахування.

## 2.7.5 Контрольні запитання

- 2.7.5.1 Що таке функція у Visual Basic-програми?
- 2.7.5.2 Правила оформлення тексту (коду) функції у Visual Basic-програми.
- 2.7.5.3 Правила складання назв функцій у Visual Basic-програми.
- 2.7.5.4 Як позначається тип функції у тексті Visual Basic-програми?
- 2.7.5.5 Правила звернення до функції у Visual Basic-програми.
- 2.7.5.6 Правила запису функції у Visual Basic-програми.

## 2.8 ЛАБОРАТОРНА РОБОТА №2.8

**2.8.1 Тема:** Розв'язання задач з повторюваними однотипними обчислювальними процедурами

**2.8.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic зі створенням та використанням підпрограм – процедур.

### 2.8.3 Теоретичні відомості

Підпрограма–процедура – це блок програми, що виконує певні операції в межах свого блоку і, на відміну від функції, може повертати одне або декілька значень–результатів у програму (процедуру) з якої було звернення.

Загальний вигляд оформлення підпрограми–процедури:

**Sub Назва\_процедури (перелік параметрів)**

*Тіло підпрограми–процедури (певні операції)*

**End Sub**

Наприклад:

**Sub Trp(X As Single, T As Single)**

$T = (\text{EXP}(2 * X) - 1) / (\text{EXP}(2 * X) + 1)$

**End Sub**

**Sub** – ключове слово, воно вказує на те, що цей блок програми є процедура.

**Назвою процедури** може бути будь-який ідентифікатор, в дужках вказується перелік змінних, від яких залежить результат роботи підпрограми–процедури (її формальні параметри). У цьому переліку зазначається і тип формальних параметрів підпрограми–процедури. У наведеному прикладі підпрограма–функція має ім'я **Trp**

**As Single** – вказується тип формальних параметрів, в даному випадку змінної **X** та результату розрахунку рівняння **T**.

**End Sub** – службові слова, що вказують на закінчення блоку підпрограми–процедури.

Щоб виконати підпрограму–процедуру, необхідно звернутися до неї. Для цього вказується **назва підпрограм–процедури (перелік фактичних параметрів)**, наприклад:

**Call Trp (A\*N+B, P)**

або

**Trp A\*N+B, P**

В цьому випадку змінній **P** надається результат обчислення **T(X)**, іншими словами – розрахунку рівняння  $T = (\text{EXP}(2 * X) - 1) / (\text{EXP}(2 * X) + 1)$ , де **замість X програма приймає значення A\*N+B**, тобто **X = A\*N+B**

Підпрограма–процедура записується у будь-якому місці коду (тексту) програми (програмного модуля), тобто вона може не бути частиною основної програми.

Приклад: скласти програму обчислення функції

$$D = \frac{\text{th}(an + b)}{\text{th}^2(a^2 + b^2)} - \frac{\text{th}(a + mb)}{\text{th}^2(a^2 - b^2)}$$
 у трьох варіантах, при цьому

обчислення гіперболічного тангенсу  $\text{th}(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$  оформити у вигляді

підпрограми—процедури.

Sub LR8 ()

Dim A!, B!, N!, M!, X!, D!, P!

A= InputBox("Введіть значення А")

B= InputBox("Введіть значення В")

N= InputBox("Введіть значення N")

M= InputBox("Введіть значення M")

Print "A=",A, "B=",B,"N=",N,"M=",M

Trp A \* N + B, P

T1 = P

Trp A ^ 2 + B ^ 2, P

T2 = P

Trp A + M \* B, P

T3 = P

Trp A ^ 2 - B ^ 2, P

T4 = P

D = T1 / T2 ^ 2 - T3 / T4

MsgBox "D = " & D, vbOKCancel, "Результат 1-шого варіанта 8-ї роботи"

End Sub

Private Sub Trp(X As Single, T As Single)

T = (Exp(2 \* X) - 1) / (Exp(2 \* X) + 1)

End Sub

Приклад коду програм до лабораторної роботи №2 8 для виконання в Microsoft Excel наводиться у додатку Й.

## 2.8.4 Хід роботи

2.8.4.1 Отримати індивідуальне завдання.

2.8.4.2 Виділити ту частину, яку доцільно оформити у вигляді окремої підпрограми—процедури.

2.8.4.3 Скласти схему алгоритму програми

2.8.4.4 Показати тексти програм викладачеві.

2.8.4.5 Запустити програму Open Office Word, (або іншу програму з Office, у тому числі з Microsoft Office)

2.8.4.6 Зайти в меню Сервіс—Макрос—Організувати—Новий модуль.

2.8.4.7 Програму називаємо LR8\_1, клацнемо мишкою на команді правки.

2.8.4.8 Після введення коментарів (Rem або апостроф), введення назви програми (Sub), набираємо текст програми.

2.8.4.9 Після набору програми збережемо файл з відповідною назвою.

2.8.4.10 Налаштувати програму та показати її роботу викладачеві для зарахування.

2.8.4.11 Повторити пункти 5 – 10 для решти варіантів програми.

### **2.8.5 Контрольні запитання**

2.8.5.1 Що таке підпрограма-процедура у Visual Basic-програмах?

2.8.5.2 Яка відмінність підпрограми-процедури від функції у Visual Basic-програмах?

2.8.5.3 Який загальний вигляд оформлення підпрограми-процедури у Visual Basic-програмах?

2.8.5.4 Як оформлюється назва підпрограми-процедури у Visual Basic-програмах?

2.8.5.5 Які правила звернення до підпрограми-процедури у Visual Basic-програмах?

2.8.5.6 У якому місці тексту програми (програмного модуля) записується підпрограма-процедура?

## 2.9 ЛАБОРАТОРНА РОБОТА №2.9

### 2.9.1 Тема: Програмування з виведенням даних у файл

**2.9.2 Мета роботи:** набуття практичних навичок програмування мовою Visual Basic з використанням виведення результатів у файл

### 2.9.3 Теоретичні відомості

Для виведення у файл потрібно попередньо відкрити цей файл оператором **Open** формат (синтаксис) запису такий:

Open "Повне ім'я\_файла" For Append As #номер\_файла\_(каналу виведення)

Де **Open** та **For Append As** — це службові слова Visual Basic. Якщо зазначеного у цьому операторі файла не існує, то він буде створений під час виконання оператора **Open**.

Слово **Append** або **Output** означає, що файл відкривається для запису даних у його кінець, тобто відкривається файл послідовного доступу.

Для запису даних у файл можна використовувати оператор **Print** формат запису якого такий:

Print #номер\_файла, список\_виведення

Де *номер\_файла* — зазначений в операторі **Open** номер файла. *Список виведення* — вирази та конструкції типу "літерал", розділені комою або крапкою з комою.

Після роботи з файлом потрібно обов'язково його закрити командою:

Close # номер\_файла

Для перегляду із програми створеного файла (після закриття його командою **Close**), наприклад текстовим редактором Блокнот (Notepad) можна скористатись функцією:

V = Shell("Notepad ім'я\_файла", vbNormalFocus)

Ця функція відкриває текстовий редактор Notepad і звантажує в його вікно файл із зазначеним в операторі ім'ям. Після перегляду відкритого таким чином файла для закінчення (продовження) роботи програми потрібно закрити вікно текстового редактора. Змінна V може бути будь-якого типу. Вона формально використовується лише для активізації функції **Shell**.

Для знищення створеного програмою файла можна використати оператор:

Kill "Повне\_ім'я\_файла"

## ПОСТАНОВКА ЗАДАЧІ

Організувати роботу усіх програм попередніх лабораторних робіт (№2.1—2.8) з виведенням результатів у файл. При цьому обов'язково створені програмами файли повинні знищуватись оператором **Kill**

### 2.9.4 Хід роботи

2.9.4.1 Доповнити тексти програм усіх попередніх лабораторних робіт (№2.1—2.8) операторами роботи з файлами.

2.9.4.2 Після коректування програм відлагодити їх роботу, продемонструвати викладачеві та зберегти у папці дев'ятої лабораторної роботи з відповідною назвою.

### 2.9.5 Контрольні запитання

2.9.5.1 Який оператор Visual Basic використовується для відкриття файлу з метою виведення в нього?

2.9.5.2 Які слова використовуються для забезпечення виведення в його кінець?

2.9.5.3 Що таке файл послідовного доступу?

2.9.5.4 Який оператор можна використати для запису даних в його кінець?

2.9.5.5 Що таке номер файла?

2.9.5.6 Що таке список виведення у файл?

2.9.5.7 Як закрити файл після роботи з ним?

2.9.5.8 Якою функцією можна скористатися для програмування перегляду вмісту файла? Як цю функцію записати?

2.9.5.9 Як запрограмувати знищення створеного у програмі файла?

### 2.9.6 Приклади програм для роботи в Microsoft Office

Приклади програм за лінійною структурою.

#### Sub ЛабРоб\_2\_9\_1()

' Тема : "Робота з файлами (Лінійний обчислювальний процес)"

' Рівнозначні варіанти оголошення змінних

'(можна , але не бажано взагалі не оголошувати)

Dim A!, X!, B!, Y!, V!

'Dim A As Single, X As Single, B As Single, Y As Single

A = 6.52

B = 3.73

X = 0.75

Y = Sqr(1 + Sin(X / 2) ^ 2) / (A + B \* Log(X + 3))

' Відкриття файла D:\Rez.txt для запису в нього

Open "D:\Rez.txt" For Append As #1



```

' Виведення у відкритий файл результату роботи програми
  Print #1, "Y="; Y
' Закривання файла з виведеним результатом
  Close #1
' Перегляд файла з результатом текстовим редактором Notepad (Блокнот)
  V = Shell("Notepad D:\Rez.txt", vbNormalFocus)
' Знищення створеного у даній програмі файла (файла D:\Rez.txt)
  Kill "D:\Rez.txt"

```

**End Sub**

### **Sub ЛабРоб\_2\_9\_2()**

```

' Тема : "Робота з файлами (Лінійний обчислювальний процес, )"
  Dim A As Single, X As Single, B As Single, Y As Single, V As Single
' Відкриття файла D:\Rez.txt для введення даних
  Open "D:\Rez.txt" For Append As #1
  Print #1, "Вихідні дані розрахунку:"
  A = Application.InputBox("Введіть значення А", "Введення початкових
данях")
' Запис у файл D:\Rez.txt значення для А
  Print #1, "A="; A
  B = Application.InputBox("Введіть значення В", "Введення початкових
данях")
  ' Запис у файл D:\Rez.txt значення для В
  Print #1, "B="; B
  X = Application.InputBox("Введіть значення X", "Введення початкових
данях")
' Запис у файл D:\Rez.txt значення для X
  Print #1, "X="; X
  Y = Sqr((1 + Sin(X / 2) ^ 2) / (A + B * Log(X + 3)))
  ' Запис у файл D:\Rez.txt результату розрахунку
  Print #1, "Результат розрахунку:", "Y = "; Y
' Закриваємо файл для його перегляду
  Close #1
' Перегляд файла D:\Rez.txt за допомогою програми "Блокнот" (Notepad)
  V = Shell("Notepad D:\Rez.txt", vbNormalFocus)
'Знищення файла D:\Rez.txt
  Kill "D:\Rez.txt"

```

**End Sub**

Приклади програм за структурою розгалуження

### **Sub ЛабРоб\_2\_9\_3\_1()**

' Структура розгалуження з використанням операторів умовного (If... Then...Else) та безумовного (Goto) переходу

```
Dim X!, R!, Y!, Z!, X1!, X2!, V As Single
X1 = Application.InputBox("Введіть значення для X1", _
    "Введення початкових даних")
X2 = Application.InputBox("Введіть значення для X2", _
    "Введення початкових даних")
R = Application.InputBox("Введіть значення для R", _
    "Введення початкових даних")
    Open "D:\Rez.txt" For Output As #1
    Print #1, "Вихідні дані розрахунку:"
    Print #1, "X1= "; X1
    Print #1, "X2= "; X2
    Print #1, "R= "; R
    Print #1, "Результат розрахунку:"
X = X1
m2:
If X <= R Then
    Y = (X ^ 2 - R ^ 2) / X
    Print #1, "Y= "; Y, "при X="; X
Else
    Z = Atn(X - R)
    Print #1, "Z= "; Z, "при X="; X
End If

If X = X2 Then GoTo m1
X = X2
GoTo m2
m1:
Close #1
V = Shell("Notepad D:\Rez.txt", vbNormalFocus)
Kill "D:\Rez.txt"
End Sub
```

### Sub ЛабРоб\_2\_9\_3\_2()

' Структура розгалуження з використанням оператора вибору Select Case та оператора умовного переходу If... Then з виведенням у файл

```
Dim X!, R!, Y!, Z!, X1!, X2!, S As Integer, V As Single
```

```
X1 = Application.InputBox("Введіть значення для X1", _
```

```

                                "Введення початкових даних")
X2 = Application.InputBox("Введіть значення для X2",_
                                "Введення початкових даних")
R = Application.InputBox("Введіть значення для R",_
                                "Введення початкових даних")
    Open "D:\Rez.txt" For Output As #1
    Print #1, "Вихідні дані розрахунку:"
    Print #1, "X1= "; X1
    Print #1, "X2= "; X2
    Print #1, "R= "; R
    Print #1, "Результат розрахунку:"
    X = X1
m2:
    If X <= R Then S = 1 Else S = 2

    Select Case S
        Case 1
            Y = (X ^ 2 - R ^ 2) / X
            Print #1, "Y= "; Y, "при X="; X
        Case 2
            Z = Atn(X - R)
            Print #1, "Z= "; Z, "при X="; X
    End Select

    If X = X2 Then GoTo m1
    X = X2
    GoTo m2
m1:
    Close #1
    V = Shell("Notepad D:\Rez.txt", vbNormalFocus)
    Kill "D:\Rez.txt"
End Sub

```

Приклади програм за циклічною структурою

**Sub Lr\_2\_9\_4\_2()**

' Варіант програми з використанням оператора For...Next для організації циклу

Dim X0 As Single, X1 As Single, X2 As Single, i As Integer, X As Single

Dim Y As Single, n As Integer, V As Single

X0 = Application.InputBox("Введіть початкове значення аргументу X0", \_  
 " 4-та лабораторна - цикл з оператором For...Next")

```

n = Application.InputBox("Введіть кількість значень функції (n)", _
    " 4-та лабораторна - цикл з оператором For...Next")
X2 = Application.InputBox("Введіть кінцеве значення аргументу X2", _
    "Цикл з оператором For...Next")
Open "D:\Rez.txt" For Output As #1
    Print #1, "Вихідні дані розрахунку:"
Print #1, "X0= "; X0, " --- початок інтервалу табулювання;"
Print #1, "n= "; n, " --- кількість розрахованих значень функції;"
Print #1, "X2= "; X2, " --- кінець інтервалу табулювання;"
Print #1, "Результат розрахунку:"
i = 0          ' Лічильник кількості елементів таблиці

For X = X0 To X2 Step (X2 - X0) / (n - 1) ' крок табулювання_
                                         обчислюється тут
    Y = Sqr(0.2 * X ^ 4 + 4 * Sqr(X))
    i = i + 1
    If i > n + 10 Then Exit For
Print #1, "X = "; X, "Y = "; Y, i, " -й елемент таблиці"
Next X

Close #1
V = Shell("Notepad D:\Rez.txt", vbNormalFocus)
Kill "D:\Rez.txt"
End Sub

```

## ЛИТЕРАТУРА

- 1 Алексеев А. П. Информатика — Солон-Р. Москва, 2001. — 364 с.
- 2 Информатика: Базовый курс / С.В. Симонович и др. — СПб.: Питер, 2001. — 640 с.
- 3 Руденко В.Д. Збірник практичних робіт з інформатики / За ред. Мадзігона В.М. — К.: Видавнича група "ВНУ", 1999. — 96 с.
- 4 Аладьев В.З., Хунт Ю.Я., Шишаков М.Л. Основы информатики: Учебное пособие. — М.: "Филинь", 1998. — 496 с.
- 5 Дьяконов В.П. Internet. Настольная книга пользователя. — М.: Солон-Р, 1999. — 576 с.
- 6 Информационные системы в экономике: Учебник / Под ред. В.В. Дика. — М.: Финансы и статистика, 1996. — 272 с.
- 7 Компьютерные системы и сети / Под ред. В.П. Косарева и Л.В. Ярёмина — М.: Финансы и статистика, 1999. — 464 с.
- 8 Стинсон К. Эффективная работа в Windows 98. — СПб.: Питер, 1999, 784 с.
- 9 Стоцкий Ю. Самоучитель Offis 2000. СПб.: Питер, 1999, — 576 с.
- 10 К. Гетц. М. Джилберт. Программирование в Microsoft Offis: пер. с англ. — К.: Издательская группа ВНУ. 2000. — 768 с.
- 11 Англо-українсько-російський словник з інформатики та обчислювальної техніки. — Львів: Світоч, 1991. — 448 с.
- 12 Грошев С.В., Коцюбинский А.О. Современный самоучитель профессиональной работы на компьютере: Практ. пособ. — "Триумф", 1999. — 368 с.
- 13 Информатика: Комп'ютерна техніка. Комп'ютерні технології / За ред. О.І. Пушкаря. — К.: Видавничий центр "Академія", 2001. — 696 с.
- 14 Коляда М.Г. Окно в удивительный мир информатики. — Д.: Стапкер, 1999. — 448 с.
- 15 Самоучитель работы на персональном компьютере / Под ред. Ю.С.Ковтансона. — К.: Юниор, 2001. — 560 с.
- 16 Симонович С.В. и др. Специальная информатика: Учеб. Пособ. — М.: АСТ-ПРЕСС КНИГА, 2000. — 480 с.
- 17 Фигурнов В.Э. IBM PC для пользователя. Издание 6-е, переработанное и дополненное. — М.: ИНФРА-М, 1995. — 432 с.

## Додаток А

### Приклади програм до лабораторної роботи №2.1 для виконання в середовищі Microsoft Office

1-й варіант програми

#### Sub ЛабРоб\_2\_1\_1()

```
' Тема : "Лінійний обчислювальний процес"  
' Рівнозначні варіанти оголошення змінних  
'(можна , але не бажано взагалі не оголошувати)  
Dim A!, X!, B!, Y!  
'Dim A As Single, X As Single, B As Single, Y As Single  
A = 6.52  
B = 3.73  
X = 0.75  
Y = Sqr((1 + Sin(X / 2) ^ 2) / (A + B * Log(X + 3)))  
MsgBox " Y= " _  
    & Y, vbOKOnly, "Результат за першою лабораторною роботою"  
End Sub
```

2-й варіант програми до 6-ї лабораторної роботи

#### Sub ЛабРоб\_2\_1\_2()

```
' Тема : "Лінійний обчислювальний процес"  
' Рівнозначні варіанти оголошення змінних  
'(можна , але не бажано взагалі не оголошувати)  
'Dim A!, X!, B!, Y!  
Dim A As Single, X As Single, B As Single, Y As Single  
A = Application.InputBox("Введіть значення А", "Введення початкових  
даних")  
B = Application.InputBox("Введіть значення В", "Введення початкових  
даних")  
X = Application.InputBox("Введіть значення Х", "Введення початкових  
даних")  
' Перевірка введених даних
```

```
MsgBox " A=" & A & "; B =" & B & "; X =" & X, _  
    vbOKOnly, "Перевірка даних"  
    Y = Sqr((1 + Sin(X / 2) ^ 2) / (A + B * Log(X + 3)))  
MsgBox " Y=" _  
    & Y, vbOKOnly, "Результат виконання програми"  
End Sub
```

## ДОДАТОК Б

### Приклади програм до лабораторної роботи №2.2 для виконання в середовищі Microsoft Office

#### 1-й варіант програми до лабораторної роботи

##### Sub ЛабРоб\_2\_2\_1()

' Тема : Структура "Розгалуження" та оператори умовного переходу \_  
If...Then...Else

Dim X!, R!, Y!, Z! ' оголошення змінних

X = Application.InputBox("Введіть значення для X", \_  
"Введення початкових даних")

R = Application.InputBox("Введіть значення для R", \_  
"Введення початкових даних")

MsgBox "Введено такі значення: X=" & X & "; R=" & R, vbOKOnly, \_  
"Перевірка початкових даних"

If X <= R Then

Y = (X ^ 2 - R ^ 2) / X

MsgBox "Обчислено значення: y=" & Y, vbOKOnly, "Результат"

Else

Z = Atn(X - R)

MsgBox "Обчислено значення: z=" & Z, vbOKOnly, "Результат"

End If

End Sub

#### 2-й варіант лабораторної роботи №2.2

##### Sub ЛабРоб\_2\_2\_2()

' Тема : Структура "Розгалуження" та оператори умовного \_  
переходу If...Then

Dim X!, R!, Y!, Z! ' оголошення змінних

X = Application.InputBox("Введіть значення для X", \_  
"Введення початкових даних")

R = Application.InputBox("Введіть значення для R", \_  
"Введення початкових даних")



```
MsgBox "Введено такі значення: X=" & X & "; R=" & R, vbOKOnly, _  
"Перевірка початкових даних"
```

```
If X <= R Then _
```

```
    MsgBox "Обчислено значення: Y=" & (X ^ 2 - R ^ 2) / X, vbOKOnly, _  
        "Результат"
```

```
If X > R Then _
```

```
    MsgBox "Обчислено значення: Z=" & Atn(X - R), vbOKOnly, "Результат"
```

```
End Sub
```

## ДОДАТОК В

### Приклади програм до лабораторної роботи №2.3 для виконання в середовищі Microsoft Office

#### 1-й варіант програми:

##### Sub ЛабРоб\_2\_3\_1()

' Структура розгалуження з використанням операторів умовного (If... Then... Else) та безумовного (Goto) переходу

Dim X!, R!, Y!, Z!, X1!, X2!

X1 = Application.InputBox("Введіть значення для X1", "Введення початкових даних")

X2 = Application.InputBox("Введіть значення для X2", "Введення початкових даних")

R = Application.InputBox("Введіть значення для R", "Введення початкових даних")

MsgBox "Введено такі значення: X1 = " & X1 & "; X2 = " & X2 & "; R = " & R, vbOKCancel, "Початкові дані"

X = X1

m2:

If X <= R Then

Y = (X ^ 2 - R ^ 2) / X

MsgBox "Обчислено значення: Y = " & Y & " при X = " & X & " та R = " & R, vbOKOnly, "Результат"

Else

Z = Atn(X - R)

MsgBox "Обчислено значення: Z = " & Z & " при X = " & X & " та R = " & R, vbOKOnly, "Результат"

End If

If X = X2 Then GoTo m1

X = X2

GoTo m2

m1:

MsgBox "Програма успішно завершила роботу", vbOKOnly, "Лаб. робота №2\_3\_1"

End Sub

#### 2-й варіант програми

##### Sub ЛабРоб\_2\_3\_2()

Структура розгалуження з використанням оператора вибору Select Case та оператора умовного переходу If...Then

```
Dim X!, R!, Y!, Z!, X1!, X2!, S As Integer
```

```
X1 = Application.InputBox("Введіть значення для X1", _  
    "Введення початкових даних")
```

```
X2 = Application.InputBox("Введіть значення для X2", _  
    "Введення початкових даних")
```

```
R = Application.InputBox("Введіть значення для R", _  
    "Введення початкових даних")
```

```
MsgBox "Введено такі значення: X1 =" & X1 & "; X2 =" & X2 & _  
    "; R =" & R, vbOKCancel, "Початкові дані"
```

```
X = X1
```

```
m2:
```

```
If X <= R Then S = 1 Else S = 2
```

```
Select Case S
```

```
Case 1
```

```
Y = (X ^ 2 - R ^ 2) / X
```

```
MsgBox "Обчислено значення: Y =" & Y & " при X =" & X & _  
    " та R =" & R, vbOKOnly, "Результат"
```

```
Case 2
```

```
Z = Atn(X - R)
```

```
MsgBox "Обчислено значення: Z =" & Z & " при X =" & X & _  
    " та R =" & R, vbOKOnly, "Результат"
```

```
End Select
```

```
If X = X2 Then GoTo m1
```

```
X = X2
```

```
GoTo m2
```

```
m1:
```

```
MsgBox "Програма успішно завершила роботу", vbOKOnly, _  
    "Лаб. робота №2_3_2"
```

```
End Sub
```

## ДОДАТОК Г

### Приклади програм до лабораторної роботи №2.4 для виконання в середовищі Microsoft Office

(прикладі програм для виконання у Microsoft Excel)

#### 1-й варіант програми

##### Sub Lr\_2\_4\_1()

' Тема: програми циклічних структур

' Варіант програми з використанням оператора If...Then в кінці тіла циклу

Dim X0 As Single, X1 As Single, X2 As Single, i As Integer, X As Single

Dim Y As Single, n As Integer

X0 = Application.InputBox("Введіть початок інтервалу табулювання X0", \_

" 4-та л.р. - цикл з If...Then")

n = Application.InputBox("Введіть кількість значень функції n", \_

" 4-та л.р. - цикл з If...Then ")

X2 = Application.InputBox("Введіть кінець інтервалу табулювання X2", \_

" 4-та л.р. - цикл з If...Then ")

MsgBox "X0 = " & X0 & "; n = " & n & "; X2 = " & X2, vbOKCancel, \_

"Перевірка початкових даних"

X1 = (X2 - X0) / (n - 1) ' Обчислення кроку табулювання

X = X0 ' Надання незалежній змінній початкового значення

i = 0 ' Лічильник кількості елементів таблиці

m1:

Y = Sqr(0.2 \* X ^ 4 + 4 \* Sqr(X))

i = i + 1

MsgBox "X = " & X & "; Y = " & Y, vbOKCancel, i & \_

"-й елемент таблиці"

X = X + X1

If i < n Then GoTo m1

MsgBox "Цикл успішно завершено", vbOKOnly, \_

"Інформація про завершення Л.Р. №2\_4\_1"

End Sub

#### 2-й варіант програми

##### Sub Lr\_2\_4\_2()

' Варіант програми з використанням оператора If...Then на початку тіла циклу

Dim X0 As Single, X1 As Single, X2 As Single, i As Integer, X As Single

Dim Y As Single, n As Integer

X0 = Application.InputBox("Введіть початок інтервалу табулювання X0", \_

```

    " 4-га л.р. - цикл з If...Then ")
n = InputBox("Введіть кількість обчислених значень функції n", _
    " 4-га л.р. - цикл з If...Then ")
X2 = Application.InputBox("Введіть кінець інтервалу табулювання X2", _
    " 4-га л.р. - цикл з If...Then ")
MsgBox "X0 = " & X0 & "; n = " & n & "; X2 = " & X2, vbOKCancel, _
    "Перевірка початкових даних"
X1 = (X2 - X0) / (n - 1) ' Обчислення кроку табулювання
X = X0 ' Надання незалежній змінній початкового значення
i = 1 ' Лічильник кількості елементів таблиці
m1:
If i > n Then GoTo m2
Y = Sqr(0.2 * X ^ 4 + 4 * Sqr(X))
MsgBox "X = " & X & "; Y = " & Y, vbOKCancel, i & "-й елемент таблиці"
i = i + 1
X = X + X1
GoTo m1
m2:
MsgBox "Цикл успішно завершено", vbOKOnly, _
    "Інформація про завершення Л.Р. №2_4_2"
End Sub

```

### 3-й варіант програми

#### Sub Lr\_2\_4\_3()

```

' Варіант програми з використанням оператора For...Next для
  організації циклу
Dim X0 As Single, X1 As Single, X2 As Single, i As Integer, X As Single
Dim Y As Single, n As Integer
X0 = Application.InputBox("Введіть початок інтервалу табулювання X0", _
    " 4-га л.р. - цикл з For...Next ")
n = Application.InputBox("Введіть кількість значень функції n", _
    " 4-га л.р. - цикл з For...Next ")
X2 = Application.InputBox("Введіть кінець інтервалу табулювання X2", _
    " 4-га л.р. - цикл з For...Next ")
MsgBox "X0 = " & X0 & "; n = " & n & "; X2 = " & X2, vbOKCancel, _
    "Перевірка початкових даних"
i = 0 ' Лічильник кількості елементів таблиці
For X = X0 To X2 Step (X2 - X0) / (n - 1) ' крок табулювання
    обчислюється тут
Y = Sqr(0.2 * X ^ 4 + 4 * Sqr(X))
i = i + 1

```

```
MsgBox "X = " & X & "; Y = " & Y, vbOKCancel, i & "-й елемент таб-  
лиці"
```

```
Next X
```

```
MsgBox "Цикл успішно завершено", vbOKOnly, _  
"Інформація про завершення Л.Р. №2_4_3"
```

```
End Sub
```

## ДОДАТОК Д

### Приклади програм до лабораторної роботи №2.5 для виконання в середовищі Microsoft Office

(приклади програм для виконання у Microsoft Excel)

#### 1-й варіант програми

##### Цикли Do...Loop

#### Sub Lr\_2\_5\_1()

```
' Варіант програми з оператором циклу Do...Loop та передумовою _  
    за допомогою ключового слова While  
Dim X0 As Single, X1 As Single, X2 As Single, i As Integer, X As Single  
    Dim Y As Single, n As Integer  
X0 = Application.InputBox("Введіть початок інтервалу табулювання X0", _  
    " 5-та лаб.- цикл з Do...Loop ")  
n = Application.InputBox("Введіть кількість значень функції n", _  
    " 5-та лаб.- цикл з Do...Loop ")  
X2 = Application.InputBox("Введіть кінець інтервалу табулювання X2", _  
    " 5-та лаб.- цикл з Do...Loop ")  
MsgBox "X0 = " & X0 & "; n = " & n & "; X2 = " & X2, vbOKCancel, _  
    "Перевірка початкових даних"  
X1 = (X2 - X0) / (n - 1) ' Обчислення кроку табулювання  
X = X0 ' Надання незалежній змінній початкового значення  
i = 1 ' Лічильник кількості елементів таблиці  
    Do While X <= X2 ' можна поставити умову i < n  
        Y = Sqr(0.2 * X ^ 4 + 4 * Sqr(X))  
        MsgBox "X = " & X & "; Y = " & Y, vbOKCancel, _  
            i & "-й елемент таблиці"  
        i = i + 1  
        X = X + X1  
    Loop  
MsgBox "Цикл успішно завершено", vbOKOnly, _  
    "Інформація про завершення Л.Р. №2_5_1"  
End Sub
```

#### 2-й варіант програми

#### Sub Lr\_2\_5\_2()

```
' Варіант програми з оператором циклу Do...Loop та післяумовою _  
    за допомогою ключового слова While
```

```

Dim X0 As Single, X1 As Single, X2 As Single, i As Integer, X As Single
Dim Y As Single, n As Integer
X0 = Application.InputBox("Введіть початок інтервалу табулювання X0", _
    " 5-та лаб.- цикл з Do...Loop ")
n = Application.InputBox("Введіть кількість значень функції n", _
    " 5-та лаб.- цикл з Do...Loop ")
X2 = Application.InputBox("Введіть кінець інтервалу табулювання X2", _
    " 5-та лаб.- цикл з Do...Loop ")
MsgBox "X0 = " & X0 & "; n = " & n & "; X2 = " & X2, vbOKCancel, _
    "Перевірка початкових даних"
X1 = (X2 - X0) / (n - 1) ' Обчислення кроку табулювання
X = X0 ' Надання незалежній змінній початкового значення
i = 1 ' Лічильник кількості елементів таблиці
Do
    Y = Sqr(0.2 * X ^ 4 + 4 * Sqr(X))
    MsgBox "X = " & X & "; Y = " & Y, vbOKCancel, _
        i & "-й елемент таблиці"
    i = i + 1
    X = X + X1
Loop While X <= X2 ' можна поставити умову i < n
MsgBox "Цикл успішно завершено", vbOKOnly, _
    "Інформація про завершення Л.Р. №2_5_2"
End Sub

```

### 3-й варіант програми

#### Sub Lr\_2\_5\_3()

```

' Варіант програми з оператором циклу Do...Loop та передумовою за _
    допомогою ключового слова Until
Dim X0 As Single, X1 As Single, X2 As Single, i As Integer, X As Single
Dim Y As Single, n As Integer
X0 = Application.InputBox("Введіть початок інтервалу табулювання X0", _
    " 5-та лаб.- цикл з Do...Loop ")
n = Application.InputBox("Введіть кількість значень функції n", _
    " 5-та лаб.- цикл з Do...Loop ")
X2 = Application.InputBox("Введіть кінець інтервалу табулювання X2", _
    " 5-та лаб.- цикл з Do...Loop ")
MsgBox "X0 = " & X0 & "; n = " & n & "; X2 = " & X2, vbOKCancel, _
    "Перевірка початкових даних"
X1 = (X2 - X0) / (n - 1) ' Обчислення кроку табулювання
X = X0 ' Надання незалежній змінній початкового значення
i = 1 ' Лічильник кількості елементів таблиці
Do Until X > X2 ' можна поставити умову i > n
    Y = Sqr(0.2 * X ^ 4 + 4 * Sqr(X))

```



```

    MsgBox "X = " & X & "; Y = " & Y, vbOKCancel, _
        i & "-й елемент таблиці"
    i = i + 1
    X = X + X1
Loop
MsgBox "Цикл успішно завершено", vbOKOnly, _
    "Інформація про завершення Л.Р. №2_5_3"
End Sub

```

#### 4-й варіант програми

##### Sub Lr2\_5\_4()

```

' Варіант програми з оператором циклу Do...Loop та післяумовою за _
    допомогою ключового слова Until
Dim X0 As Single, X1 As Single, X2 As Single, i As Integer, X As Single
Dim Y As Single, n As Integer
X0 = Application.InputBox("Введіть початок інтервалу табулювання X0", _
    " 5-та лаб.- цикл з Do...Loop ")
n = Application.InputBox("Введіть кількість значень функції n", _
    " 5-та лаб.- цикл з Do...Loop ")
X2 = Application.InputBox("Введіть кінець інтервалу табулювання X2", _
    " 5-та лаб.- цикл з Do...Loop ")
MsgBox "X0 - " & X0 & "; n - " & n & "; X2 - " & X2, vbOKCancel, _
    "Перевірка початкових даних"
X1 = (X2 - X0) / (n - 1) ' Обчислення кроку табулювання
X = X0 ' Надання незалежній змінній початкового значення
i = 1 ' Лічильник кількості елементів таблиці

Do
    Y = Sqr(0.2 * X ^ 4 + 4 * Sqr(X))
    MsgBox "X = " & X & "; Y = " & Y, vbOKCancel, _
        i & "-й елемент таблиці"
    i = i + 1
    X = X + X1
Loop Until X > X2 ' можна поставити умову i > n

MsgBox "Цикл успішно завершено", vbOKOnly, _
    "Інформація про завершення Л.Р. №2_5_4"

End Sub

```

## ДОДАТОК Е

### Приклади програм до лабораторної роботи №2.6 для виконання в середовищі Microsoft Office

(прикладі програм для виконання у Microsoft Excel)

#### 1-й варіант програми

##### Private Sub Lr2\_6\_1()

' Лабораторна робота № 6 за темою "Функції та підпрограми (процедури)"

' Варіант з текстом підпрограми, розташованим у тексті основної програми

Dim A As Single, B As Single, N As Single, M As Single, T As Single, \_

X As Single

Dim D As Single, T1 As Single, T2 As Single, T3 As Single, T4 As Single

A = Application.InputBox("Введіть значення А", \_

" 6-та лабораторна - підпрограми та функції")

B = Application.InputBox("Введіть значення В", \_

" 6-та лабораторна - підпрограми та функції")

N = Application.InputBox("Введіть значення N", \_

" 6-та лабораторна - підпрограми та функції")

M = Application.InputBox("Введіть значення М", \_

" 6-та лабораторна - підпрограми та функції")

MsgBox "A = " & A & "; B = " & B & "; N = " & N & "; M = " \_

& M, vbOKCancel, \_

"Перевірка початкових даних"

X = A \* N + B

GoSub m220

T1 = T

X = A ^ 2 + B ^ 2

GoSub m220

T2 = T

X = A + M \* B

T3 = T

X = A ^ 2 - B ^ 2

GoSub m220

T4 = T

D = T1 / T2 ^ 2 - T3 / T4

MsgBox "D = " & D, vbOKCancel, "Результат 6 -ї роботи"

GoTo En

' \*\*\*\*\* ПІДПРОГРАМА (початок) \*\*\*\*\*

m220:

$$T = (\text{Exp}(2 * X) - 1) / (\text{Exp}(2 * X) + 1)$$

Return

'\*\*\*\*\* ПІДПРОГРАМА (кінець) \*\*\*\*\*'

En:

**End Sub**

## ДОДАТОК Ж

### Приклади для виконання програм до лабораторної роботи №2.7 в Microsoft Office

(зокрема для виконання в Microsoft Excel)

#### Private Sub Lr\_2\_7\_1()

' Лабораторна робота № 7 за темою "Функції та підпрограми (процедури)"

' Варіант з однотипними обчисленнями у процедурі типу Function

Dim A As Single, B As Single, N As Single, M As Single, X As Single

Dim D As Single, T1 As Single, T2 As Single, T3 As Single, T4 As Single

' Зверніть увагу на те, що ідентифікатор (ім'я) функції T в \_  
операторі Dim визначати не потрібно. Воно визначається \_  
наявністю самої функції у тексті модуля та зверненням \_  
до цієї функції у тексті даної процедури (Lr\_7\_1())

A = Application.InputBox("Введіть значення A", \_  
" 7-ма лабораторна - підпрограми та функції")

B = Application.InputBox("Введіть значення B", \_  
" 7-ма лабораторна - підпрограми та функції")

N = Application.InputBox("Введіть значення N", \_  
" 7-ма лабораторна - підпрограми та функції")

M = Application.InputBox("Введіть значення M", \_  
" 7-ма лабораторна - підпрограми та функції")

MsgBox "A =" & A & "; B =" & B & "; N =" & N & "; M ="

& M, vbOKCancel, "Перевірка початкових даних"

D = T(A \* N + B) / T(A ^ 2 + B ^ 2) ^ 2 - T(A + M \* B) / T(A ^ 2 - B ^ 2)

MsgBox "D =" & D, vbOKCancel, "Результат роботи"

' Виведення результату програми

End Sub

Function T(X) As Single ' функція для обчислення T

T = (Exp(2 \* X) - 1) / (Exp(2 \* X) + 1)

End Function

#### 2-й варіант програми

#### Private Sub Lr\_2\_7\_2()

' Лабораторна робота № 7 за темою "Функції та підпрограми (процедури)"

' Варіант з однотипними обчисленнями у процедурі типу Function

та обчисленням результату програми в операторі виведення (MsgBox або  
Print)

Dim A As Single, B As Single, N As Single, M As Single, X As Single

Dim D As Single, T1 As Single, T2 As Single, T3 As Single, T4 As Single

A = Application.InputBox("Введіть значення А", \_  
" 7-ма лабораторна - підпрограми та функції")

B = Application.InputBox("Введіть значення В", \_  
" 7-ма лабораторна - підпрограми та функції")

N = Application.InputBox("Введіть значення N", \_  
" 7-ма лабораторна - підпрограми та функції")

M = Application.InputBox("Введіть значення М", \_  
" 7-ма лабораторна - підпрограми та функції")

MsgBox "A = " & A & "; B = " & B & "; N = " & N & "; M = "

& M, vbOKCancel, "Перевірка початкових даних"

MsgBox "D = "&T(A\*N + B)/T(A ^ 2 + B ^ 2) ^ 2 - T(A + M \* B)/T(A ^ 2 - B ^ 2), \_  
vbOKCancel, "Результат 7 -ї роботи"

**End Sub**

Function T(X) As Single

T = (Exp(2 \* X) - 1) / (Exp(2 \* X) + 1)

End Function

## ДОДАТОК И

### Приклади для виконання програм до лабораторної роботи №2.8 в Microsoft Office

(зокрема для виконання в Microsoft Excel)

#### Private Sub Lr\_2\_8\_1()

' Лабораторна робота №2.8 за темою "Функції та підпрограми (процедури)"

' Варіант з текстом підпрограми, розташованим у \_

процедурі Tpr(X As Single, T As Single)\_

та звернення до підпрограми-процедури із зазначенням її імені з \_  
формальними параметрами без дужок (1-ший варіант програми)

Dim A As Single, B As Single, N As Single, M As Single, T As Single, \_  
X As Single

Dim D As Single, T1 As Single, T2 As Single, T3 As Single, T4 As Single, \_  
P As Single

A = Application.InputBox("Введіть значення A", \_  
"Введення даних до 1-шого вар. роботи")

B = Application.InputBox("Введіть значення B", \_  
"Введення даних до 1-шого вар. роботи")

N = Application.InputBox("Введіть значення значення N", \_  
"Введення даних до 1-шого вар. роботи")

M = Application.InputBox("Введіть значення M", \_  
"Введення даних до 1-шого вар. роботи")

MsgBox "A =" & A & "; B =" & B & "; N =" & N & "; M =" & \_  
M, vbOKCancel, "Перевірка початкових даних"

Tpr A \* N + B, P

T1 = P

Tpr A ^ 2 + B ^ 2, P

T2 = P

Tpr A + M \* B, P

T3 = P

Tpr A ^ 2 - B ^ 2, P

T4 = P

D = T1 / T2 ^ 2 - T3 / T4

MsgBox "D =" & D, vbOKCancel, "Результат 1-шого варіанта роботи"

**End Sub**

Private Sub Tpr(X As Single, T As Single)

T = (Exp(2 \* X) - 1) / (Exp(2 \* X) + 1)

End Sub

## Другий варіант програми

### Private Sub Lr\_2\_8\_2()

' Лабораторна робота № 8 за темою "підпрограми (процедури)"

' Варіант з текстом підпрограми, розташованим у процедурі\_

  Tpr(X As Single, T As Single)

' та звернення до підпрограми-процеури із зазначанням її імені з \_  
  формальними параметрами у дужках

Dim A As Single, B As Single, N As Single, M As Single, T As Single,\_  
  X As Single

Dim D As Single, T1 As Single, T2 As Single, T3 As Single, T4 As Single,\_  
  P As Single

  A = Application.InputBox("Введіть значення A", \_  
    "Введення даних до 2-го вар. роботи")

  B = Application.InputBox("Введіть значення B", \_  
    "Введення даних до 2-го вар. роботи")

  N = Application.InputBox("Введіть значення значення N", \_  
    "Введення даних до 2-го вар. роботи")

  M = Application.InputBox("Введіть значення M", \_  
    "Введення даних до 2-го вар. роботи")

  MsgBox "A = " & A & "; B = " & B & "; N = " & N & "; M = " &  
    M. vbOKCancel. "Перевірка початкових даних"

  Call Tpr(A \* N + B, P)

  T1 = P

  Call Tpr(A ^ 2 + B ^ 2, P)

  T2 = P

  Call Tpr(A : M \* B, P)

  T3 = P

  Call Tpr(A ^ 2 - B ^ 2, P)

  T4 = P

  D = T1 / T2 ^ 2 - T3 / T4

  MsgBox "D = " & D, vbOKCancel, "Результат 2-го варіанта роботи"

End Sub

Private Sub Tpr(X As Single, T As Single)

  T = (Exp(2 \* X) - 1) / (Exp(2 \* X) + 1)

End Sub

### Третій варіант програми

#### Private Sub Lr\_3\_8\_3()

' Лабораторна робота № 2.8 за темою "підпрограми (процедури)"

' Варіант з розрахунками та виведенням результату у підпрограмі-процедурі

```
    з використанням підпрограми-функції, _  
    та введенням даних для розрахунку в основній програмі (процедурі)  
Dim A As Single, B As Single, N As Single, M As Single, T As Single, _  
    X As Single  
Dim D As Single, T1 As Single, T2 As Single, T3 As Single, T4 As Single  
    A = Application.InputBox("Введіть значення A", _  
        "Введення даних до 3-шого вар. роботи")  
    B = Application.InputBox("Введіть значення B", _  
        "Введення даних до 3-шого вар. роботи")  
    N = Application.InputBox("Введіть значення значення N", _  
        "Введення даних до 3-шого вар. роботи")  
    M = Application.InputBox("Введіть значення M", _  
        "Введення даних до 3-шого вар. роботи")  
MsgBox "A = " & A & ": B = " & B & ": N = " & N & ": M = " & _  
    M, vbOKCancel, _  
    "Перевірка початкових даних"  
  
Call Tpr_3(A, B, N, M)  
MsgBox "Програма 3-го варіанта успішно виконана", , _  
    "Підсумковий результат"  
End Sub
```

```
Private Sub Tpr_3(X As Single, Y As Single, Z As Single, K As Single)  
Dim D As Single  
D = T(X * Z + Y) / T(X ^ 2 + Y ^ 2) - T(X + K * Y) / T(X ^ 2 - Y ^ 2)  
MsgBox "D = " & D, vbOKCancel, "Результат 3-го варіанта роботи"  
End Sub
```

```
Function T(X) As Single  
    T = (Exp(2 * X) - 1) / (Exp(2 * X) + 1)  
End Function
```



*Навчальне видання*

**Пішеніна Надія Володимирівна  
Пішенін Володимир Олексійович**

**ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА  
ПРОГРАМУВАННЯ**

**Частина 1**

**Лабораторний практикум**

Оригінал-макет підготовлено В.О. Пішеніним

Редактор В.О. Дружиніна  
Коректор З.В. Поліщук

Навчально-методичний відділ ВНТУ  
Свідоцтво Держкомінформу України  
серія ДК №746 від 25.12.2001  
21021, м. Вінниця, Хмельницьке шосе, 95, ВНТУ

Підписано до друку *30.12.03р*  
Формат 29,7×42<sup>1</sup>/<sub>4</sub>  
Друк різнографічний  
Тираж **80** прим.  
Зам. № *2004-2*

Гарнітура Times New Roman  
Папір офсетний  
Ум. друк. арк. *4,85*

Віддруковано в комп'ютерному інформаційно-видавничому центрі  
Вінницького національного технічного університету  
Свідоцтво Держкомінформу України  
серія ДК №746 від 25.12.2001  
21021, м. Вінниця, Хмельницьке шосе, 95, ВНТУ