

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
АВТОМОБІЛЬНО-ДОРОЖНИЙ ІНСТИТУТ

Кафедра «Прикладна математика та інформатика»

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ПРАКТИКУМУ З ДИСЦИПЛІНИ
«ІНФОРМАТИКА». ЧАСТИНА 1
(ДЛЯ СТУДЕНТІВ НАПРЯМУ ПІДГОТОВКИ
6.060101 «БУДІВНИЦТВО»
УСІХ ФОРМ НАВЧАННЯ)**

22/25-2014

Горлівка – 2014

Методичні вказівки до практикуму з дисципліни «Інформатика». Частина 1 (для студентів напряму підготовки 6.060101 «Будівництво» усіх форм навчання) [Електронний ресурс] / укладачі В. Г. Хребет, Р. С. Кравченко. – Електрон. дані. – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2014. – 1 електрон. опт. диск (CD-R); 12 см. – Систем. вимоги: Pentium; 32 MBRAM; WINDOWS XP/Vista/7; MS Word 2000-2010. – Назва з титул. екрану.

Методичні вказівки містять відомості, що необхідні під час проходження практикуму з дисципліни «Інформатика», які допомагають студентам: у роботі з текстовими документами, у розвитку навичок формалізації задач, розробці структури електронної таблиці, відображенні результатів розрахунків, тестуванні електронних таблиць, у розробці програм та реалізації основних типів алгоритмів, та ознайомлення з особливостями створення графічних додатків.

Укладачі:

Хребет В. Г., к.ф.-м.н., доц.
Кравченко Р. С.

Відповідальний за випуск:

В. Г. Хребет, к.ф.-м.н., доц.

Рецензент:

В. П. Вовк, д.т.н., проф.,
зав. каф. «Вища математика»

© Державний вищий навчальний заклад
«Донецький національний технічний університет»
Автомобільно-дорожній інститут, 2014

ЗМІСТ

ВСТУП	4
1 ЛАБОРАТОРНА РОБОТА №1	5
1.1 Теоретичні відомості.....	5
1.2 Практичне завдання.....	7
1.3 Практичне завдання.....	7
2 ЛАБОРАТОРНА РОБОТА №2	13
2.1 Теоретичні відомості.....	13
2.2 Практичне завдання.....	33
3 ЛАБОРАТОРНА РОБОТА №3	35
3.1 Короткі теоретичні відомості.....	35
3.2 Практичне завдання.....	37
4 ЛАБОРАТОРНА РОБОТА №4	39
4.1 Теоретичні відомості.....	39
4.2 Практичне завдання.....	51
5 ЛАБОРАТОРНА РОБОТА №5	53
5.1 Теоретичні відомості.....	53
3.5 Розв'язок задачі «Алгоритм розгалуження 4 гілки» (Select Case)....	60
5.2 Практичне завдання.....	63
6 ЛАБОРАТОРНА РОБОТА №6	65
6.1 Теоретичні відомості.....	65
6.2 Практичне завдання.....	74
ДОДАТОК А.....	77
А.1 Алгоритм. Види алгоритмів. Блок-схеми	77
ДОДАТОК Б	79
Б.1 Клас Windows Forms. Відомості про об'єктно-орієнтоване програмування	79
Б.2 Загальні властивості для основних елементів управління.....	81
ДОДАТОК В СТРУКТУРНІ ЕЛЕМЕНТИ КУРСОВОЇ РОБОТИ	88
ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ.....	91

ВСТУП

Практикум є безпосереднім продовженням теоретичного курсу дисципліни «Інформатика» у поглибленому вигляді:

- з точки зору систематизації студентами знань, професійних умінь і навичок, що одержані в процесі вивчення відповідних дисциплін;
- з точки зору прикладного (техніко-економічного) аспекту формалізації та розв’язку деяких практичних задач (моделей);
- з точки зору алгоритмізації розв’язку задач (моделей), та автоматизації розрахунків за допомогою мови програмування Visual Basic .Net.

Завданням проведення практикуму є:

- оволодіння навичками роботи з сучасним апаратним та програмним забезпеченням;
- набуття досвіду роботи з найпоширенішими пакетами прикладних програм: MS Excel та MS Word, що входить у пакет Microsoft Office;
- формування та розвиток алгоритмічного мислення у студентів;
- використання програмних засобів для формалізації техніко-економічних задач у вигляді математичних моделей, та їх автоматизація;
- вивчення основних етапів розробки програмних продуктів.

Мета методичних вказівок полягає в наданні допомоги студентам під час проходження практикуму, а саме: у розвитку навичок формалізації прикладних задач, розробці структури електронної таблиці, відображення результатів розрахунків, тестування електронних таблиць.

1 ЛАБОРАТОРНА РОБОТА №1

Тема: робота з ОС Windows.

Мета: навчитися основних прийомів роботи з та її додатками.

1.1 Теоретичні відомості

Windows 7 – операційна система Windows, котра вийшла після Windows Vista у 2009 році.

У Windows 7 з'явилися нові можливості та функції інтерфейсу.

1. Shake. У інтерфейс Windows Aero додана нова функція Aero Shake, що дозволяє скрутити всі неактивні застосування рухом миші. Для її активації досить захопити заголовок вікна і трохи «потрясти» вліво-управо.

2. Peek. Функція Aero Peek дозволяє відображувати зменшені копії вікон при наведенні миші на значок панелі завдань, перемикатися між вікнами застосунку простим кліком по значку, перетягувати і фіксувати на панелі завдань різні вікна і програми, переглядати робочий стіл одним наведенням в спеціальну область екрану і багато що інше.

3. Snap. Аналогічно функції Shake функція Aero Snap дозволяє рухом миші розвертати вікно підлоги-екрану, весь екран або лише по вертикальній осі.

Стилі оформлення

Windows 7 підтримує три варіанти оформлення, призначеного для користувацького інтерфейсу:

Windows Aero — це оригінальний стиль оформлення з прозорими багатоколірними рамками вікон, вживаний за умовчанням для комп'ютерів з більш ніж 1 Гб ОЗУ. Доступний на Windows 7 Home Premium, Windows 7 Professional і старших редакціях.

Windows 7 - спрощений стиль — Windows Aero з деякими відключеними можливостями (наприклад, прозорість вікон і Windows Flip 3d). Вимоги до системи — такі ж, як і в Windows Aero. Доступний на всіх редакціях Windows 7, і є основним в редакції Windows 7 Starter. Цей стиль так само застосовується при запуску програм в режимі сумісності.

Класична — мінімальні вимоги до системи, оформлення вікон в стилі «класичної» теми Windows XP. Доступні різні колірні схеми, зокрема, подібні до схем Windows 98; користувач може створювати свої колірні схеми.

Мультимедіа

У Windows 7 використовується DirectX 11 і Windows Media Player 12. Останній — отримав новий інтерфейс і став воістину всеїдним, на відміну від попередника, якому була потрібна велика кількість кодеків для відтворення. Проте, він не може відтворювати ліцензійні диски Blu-ray з відео, але є можливість зчитувати і записувати дані.

Система

Microsoft планує встановлювати Windows 7, зокрема, і на нетбуки, що відрізняються невисокою потужністю. У зв'язку з цим вимоги Windows 7 до апаратного забезпечення – за що так лають Windows Vista – знижені. З другого боку, нова операційна система зможе підтримувати до 256 процесорів.

Представники Microsoft відзначили, що в новій Windows вдосконалено систему безпеки. Зокрема, покращено роботу User Account Control (UAC). У новій версії користувач зможе виставити рівень доступу для того, щоб вікно UAC не виникало кожного разу при спробі змінити системні параметри.

Платформа розбита на три частини: ядро Windows 7 і основні файли, пакет Windows Live + Windows Live Essentials і пакет Windows Live Services. Виробники ПК дістануть можливість встановлювати пакет Windows Live Essentials, який забезпечить користувачів всім необхідним для початку роботи. До складу пакету увійдуть додатки Windows Live Mail, Messenger, Movie Maker, Photo Gallery, Writer і деякі інші. Пакет Services, у свою чергу, призначений для роботи з Hotmail і іншими службами в інтернеті.

Обіцяно, що особлива увага була приділена сумісності платформ. Windows 7 розроблена на основі Vista, і всі додатки, випущені для Vista, працюватимуть і на Windows 7. Кожного разу при переході на нову ОС проблема сумісності викликає море критики. Користувачі виявили, що додатки, які вони запускали на Windows XP, відмовилися працювати на Vista.

Впроваджено нову модель WDDM 1.1, що входить до DirectX 11 та яка має скоротити обсяги споживання відеопам'яті системою.

У Windows 7 також вбудовано Windows Defender та Брандмауер Windows.

1.2 Практичне завдання

Виконати запропоновані вправи, згідно варіанту. Представити опис (з ілюстраціями) процесу та результатів їх виконання у вигляді звіту.

Знайти відповіді на теоретичні запитання (табл. 1.1). Результат оформити у вигляді коротких теоретичних відомостей до даної роботи.

1.3 Практичне завдання

Варіант 1

1. Відкрити вікно-папку Мій комп'ютер подвійним клацанням лівої кнопки миші.
2. Створити нову папку на робочому диску (у якому дозволено створювати та зберігати папки і файли). Для цього в папці Мій комп'ютер вибрати значок папки диска й розкрити його вікно подвійним клацанням лівої кнопки миші. Виконати команди Файл > Створити > Папка. Привласнити новій папці імені, яке відповідає вашому прізвищу, і нажати клавішу <Enter>. Подвійним натисканням лівої кнопки миші відкрити вікно нової папки.
3. У вікні Мій комп'ютер активізувати вікно диска: подвійним клацанням лівої кнопки миші відкрити в ньому вікно якої-небудь папки та скопіювати звідти у вікно папки Прізвище методом перетаскування два будь-які файли, а за допомогою команд Виправлення > Копіювати та Виправлення > Вставити — два файли з назвою малюнків. Пояснити відмінності в результатах копіювання.
4. Вилучити файли з назвою рисунків зі знову створеної папки.
5. Відкрити папку Кошик, знайти вилучені файли та відновити їх.
6. Закрити всі вікна, крім вікна Мій комп'ютер.
7. Створити ярлик для папки Прізвище та помістити його на Робочий стіл.
8. Змінити для створеного ярлика значок.
9. Вилучити ярлик різними способами.

Варіант 2

1. Відкрити вікно Провідника будь-яким способом.
2. У лівій області активізувати робочий диск.
3. Створити в ньому нову папку з іменем, яке відповідає вашому прізвищу (папка Прізвище).
4. Активізувати нову папку й у ній створити папку з назвою групи (папка Група). Активізувати останню.

5. У лівій частині Провідника перейти в будь-яку папку робочого диска, виділити кілька файлів, які йдуть не підряд і скопіювати їх в останню створену папку (Група), використовуючи контекстне меню.
6. Показати файли в активній папці у вигляді таблиці.
7. Побудувати файли в порядку дати зміни, потім по імені файлу.
8. Змінити вид подання файлів у вигляді списку із дрібними, потім з більшими значками.
9. Двічі клацнути по файлу зі значком додатка Word. Пояснити результат.
10. Повернутися в Провідник, закривши завантажений додаток.
11. Вікно Провідника показати у відновленому стані (вікно повинне займати не весь екран).

Варіант 3.

1. У вікні Мій комп'ютер відкрити папку робочого диска, потім папку з назвою свого прізвища.
2. Згорнути всі відкриті вікна, крім вікна Провідник і вікна папки Прізвище.
3. Розташувати поруч на Робочому столі вікно папки Прізвище й вікно Провідника.
4. Активізувати вікно Провідника, а в ньому — папку Група.
5. Виділити всі файли й методом перетаскування перенести їх у вікно папки Прізвище.
6. У вікні Провідника переконатися, що папка Група порожня, потім вилучити її.
7. Вилучити папку Прізвище разом з файлами з диска. Перевірити видалення у вікні папки робочого диска.
8. Закрити всі вікна на Робочому столі.

Варіант 4

1. Відкрити папку Панель керування будь-яким способом.
2. Представити значки в папці у вигляді Таблиці, Списку, дрібних, великих значків.
3. У вікні Властивості: Миша на вкладці Кнопки миші встановити зручну швидкість подвійного натискання й перевірити її в Області перевірки.
4. Відкрити вікно папки Екран. Використовуючи розділи Фон, Заставка, Оформлення. Змінити: фон або рисунок Робочого стола, колір елементів екрана, підібрати та перевірити заставку екрана у вигляді рядка тексту, що біжить, встановивши інтервал — 1 хв.
5. Установити властивості Панелі задач так, щоб вона зникла з екрану.

6. Переконайтеся, що панель з'являється на екрані при переміщенні курсору миші на Панель задач і зникає при переміщенні курсору за її границі.
7. Установити властивості Панелі задач так, щоб вона завжди була видна на екрані.
8. Установити властивості Панелі задач так, щоб відкриті вікна перекривали Панель задач.

Варіант 5

1. Відкрити додаток Paint.
2. Використовуючи Панель інструментів, намалювати елементарний рисунок. Унизу рисунку зробити підпис: ким і коли зроблений рисунок, установивши шрифт із кирилицею, розмір 16 пт, жирний курсив.
3. Виділити фрагмент рисунку. Утримуючи клавішу Ctrl, скопіювати його в інше місце на екрані.
4. Виділити інший фрагмент і командою Виправлення > Копіювати у файл на робочий диск.
5. Вставити в поточний рисунок збережений фрагмент командою Виправлення > Вставити з файлу.
6. Використовуючи меню Рисунок, нахилити фрагмент на 40% і розтягти його на 200%.
7. Зберегти рисунок на робочий диск і закрити додаток.

Варіант 6

1. Відкрити додаток Wordpad.
2. Увести будь-який текст, установивши відступ першого рядка 1,5 одиниць, вирівнювання по ширині.
3. Скопіювати абзац у кінець тексту.
4. Установити для першого абзацу розмір шрифту 14 пт, для другого – 12 пт і шрифт курсив.
5. Для останнього абзацу задати відступ ліворуч 5 одиниць.
6. Відкрити додаток Paint, і намалювати в ньому будь-який рисунок.
7. Виділити фрагмент рисунку та скопіювати його в документ Wordpad на порожній рядок перед другим абзацом. Подвійним клацанням лівої кнопки миші по рисунку перейти в додаток Paint, внести зміни у фрагмент, вернутися в Wordpad, зберегти зміни.

Варіант 7.

1. Відкрийте папку Кошик.
2. Перемістити папку Кошик так, щоб вона перекривала Панель задач.

3. Установите властивості Панелі задач так, щоб вона завжди перекривала усі відкриті вікна.
4. Закрийте папку Кошик.
5. Перемістити Панель задач до верхньої границі екрана.
6. Відобразити на Панелі задач усі панелі інструментів.
7. Закрийте всі Панелі інструментів, крім панелі Робочий стіл одним зі способів.
8. Перемістити панель інструментів Робочий стіл з Панелі задач на екран робочого стола.

Варіант 8.

1. Перемістити панель інструментів Робочий стіл з Панелі задач на екран робочого стола.
2. Перемістити панель інструментів Робочий стіл до нижнього краю робочого стола.
3. Відобразити значки на панелі Робочий стіл великими значками та забрати підписи під значками.
4. Створіть на Панелі задач власну панель інструментів, яка відображає диск E.
5. Закрийте панель інструментів Робочий стіл.
6. Відкрийте папку Кошик.
7. Змініте розміри вікна Кошик: по горизонталі, по вертикалі, у двох напрямках.

Варіант 9.

1. Установите для вікна Мій комп'ютер режим виведення великих значків.
2. Сховайте, а потім відобразите Панель інструментів Звичайні кнопки для відкритого вікна Мій комп'ютер.
3. Активізуйте по черзі вікна Мій комп'ютер і Кошик різними способами.
4. Розташуєте вікна Каскадом і Мозаїкою.
5. Згорніть вікно Кошик одним зі способів.
6. Активізуйте папку Кошик з Панелі задач.
7. Розгорніть вікно Мій комп'ютер

Варіант 10.

1. Відкрийте папку Кошик.
2. Перемістити папку Кошик так, щоб вона перекривала Панель задач.

3. Установите властивості Панелі задач так, щоб вона завжди виводилася понад усі відкриті вікна.
4. Закрийте папку Кошик.
5. Створіть на Панелі задач власну панель інструментів, яка відображає диск Е.

Варіант 11.

1. Відкрити папку Панель керування будь-яким способом.
2. Представити значки в папці у вигляді Таблиці, Списку, дрібних, більших значків, використовуючи меню Вид.
3. У вікні Властивості: Миша на вкладці Кнопки миші встановити зручну швидкість подвійного натискання й перевірити її в Області перевірки.
4. Відкрити вікно папки Екран. Використовуючи вкладки Тло, Заставка, Оформлення. Змінити: тло або рисунок Робочого стола, колір елементів екрана, підібрати та перевірити заставку екрана у вигляді рядка тексту, що біжить, установивши інтервал – 1 хв.
5. Установити властивості Панелі задач так, щоб вона автоматично вбиралася з екрана.
6. Переконаєтесь, що панель з'являється на екрані при переміщенні курсору миші на Панель задач і зникає при переміщенні курсору за її границі.

Варіант 12

1. Перемістити панель інструментів Робочий стіл з Панелі задач на екран робочого стола.
2. Перемістити панель інструментів Робочий стіл до нижнього краю робочого стола.
3. Відобразити значки на панелі Робочий стіл великими значками та забрати підписи під значками.
4. Створіть на Панелі задач власну панель інструментів, яка відображає диск Е.
5. Закрийте панель інструментів Робочий стіл.
6. Відкрийте папку Кошик.
7. Змініте розміри вікна Кошик: по горизонталі, по вертикалі, у двох напрямках.

Варіант 13

1. Подивитися та запишіть інформацію про систему. (Мій комп'ютер → Властивості).
2. Подивитися обладнання, представлені в системі.
3. Поміняйте фон Робочого стола.
4. Поміняйте заставку. Установите інтервал запуску заставки рівним 10 хвилинам.

5. Змініте колір робочого поля вікна.
6. Установите відображення великих значків на Робочому столі.
7. Змініте колірну палітру та параметри екрану.
8. Відновите вихідні параметри екрану.

Варіант 14

1. Створіть ярлик на папку Мої документи в пункті меню Програми й меню Пуск.
2. Упорядкуєте елементи пункту меню Програми та меню Пуск по імені.
3. Відключите відображення годинника на Панелі задач.
4. Очистить пункт меню Документи та меню Пуск від документів.
5. Знайдіть усі файли з расширением.bmp. (*.bmp).

Варіант 15

1. Знайдіть усі файли, що мають в імені послідовність «док», а в розширенні букву «d».
2. Призначте англійську мову такою, яка використовується за замовчуванням.
3. Призначте для перемикавання розкладок клавіатури комбінацію клавіш Ctrl+Shift.
4. Установите відображення шлейфа миші.

Таблиця 1.1 – Теоретичні питання

Варіант	Теоретичне питання
1-15	Призначення й основні можливості Windows
1	Різноманітність вікон Windows і робота з ними
2	Робочий стіл
3	Робота з папками
4	Панель задач
5	Операції над об'єктами
6	Головне й контекстне меню
7	Папка Мій комп'ютер
8	Мережне оточення
9	Робота з ярликами
10	Робота з об'єктами за допомогою програми Провідник
11	Групові операції над об'єктами
12	Панель керування
13	Стандартні додатки: Графічний редактор Paint
14	Стандартні додатки: Текстовий редактор Wordpad
15	Настроювання екрана Windows

2 ЛАБОРАТОРНА РОБОТА №2

Тема: робота зі стилями в MS Word. редагування автоматичного змісту

Мета: створити шаблон курсової роботи з використанням стилів

2.1 Теоретичні відомості

Створення та редагування стилів

У Microsoft Word 2007 і 2010, на відміну від попередніх версій Word (до 2003 включно) стилі винесені на головну вкладку (рис. 2.1).

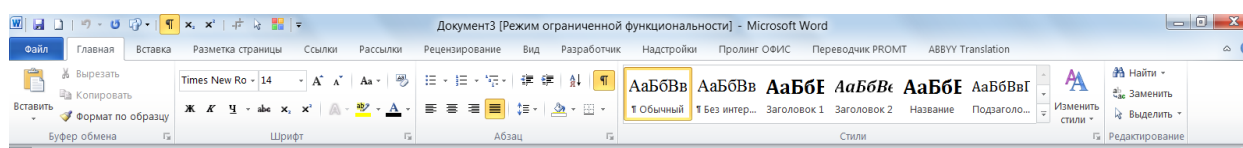


Рисунок 2.1 – Стилi на головній вкладці

Уявіть, що ви працюєте над досить об'ємним матеріалом на десять або більше сторінок. Ви оформили текст і показали його замовнику (начальнику, керівнику, викладачу), але він залишився незадоволений обраним шрифтом, кольоровою гамою, вважає, що міжрядкові інтервали занадто вузькі. Розв'язати задачу можливо лише зміною параметрів оформлення тексту, переглянувши весь документ. У випадку ж використання для оформлення стилів, досить поміняти кілька параметрів одного заголовка, щоб усі вони змінилися відповідно до нових вимог.

Стилi зручні при оформленні великої кількості однотипних документів. Їх можна використовувати в організації для створення «корпоративного стандарту». Крім того, за допомогою стилів задається структура документа, що полегшує навігацію й побудову змісту.

Розмітка стандартними стилями

Для застосування стандартних наборів стилів, достатньо виділити фрагмент тексту й вибрати необхідний стиль на панелі стилів (рис. 2.2).

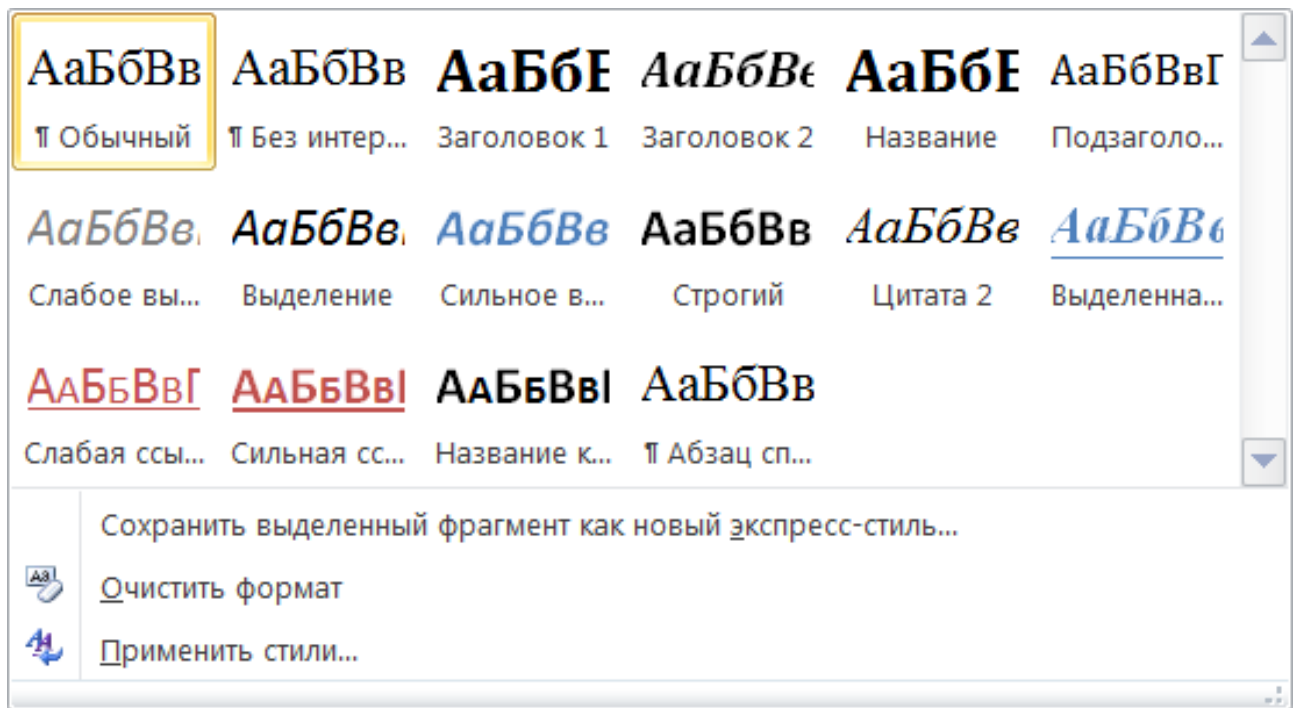


Рисунок 2.2 – Набір стилів

Тут є кілька рівнів заголовків, цитати, абзаци, назви різних об'єктів і т. д. Стиль «Обычный» використовується для тексту за замовчуванням. Спеціально його призначати не потрібно, але ви можете застосувати його до тексту, щоб вилучити задане оформлення. Альтернативний варіант повернення до звичайного стилю – комбінація клавіш «Ctrl+Shift+N». Так ви можете швидко очистити весь документ від елементів структури й оформлення. Натиснувши «Ctrl+Пробіл», ви повернете документу стандартний набір стилів, вилучивши додане вручну форматування.

Глави в тексті необхідно визначити як заголовки першого рівня, а статті – другого. Деякі абзаци можна оформити маркірованим списком, а терміни виділити стилем «Строгий» або просто напівжирним шрифтом.

Після розмітки структури документ виглядає вже набагато краще, ніж суцільний текст, але того ж ефекту ви могли добитися й задаючи конкретні параметри оформлення. От тільки витратили б більше часу.

Тепер ви можете подивитися, як легко повністю поміняти оформлення документа, змінивши лише набір стилів (рис. 2.3). Зміни відображаються відразу, як тільки ви наводите курсор на новий пункт меню.

Аналогічно ви можете змінити колір, шрифт й інтервали між абзацами для поточного набору стилів.

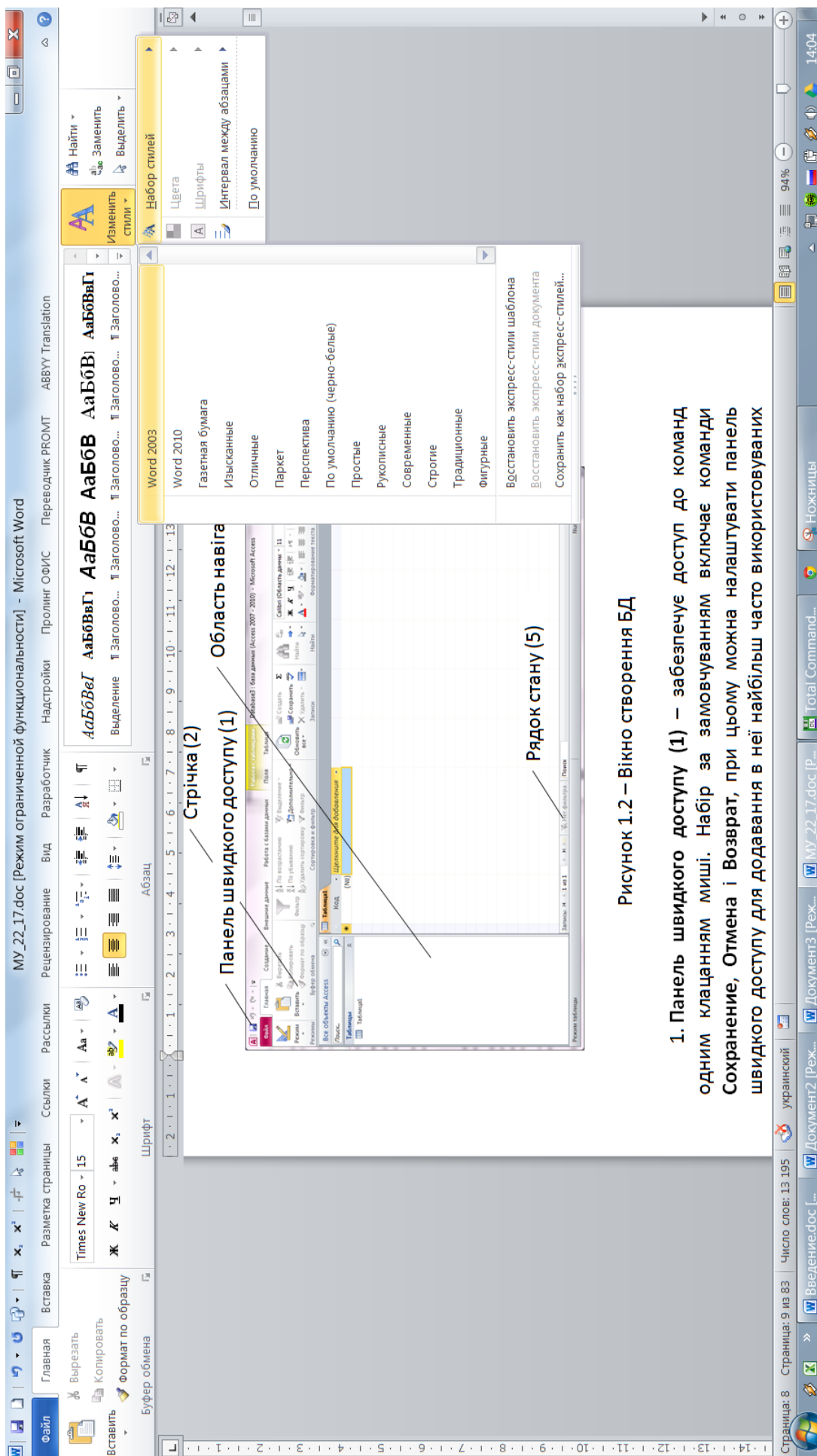


Рисунок 1.2 – Вікно створення БД

1. Панель швидкого доступу (1) – забезпечує доступ до команд одним клацанням миші. Набір за замовчуванням включає команди **Сохранение, Отмена і Возврат**, при цьому можна налаштувати панель швидкого доступу для додавання в неї найбільш часто використовуваних

Рисунок 1.3 – Вибірємо новий набір стилів

Створення користувацького набору стилів

Якщо вас не влаштовує жоден зі стандартних наборів, ви можете створити користувацький стиль із необхідними параметрами редагування. Для цього натисніть кнопку відкриття вікна стилів (рис. 2.4), щоб отримати доступ до налаштувань. У вікні стилів ви можете вибирати між відображенням назв і назв із оформленням, встановлюючи й знімаючи прапорець «Предварительный просмотр».

Створимо стиль «Название», зробимо його менш громіздким. Цього можна добитися двома основними способами.

Спосіб 1. Відформатуйте фрагмент із назвою згідно зі своїми вимогами, використовуючи налаштування шрифта й абзаца вкладки «Главная», після чого в меню стилю «Название» (рис. 2.5) виберіть пункт «Обновить Название в соответствии с выделенным фрагментом». Стиль буде оновлений і всі інші назви, якщо вони є у вашому документі, також змінять свій зовнішній вигляд.

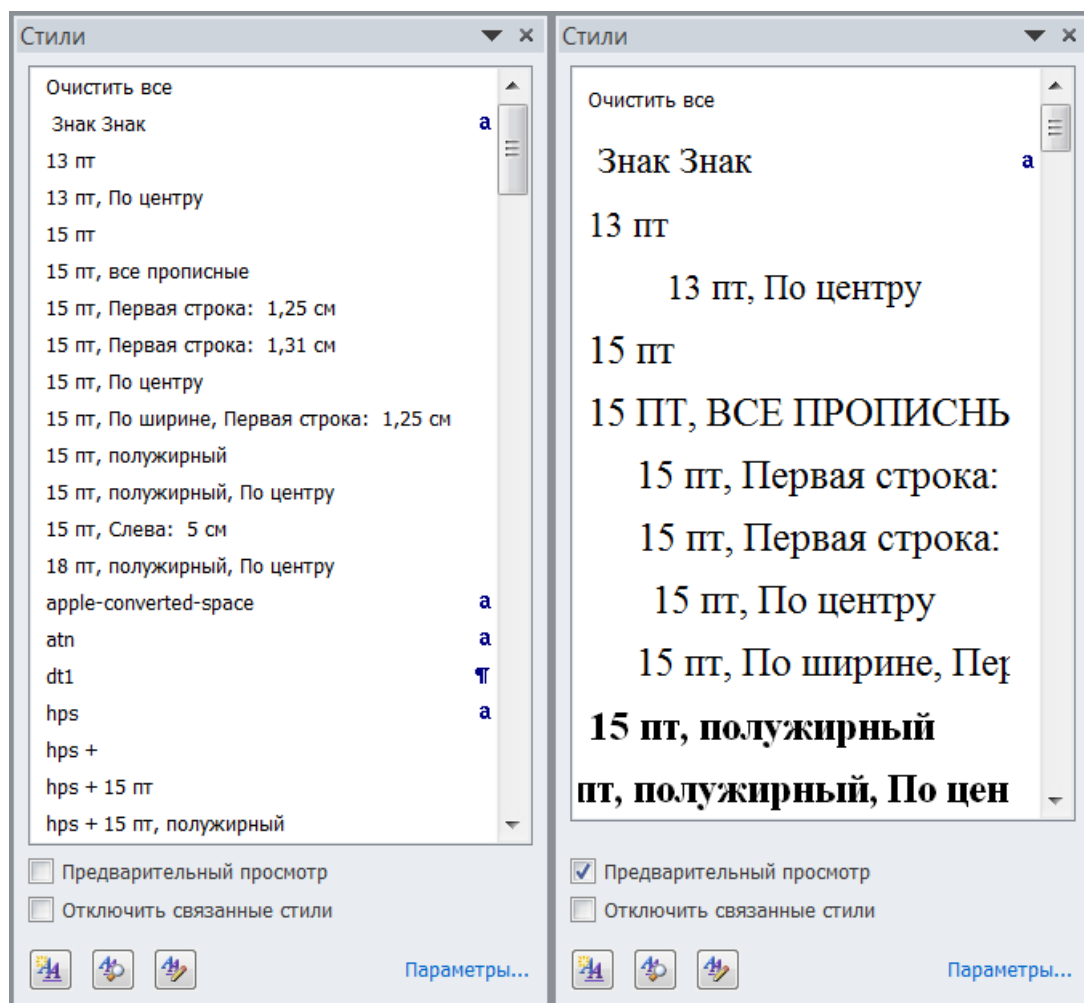


Рисунок 2.4 – Вікно стилів. Ліворуч – тільки назви, праворуч – назви з оформленням

Спосіб 2. Наведіть курсор на рядок із назвою стилю й натисніть кнопку списку. У меню виберіть пункт «Изменить» (рис. 2.5). Зверніть увагу, що якщо ви натиснете на сам рядок, то відповідний стиль буде застосований до виділеного фрагмента, або до майбутнього тексту від поточного положення курсора, якщо фрагмент не виділений; нам же потрібна кнопка меню.

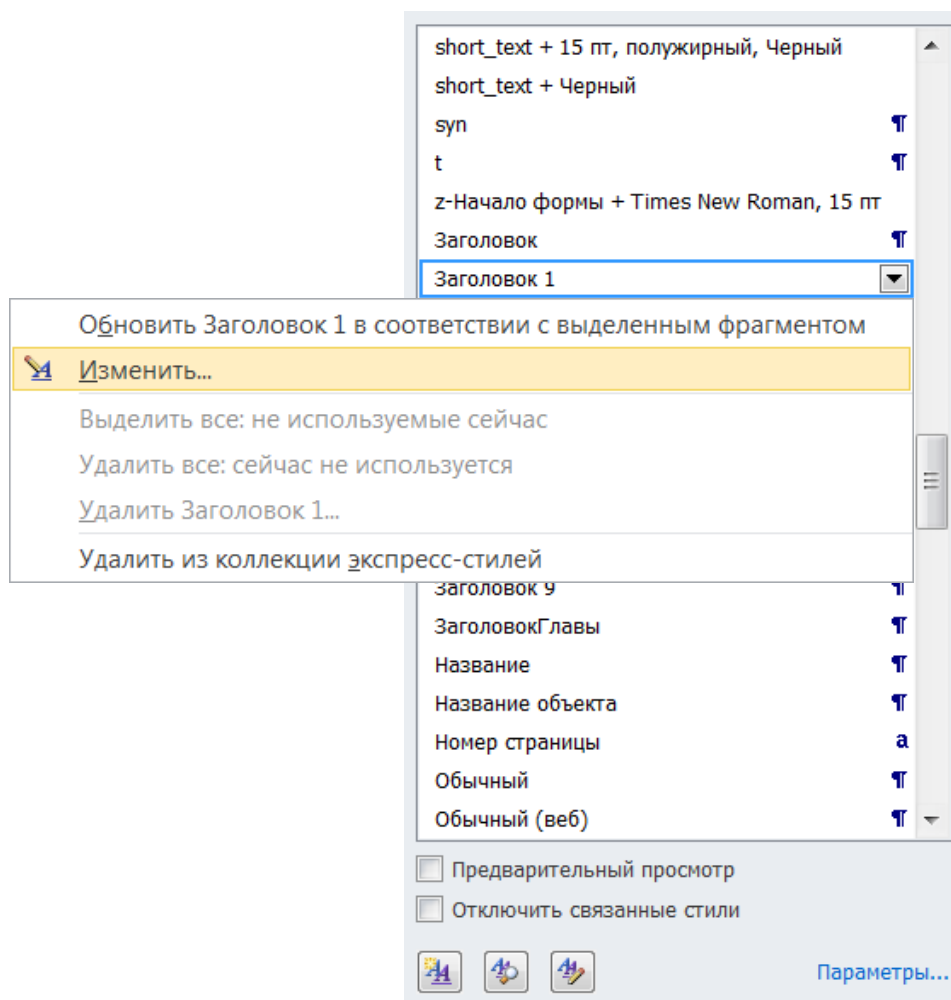


Рисунок 2.5 – Меню стилю

Буде відкрите вікно (рис. 2.6), де ви можете вибрати стиль із іншого набору або настроїти параметри самостійно, наприклад, змінити розмір шрифту або колір тексту. Зверніть увагу на список «Стиль следующего абзаца». Якщо ви вкажете стиль, що створюєте, то новий абзац (після натискання клавіші «Enter») буде оформлений так само. При налаштуванні заголовків краще вказати «Обычный», адже в переважній більшості випадків після заголовка слідує саме звичайний текст.

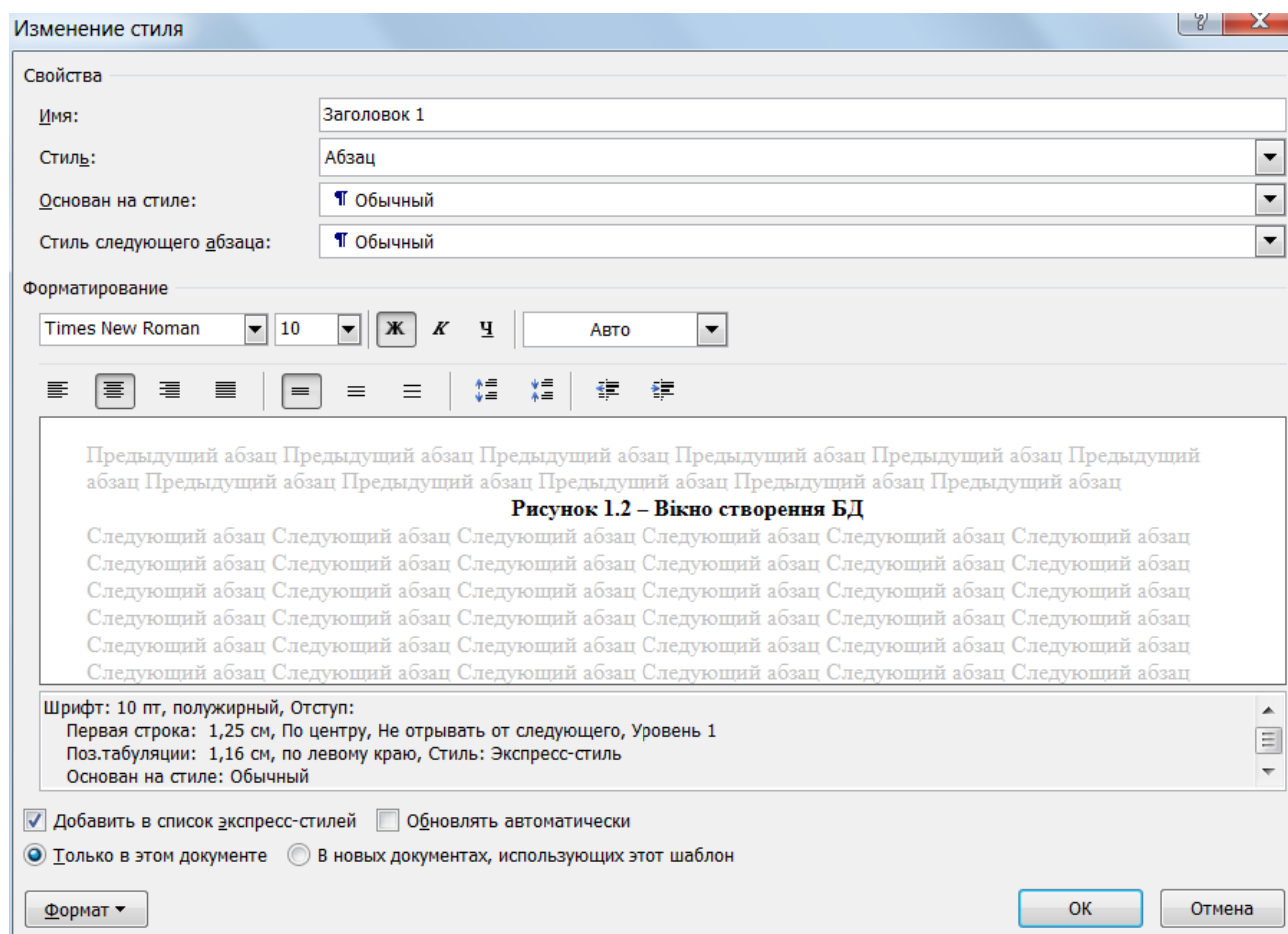


Рисунок 2.6 – Вікно зміни стилю

Якщо той або інший стиль не використовується, то ви можете вилучити його з меню (рис. 2.5), однак не завжди слід відмовлятися від нього повністю. Зніміть прапорець із пункту «Добавить в список экспресс-стилей», щоб вилучити зайву кнопку із вкладки «Главная», зберігши лише стилі, які використовуються. До інших ви зможете одержати доступ, відкривши плаваюче вікно.

Установивши прапорець «Обновлять автоматически», ви будете змінювати стиль і, відповідно, оформлення фрагментів тексту, в якому використовується цей стиль, щоразу, коли застосовуєте нове налаштування до тексту (автоматичний аналог першого способу).

Щоб одержати доступ до інших налаштувань, натисніть кнопку «Формат» і вибирайте потрібну вам групу параметрів зі списку (рис. 2.7).

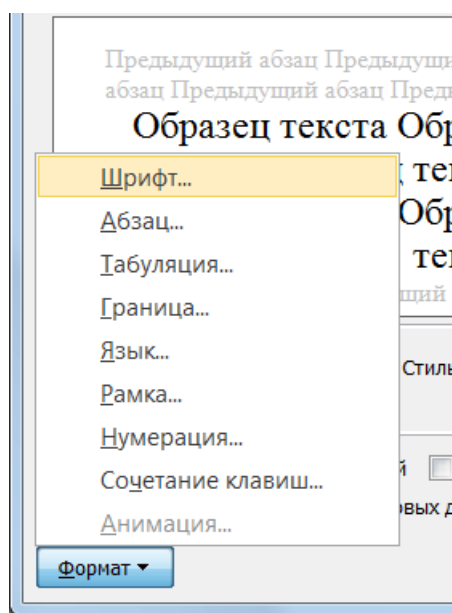


Рисунок 2.7 – Настроювання форматування стилю

Наприклад, клацнувши по пункту «Граница», ви викличете стандартне вікно «Границы и заливка» (рис. 2.8). Тут ви можете додати до заголовка підкреслення по лівому й нижньому краю.

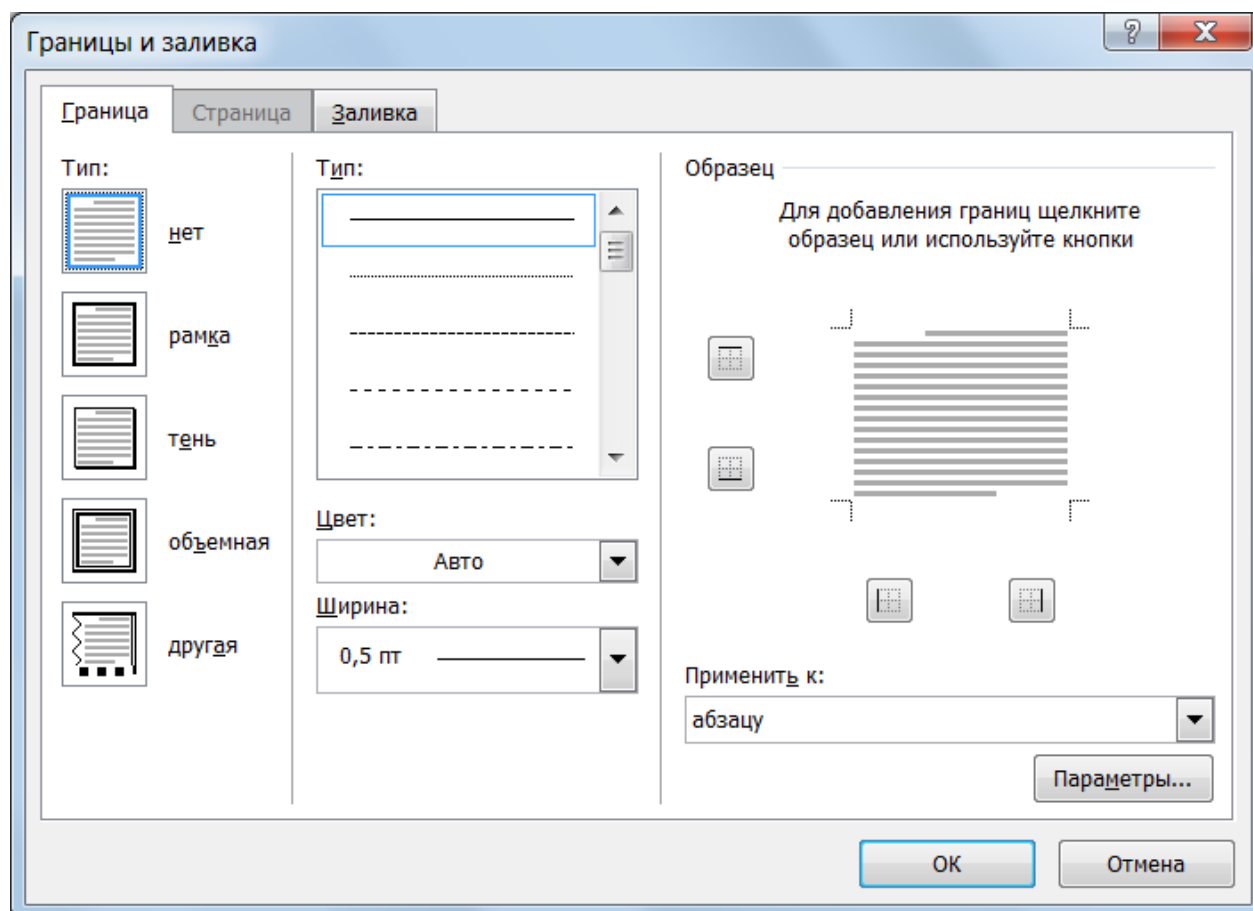


Рисунок 2.8 – Додаємо граничні лінії до заголовка

Замість того щоб змінювати наявні стилі, ви можете створити користувацький стиль. Натисніть кнопку «Создать стиль»  у вікні «Стили» (рис. 2.9) і робіть всі ті ж маніпуляції, що ви робили при редагуванні. Тут є лише дві основні відмінності:

1. Потрібно задати ім'я стилю (втім, ви могли перейменувати й наявний, але тут це більш важливо, тому що по іменам «Стиль1», «Стиль2» і т. д. орієнтуватися буде неможливо). Використовуйте «інформативне» найменування. Наприклад, ви можете використовувати назви «Стиль-Текст», «СтильГлава», «СтильТаблица» і т. д. Тоді буде зрозуміло для якого елемента розроблявся стиль.

2. Потрібно вказати тип у списку, що розкривається, «Стиль». Є п'ять заготовок:

- Абзац. Впливає тільки на окремі абзаци. Якщо окремі слова розмічені іншими стилями, то вони будуть збережені при застосуванні нового;
- Знак. Ці налаштування будуть впливати тільки на текст, але не на стиль абзацу. Природно, вам будуть недоступні вирівнювання, нумерація й інші «непотрібні» параметри;
- Зв'язаний (абзац і знак). Визначає зовнішній вигляд і абзацу, і тексту. Ідеально підходить для оформлення заголовків;
- Таблица. Впливає винятково на зовнішній вигляд таблиць;
- Список. Указує стиль оформлення маркірованих і нумерованих списків.

Зверніть увагу, що й при використанні нових стилів ви зможете швидко змінити колірну гаму документа. Однак для цього ви повинні використовувати тільки кольори теми (рис. 2.10).

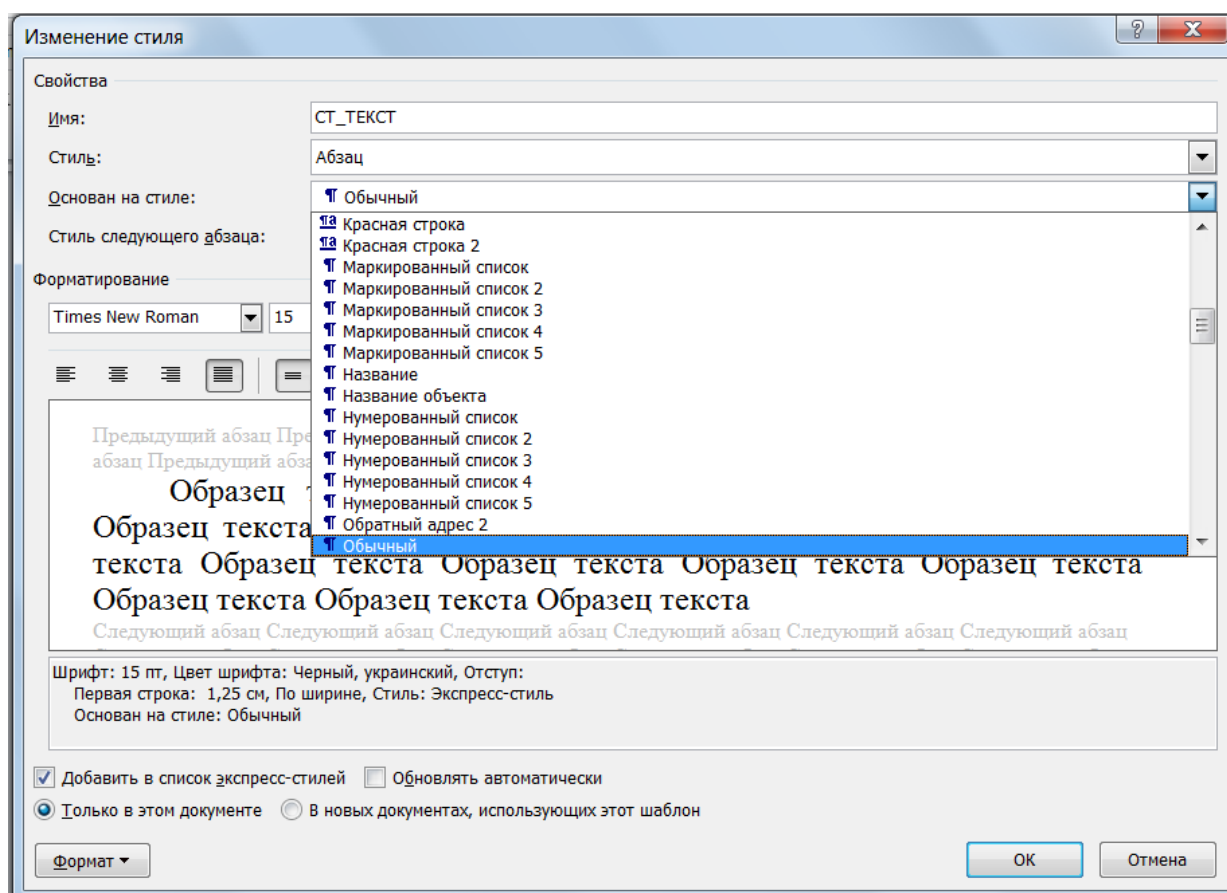


Рисунок 2.9 – Створення нового стилю

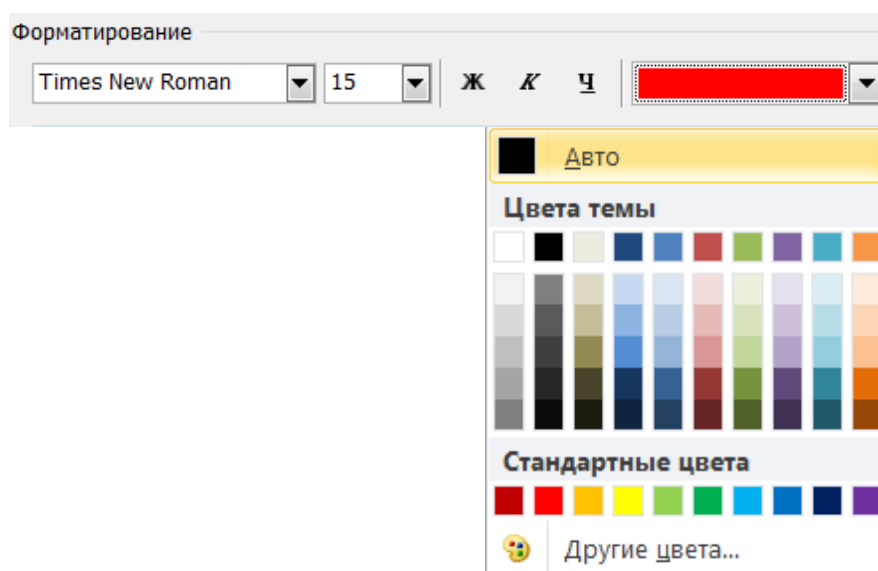


Рисунок 2.10 – Вибір кольору тексту

Якщо ви вибираєте інші кольори, то вони не будуть змінюватися при виборі нової колірної теми.

Після створення користувацького набору стилів, у Word 2010 є кілька можливостей подальшого використання стилів.

По-перше, ви можете обрати копію документа зі стилями, вилучити з нього весь текст і вставити (або вписати) новий уміст. Найкраще вилучити весь уміст із файлу перед збереженням його у вигляді шаблону. Стили залишаться навіть в «порожньому» документі. Не забудьте поставити на файл атрибут «Только для чтения», щоб ніхто його не перезаписав помилково.

По-друге, ви можете зберегти поточний набір стилів у окремому файлі. Для цього виберіть пункт меню «Сохранить как набор экспресс-стилей» (рис. 2.11).

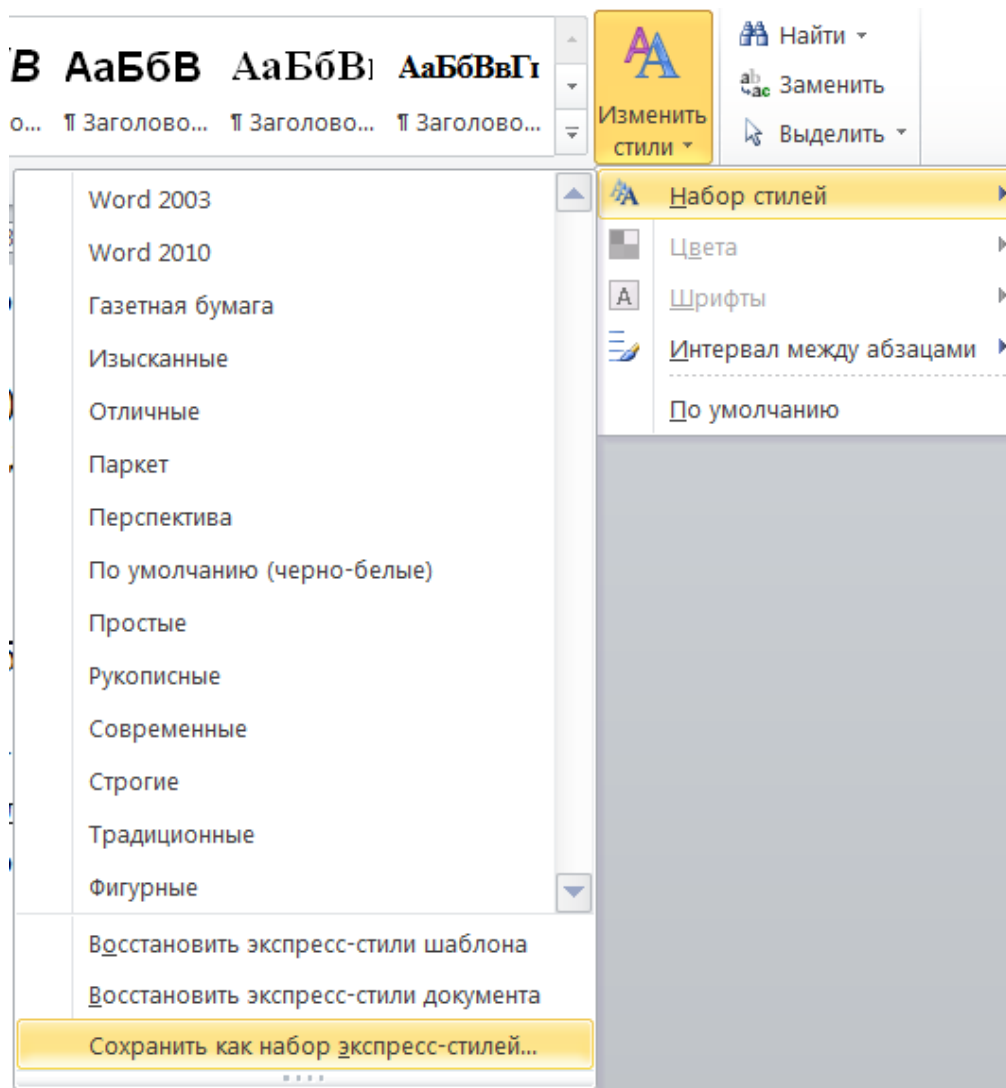


Рисунок 2.11 – Збереження набору стилів

Буде викликано діалогове вікно збереження файлу у форматі шаблону Word (*.dotx). Щоб скористатися ним, скопіюйте текст у порожній документ шаблону й збережіть під новим іменем або поверх файлу з документом без оформлення.

Зміст

Найбільш очевидне застосування – створення змісту. Його звичайно розміщують на початку або наприкінці книги, але для електронного документа найбільш актуальним буде варіант із розміщенням на початку, особливо якщо ви створюєте документ із невеликою кількістю розділів. Варто відзначити, що зміст електронного документа дозволяє здійснювати швидку навігацію. Досить клацнути мишею по рядку списку, утримуючи клавішу «Ctrl», щоб перейти до відповідного фрагмента.

Для вставки змісту відкрийте вкладку «Ссылки» і виберіть варіант із автоматичним складанням (рис. 2.12).

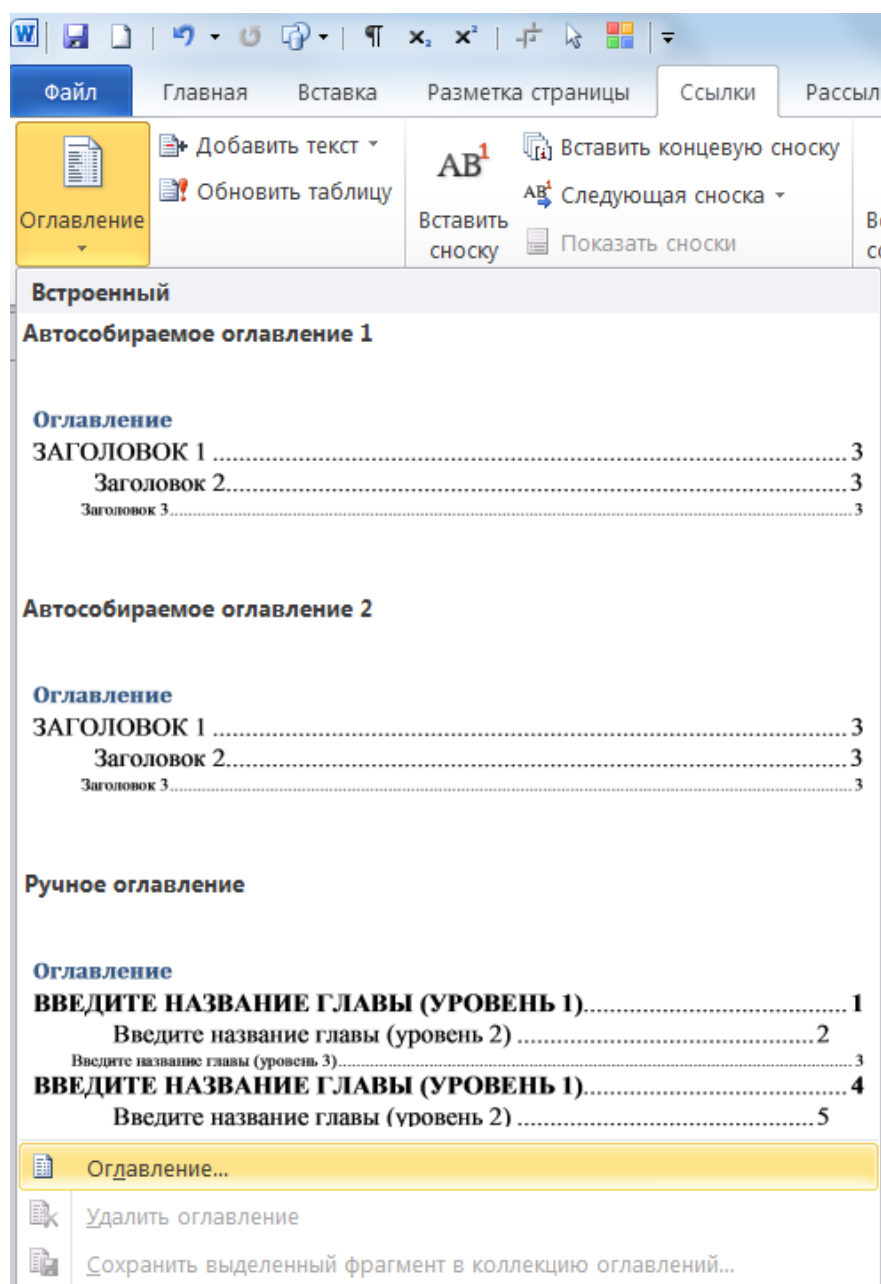


Рисунок 2.12 – Вставка змісту

Зразковий формат відображається вже в списку вибору – тут також використовуються стилі вашого документа. Якщо не влаштовує стандартний формат, можна зайти в меню «Оглавление» і настроїти його відповідно до необхідних вимог.

При використанні стилів ви значно збільшите ефективність своєї роботи. Вони не тільки дозволяють швидко змінювати оформлення документів, але й визначають структуру, що використовується при читанні, пошуку та при створенні змісту.

Добре продуманий шаблон зі стилями допоможе підняти ефективність роботи, а також привести велику кількість однотипних документів до єдиного стандарту оформлення.

Створення й оновлення змісту

Для створення змісту потрібно застосувати до тексту, який потрібно в нього включити, стилі заголовків, такі як «Заголовок 1», «Заголовок 2» або «Заголовок 3». Додаток Microsoft Word шукає ці заголовки й вставляє в документ зміст.

Створивши зміст таким способом, можна автоматично оновлювати його при внесенні в документ змін.

Microsoft Word 2010 містить колекцію автоматичних стилів змісту. Позначте елементи змісту й виберіть стиль змісту в колекції.

Створити зміст, що настроюється з обраними параметрами й користувачькими стилями, можна за допомогою діалогового вікна

Уведення змісту вручну

Елементи змісту можна ввести вручну й використати табуляцію для створення пунктирних ліній або відступів у вигляді крапок між записами та номерами сторінок:

- 1) уведіть перший запис;
- 2) натисніть клавішу «ТАВ» і введіть номер сторінки для першого запису;
- 3) виберіть символ табуляції (якщо символи табуляції не відображаються, на вкладці «Главная» у групі «Абзац» оберіть команду «Показать или скрыть»);
- 4) на вкладці «Разметка страницы» натисніть кнопку виклику діалогового вікна «Абзац»;
- 5) натисніть кнопку «Табуляция»;
- 6) у поле «Позиции табуляции» уведіть позицію для розміщення номера сторінки; щоб відобразити лінійку для точної вказівки відступу пра-

воруч, натисніть кнопку «Показать линейку» вгорі вертикальної смуги прокручування;

7) у групі «Выравнивание» оберіть варіант «По правому краю»;

8) у групі «Заполнитель» оберіть потрібні параметри й натисніть кнопку «ОК»;

9) натисніть клавішу «ВВОД», а потім уведіть текст для нового запису;

10) натисніть клавішу «ТАВ» і введіть номер сторінки для другого запису;

11) повторюючи ці дії, уведіть увесь зміст.

Якщо заголовки або сторінки в документі були змінені, зміст необхідний буде оновити вручну.

Створення змісту автоматично

Найбільш простим способом створити зміст є використання вбудованих стилів заголовків («Заголовок 1»,...«Заголовок 9»). Крім того, можна створити зміст за допомогою користувацьких стилів, які були застосовані до документа. Можна також призначити рівні елементів змісту окремим фрагментам тексту.

Позначка елементів змісту із застосуванням вбудованих стилів заголовків

1. Виділіть текст, який потрібно включити в зміст.

2. На вкладці «Главная» у групі «Стили» виберіть необхідний стиль (рис. 2.13).

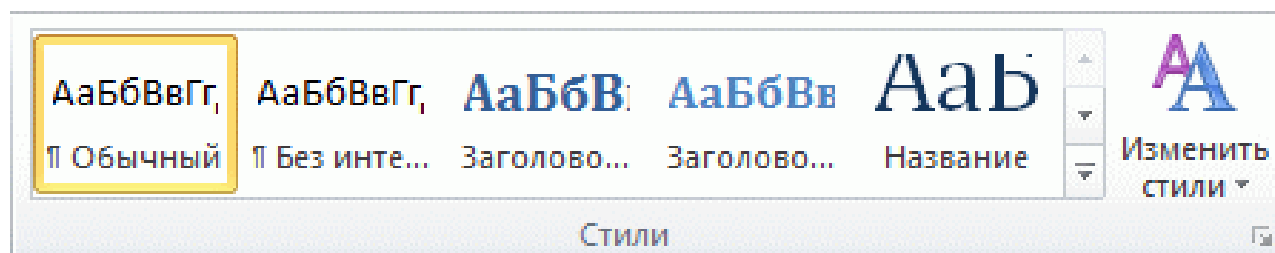


Рисунок 2.13 – Панель стилів

Наприклад, існує виділений текст, до якого слід застосувати стиль основного заголовка, виберіть у колекції «Экспресс-стили» стиль із іменем «Заголовок 1».

Створення змісту з колекції

Після того як усі елементи змісту будуть позначені, можна приступити до складання змісту.

1. Клацніть у документі місце, куди потрібно вставити зміст (звичайно це початок документа).

2. На вкладці «Ссылки» у групі «Оглавление» виберіть команду «Оглавление», а потім виберіть необхідний стиль змісту (рис. 2.14).

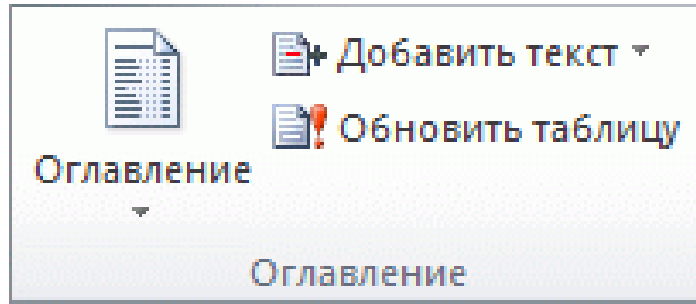


Рисунок 2.14 – Панель «Оглавление»

Щоб задати додаткові параметри, наприклад кількість рівнів заголовків, які повинні відображатися, натисніть кнопку «Оглавление». Відкриється діалогове вікно «Оглавление».

Створення змісту, що настраюється

На вкладці «Ссылки» у групі «Оглавление и указатели» клацніть «Оглавление», а потім виберіть у списку пункт «Добавить оглавление».

1. У діалогові вікні «Оглавление» виконайте кожну із зазначених нижче дій:

1) щоб змінити кількість рівнів заголовків, що відображаються і змісті, введіть потрібне число в поле виведення рівнів у розділі «Общее»;

2) щоб змінити загальний вид змісту, виберіть інший формат у списку «Форматы». В областях «Предварительный просмотр» і «Образец веб-документа» можна побачити, як виглядає обраний варіант;

3) щоб змінити тип ліній, що відображаються між текстом записів і номерами сторінок, оберіть потрібний варіант у списку «Символ заполнения»;

4) щоб змінити спосіб відображення рівнів заголовків у змісті, натисніть кнопку «Изменить». У діалогові вікні «Стиль» виберіть рівень, який необхідно змінити, і натисніть кнопку «Изменить». У діалогові вікні «Изменение стиля» можна вибрати шрифт, розмір і відступ.

2. Щоб використовувати в змісті стилі, що настроюються, натисніть кнопку «Параметры» і виконайте зазначені нижче дії:

1) у групі «Доступные стили» знайдіть стиль, що застосовується до заголовків у документі;

2) щоб указати рівень, який повинен бути представлений стилем заголовка, уведіть ціле число від 1 до 9 у поле групи «Уровень», що розташоване поруч із іменем стилю. Якщо потрібно застосувати тільки користувацькі стилі, вилучте номери рівня для вбудованих стилів, наприклад для стилю «Заголовок 1».

3) повторіть дії 1 і 2 для кожного стилю заголовка, який потрібно включити в зміст;

4) натисніть кнопку «ОК».

3. Виберіть зміст відповідно до типу документа, як зазначено нижче.

Паперовий документ. При створенні документа, який читачі одержать у друкованому вигляді, слід створювати зміст таким чином, щоб кожний елемент складався із заголовка й номера сторінки, на якій розташований цей заголовок. У цьому випадку читачі зможуть звернутися до потрібної сторінки.

Веб-документ. У документі, який читачі одержать по мережі й будуть читати в Word, можна формувати елементи змісту у вигляді гіперпосилань, щоб можна було перейти до потрібного заголовка, клацнувши його в змісті.

Форматування змісту

Додавши в документ Word зміст, можна настроїти його зовнішній вигляд. Наприклад, можна вказати, скільки рівнів заголовків повинне відображатися, а також вивести лінії із крапками між назвами розділів і відповідними номерами сторінок.

Також можна змінити формат тексту змісту, і він збережеться незалежно від того, скільки разів зміст буде оновлений.

Вставка змісту, що настроюється

Якщо в документі вже є зміст, ця процедура призведе до його зміни відповідно до настроювань.

1. На вкладці «Ссылки» у групі «Оглавление» натисніть кнопку «Оглавление» і виберіть пункт «Оглавление» (рис. 2.15).

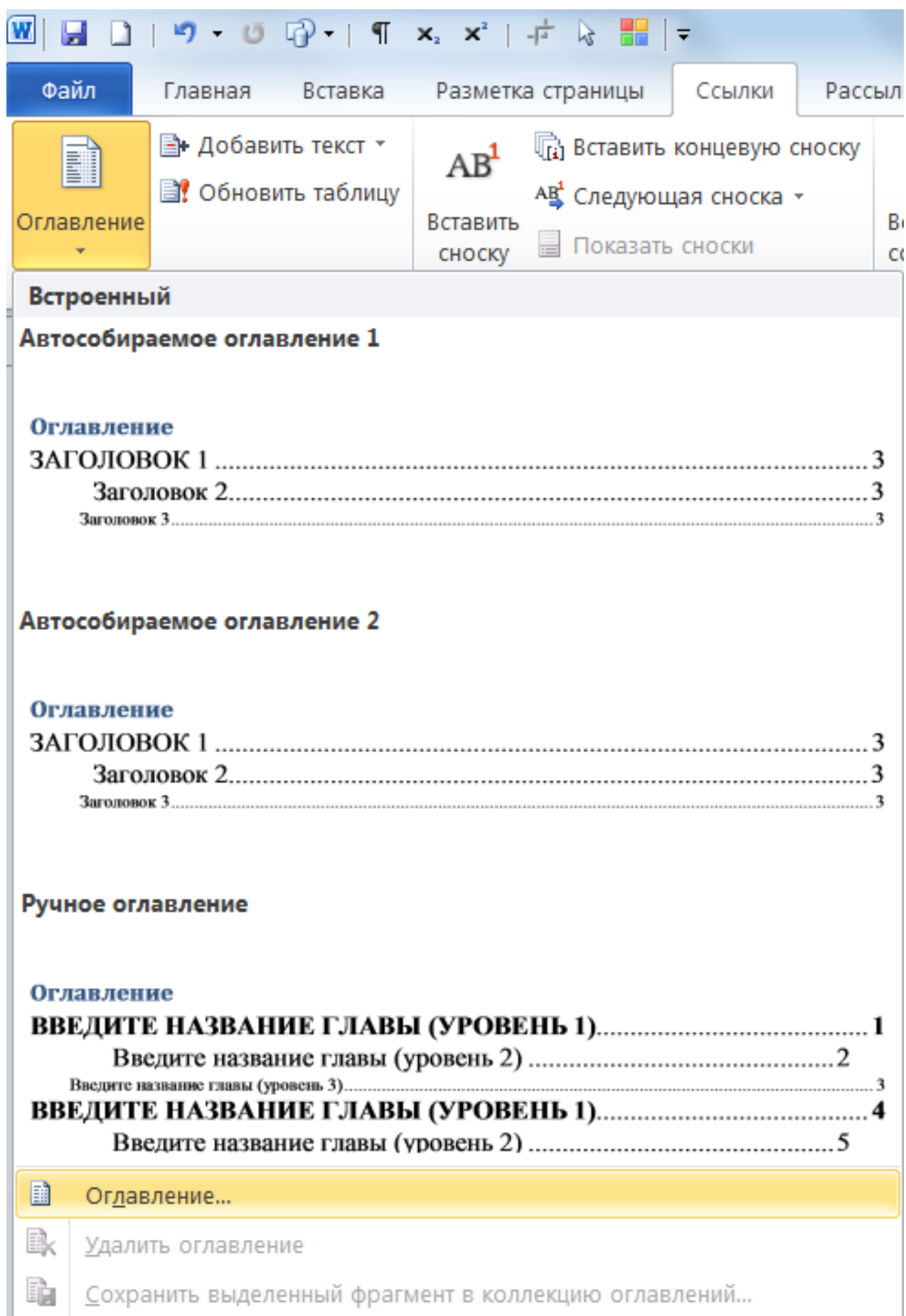


Рисунок 2.15 – Панель «Оглавление»

Відкриється діалогове вікно «Оглавление» (рис. 2.16).

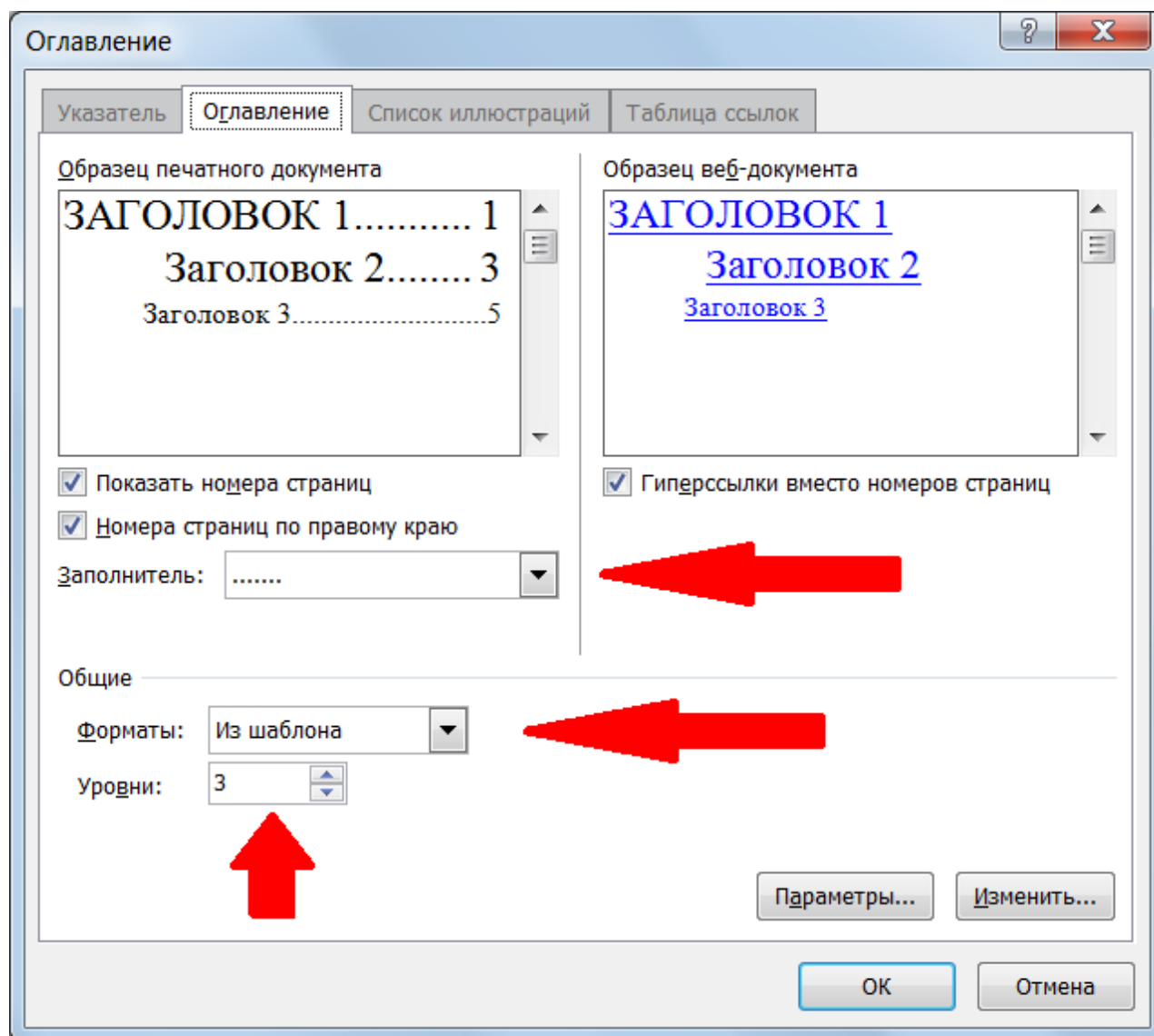


Рисунок 2.16 – Вікно «Оглавление»

2. Виконаєте одну з дій, що зазначені нижче:

1) щоб додати лінію із крапками між назвою кожного з розділів і відповідним номером сторінки, відкрийте список «Заполнитель» і виберіть лінію із крапками;

2) щоб змінити зовнішній вигляд змісту в цілому, відкрийте список «Форматы» і виберіть потрібний формат;

3) щоб змінити кількість рівнів, які повинні відображатися в змісті, задайте потрібне число в поле «Уровни».

Форматування тексту в змісті

Щоб змінити формат тексту в змісті, що створений додатком Word, необхідно задати стиль для кожного його рівня. Це ніяк не пов'язано зі зміною стилів, що застосовані до заголовків у документі.

Змінений стиль буде використовуватися додатком Word при кожному відновленні змісту.

1. На вкладці «Ссылки» у групі «Оглавление» натисніть кнопку «Оглавление» і виберіть пункт «Оглавление».

2. У діалогові вікні «Оглавление» натисніть кнопку «Изменить» (рис. 2.17).

3. У списку «Стили» виберіть рівень, який потрібно змінити, і натисніть кнопку «Изменить» (рис. 2.18).

4. У діалогові вікні «Изменение стиля» внесіть необхідні зміни у формат і натисніть кнопку «ОК».

5. Повторіть дії 3 і 4 для всіх рівнів, які повинні відображатися в змісті.

Відновлення змісту

Якщо заголовки або інші елементи змісту були додані в документ або вилучені з нього, можна швидко оновити зміст.

1. На вкладці «Ссылки» у групі «Оглавление» виберіть команду «Обновить таблицу».

2. Установіть перемикач у положення «обновить только номера страниц» або «обновить целиком».

Видалення змісту

На вкладці «Ссылки» у групі «Оглавление» натисніть кнопку «Оглавление».

Виберіть команду «Удалить оглавление».

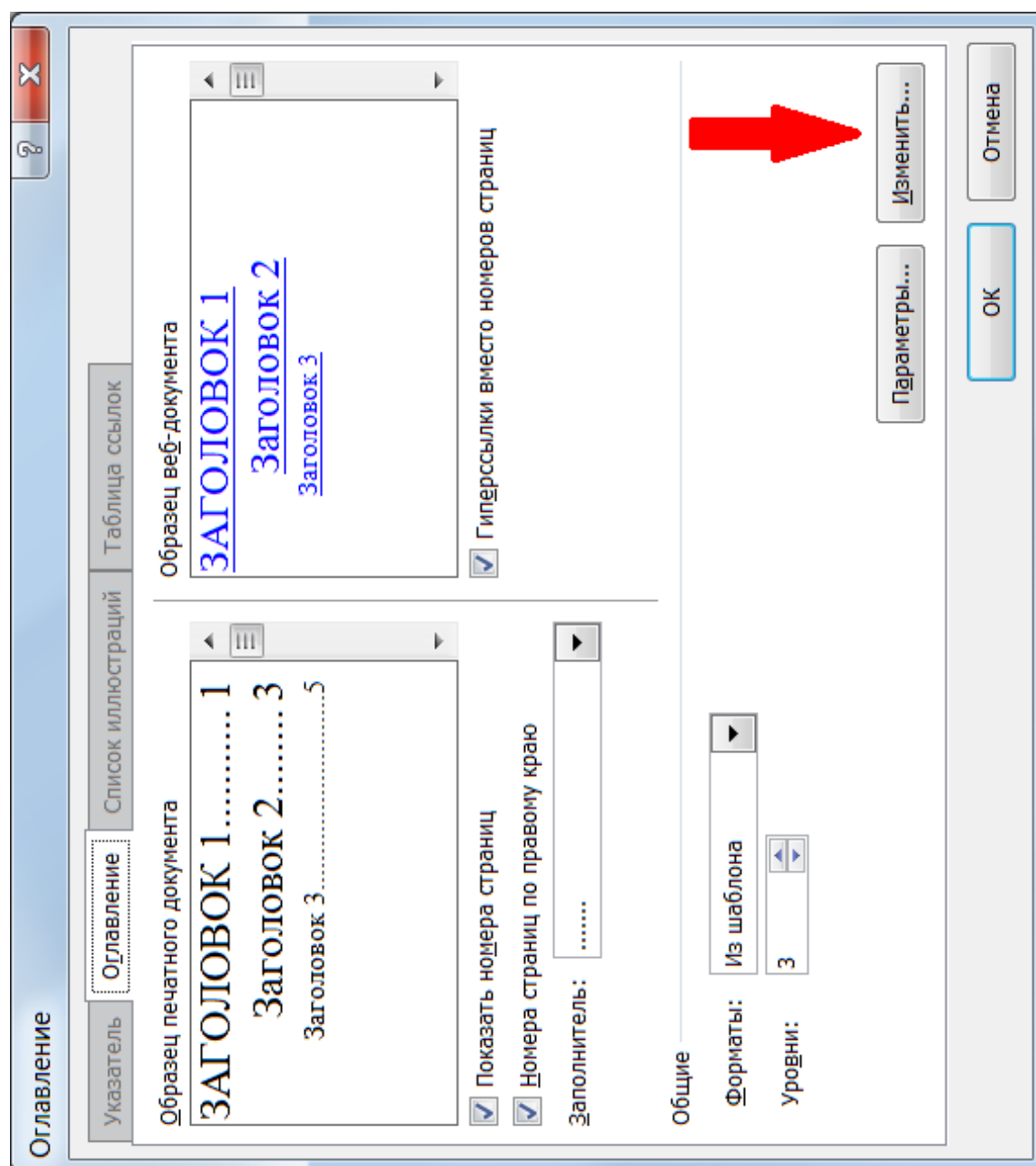


Рисунок 1.17 – Вікно «Оглавление»

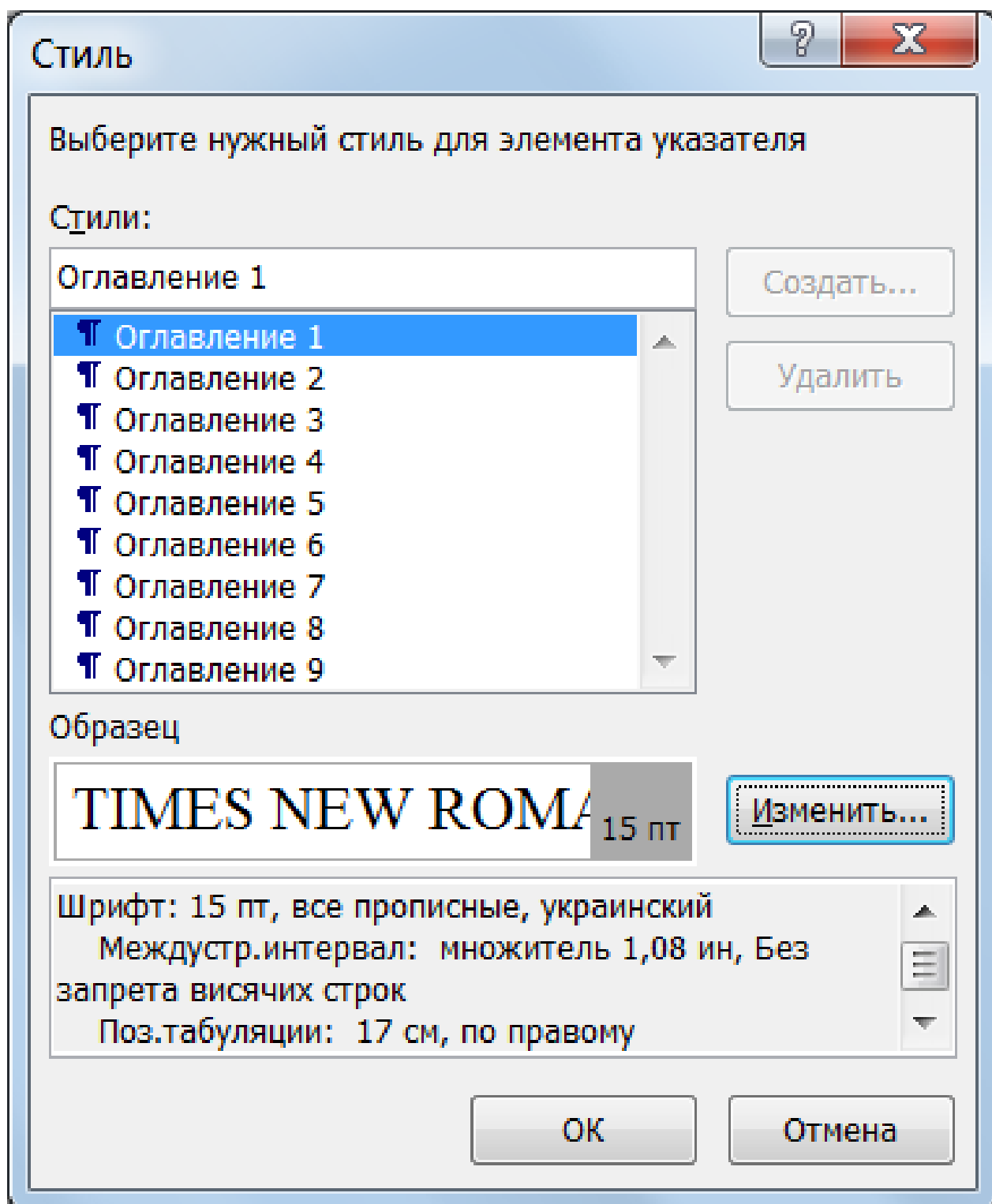


Рисунок 2.18 – Вікно «Стиль»

Відомості про виконання курсової роботи

Курсова робота – це самостійне навчально-наукове дослідження студента, яке виконується з певного навчального курсу або за певними його розділами. Вона повинна вирішувати технічне завдання дисципліни (курсу) і бути пророблена до рівня пропозиції застосування.

Курсова робота виконується з метою закріплення, поглиблення й узагальнення знань, що отримані студентами піл час навчання, і їх застосування до комплексного розв'язку конкретного професійного завдання.

Тематика курсових робіт повинна відповідати завданням навчальної дисципліни й бути тісно пов'язана з практичними питаннями конкретної спеціальності. Вона затверджується на засіданні кафедри. Строки виконання курсових робіт і час їх захисту визначається навчальним планом.

Сучасна методика підготовки фахівців із вищою освітою вимагає обов'язкової наявності в згаданих роботах елементів наукових досліджень. Тому визначальними стандартами на структуру й оформлення результатів таких робіт прийняті:

1) ДСТУ 3008-95 «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення»;

2) Держстандарт 2.105-95 «Загальні вимоги до текстових документів».

Курсова робота на тему: «Створення проектів, робота із зовнішніми файлами, масиви, модулі, графіка» передбачає реалізацію програмного додатка, який представляє собою комплекс програмних модулів, які формалізують розв'язання задач різноманітної спрямованості на мови програмування Visual Basic .Net (базові поняття з цього питання розглянуті у [8]), що входить до складу інтегрованого середовища розробки програмного забезпечення Microsoft Visual Studio.

Після створення робочого проекту та його ретельного тестування, ключові етапи формалізації задач, представлення алгоритмів у вигляді блок-схем та результати тестування відображаються у відповідних розділах звіту з курсової роботи. Крім того звіт повинен містити короткі теоретичні відомості з обраної тематики у вигляді реферативних повідомлень. Звіт оформлюється згідно з вимогами [10-11]

Перелік практичних індивідуальних завдань та блок теоретичних питань автоматично генерується у програмному мультимедійному навчальному комплексі «АРМ викладач-студент» [12]. Викладач за власним розсудом може вносити корективи до цього аркуша завдань, але загальна спрямованість задач та теоретичних питань повинна залишатися незмінною.

2.2 Практичне завдання

Створити шаблон звіту з курсової роботи, яких обов'язково повинен містити наступні розділи та елементи.

1. Титульний аркуш (дивіться додаток В).

2. Лист завдань (видається на першому практичному занятті).
3. Реферат (анотація) до курсової роботи (дивіться приклад у додатку В, вимоги у [10]).
4. Зміст (дивіться приклад у додатку В).
5. Вступ (дивіться вимоги у [10]).
6. Розділ 1 (теоретична частина).
7. Розділ 2 (практична частина).
8. Висновки (дивіться вимоги у [10]).
9. Перелік посилань (перелік джерел та використаної літератури).

При оформленні звіту з курсової роботи необхідно дотримуватися наступних вимог.

1. Макет сторінки. Поля: верх – 2 см., низ – 2 см., зліва – 3 см., праворуч – 1 см.

2. Шрифт – Times New Roman, розмір шрифту – 14.

3. Абзац. Інтервал – 1,5. Відступ першого рядка – 1,25.

Структурні елементи «РЕФЕРАТ», «ЗМІСТ», «ВСТУП», «ВИСНОВКИ», «ПЕРЕЛІК ПОСИЛАНЬ» починаються з нової сторінки, не нумерують, пишуть великими буквами, крапку не ставлять.

Інші елементи оформлюються згідно таблиці 2.1.

Таблиця 2.1 – Стилi форматування

Елемент	Параметри
«РЕФЕРАТ», «ЗМІСТ»	Без стилю. Times New Roman, 14 пт, всі прописні, по центру
Рівень 3	Стиль СТ_КР_3 Times New Roman, 14 пт, відступ – 2 см., вирівнювання зліва, 1,5 інтервал, стиль наступного абзацу – «Обычный»
Рівень 2	Стиль СТ_КР_2 Times New Roman, 14 пт, відступ – 1,25 см., вирівнювання зліва, 1,5 інтервал стиль наступного абзацу – «Обычный»
Рівень 1	Стиль СТ_КР_1 Times New Roman, 14 пт, всі прописні вирівнювання за центром, 1,5 інтервал, стиль наступного абзацу – «Обычный»
Текст пояснювальної записки (текст курсової роботи)	СТ_КР_Абзац Times New Roman, 14 пт, відступ – 1,25 см., вирівнювання по ширині, 1,5 інтервал, стиль наступного абзацу – «Обычный»

3 ЛАБОРАТОРНА РОБОТА №3

Тема: електронні таблиці MS Excel. «Підбор параметру»

Мета: засвоїти основні принципи роботи з електронними таблицями MS Excel. Навчитися користуватися інструментом оптимізації «Підбір параметру»

3.1 Короткі теоретичні відомості

Сформулюємо задачу пошуку коренів кубічного рівняння:

$$x^3 - 0,01x^2 - 0,7044x + 0,139104 = 0$$

Відомо, що поліном третього ступеня має не більш трьох різних дійсних коренів. Для того, щоб знайти дійсні корені алгебраїчного рівняння, спочатку їх потрібно локалізувати. З цією метою належить побудувати графік функції або її протабулювати. Протабулюємо наш поліном і побудуємо його графік на відрізку $x \in [-1, 1]$, із кроком 0,2. Для цього у діапазоні A2:A12 занесемо значення змінної x : -1; -0,8 ... 1. Потім у комірку B2 запишемо формулу:

$$=A2^3-0,01*A2^2-0,7044*A2+0,139104$$

Простягнемо її на діапазон B2:B12. По отриманим даним побудуємо точковий графік. Таблиця та графік наведені на рисунку 3.1.

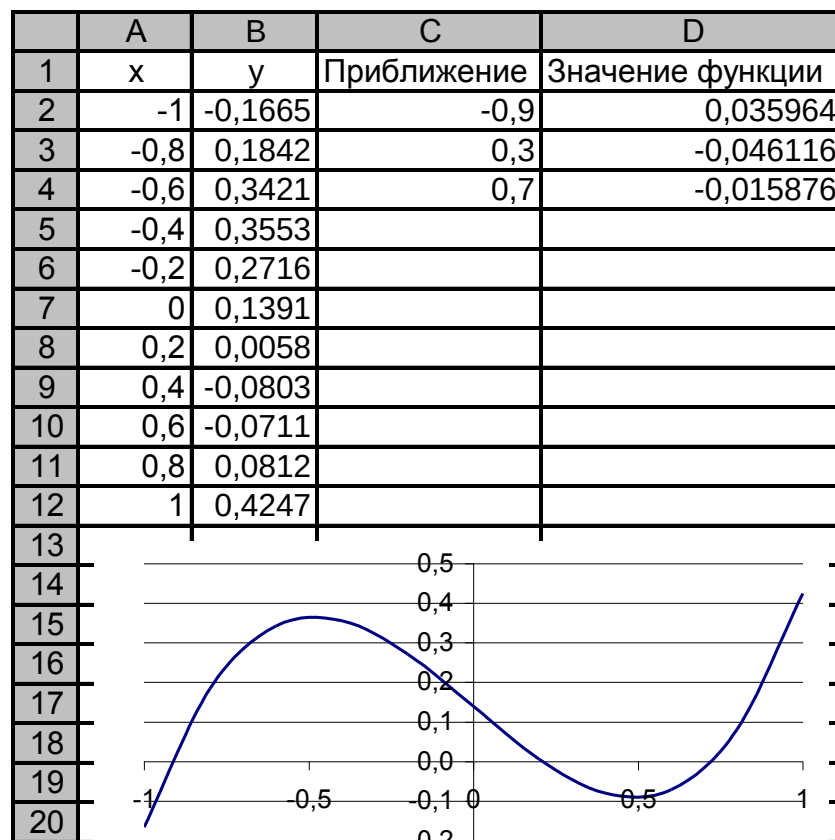


Рисунок 3.1 – Таблиця та графік коренів рівняння

З графіка видно, що крива перетинає вісь x у трьох точках, виходить, ми локалізували всі три корені. Як початкові наближення до коренів можна взяти будь-які точки з відрізків локалізації коренів. Візьмемо, наприклад, точки $-0,9$; $0,3$; $0,7$ і введемо їх у діапазон $C2:C4$. У комірку $D2$ введемо формулу:

$$=C2^3-0,01*C2^2-0,7044*C2+0,139104$$

Виділимо цю комірку й за допомогою маркера заповнення протягнемо введену в неї формулу на діапазон $D2:D4$. Таким чином, у комірках $D2:D4$ обчислюються значення полінома при значеннях аргументу, введенного в комірки $C2:C4$, відповідно. Знайдемо корені полінома за допомогою команди: вкладка «Данные→Анализ «что если»→Подбор параметра...». Відносна похибка обчислень і граничне число ітерацій задаються на вкладці «Файл → Параметры → Формулы». Задамо відносну похибку й граничне число ітерацій $0,00001$ і 1000 , відповідно.

Тепер виконаємо команду: вкладка «Данные→Анализ «что если»→Подбор параметра...», і заповнимо діалогове вікно «Подбор параметра» так, як показано на рисунку 1 і натискаємо кнопку **Ок**. Результат виконання команди показаний на рисунку 3.2 Знайдений корінь команда заносить у комірку $D2$. Цю же процедуру пошуку кореня повторюємо для комірок $D3$, $C3$ і $D4$, $C4$.

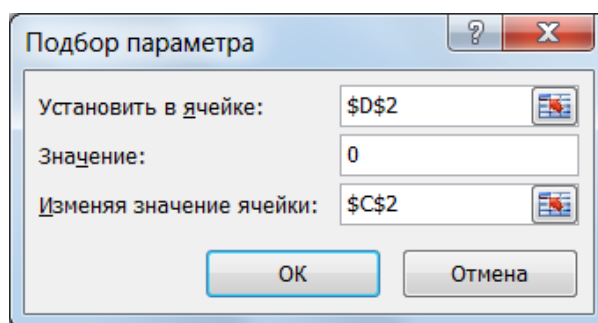


Рисунок 3.2 – Підбір параметра

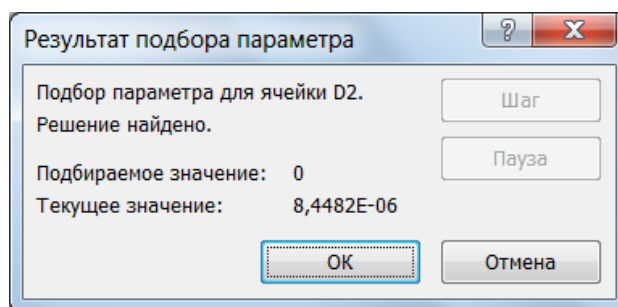


Рисунок 3.3 – Результат підбору параметра

3.2 Практичне завдання

Знайти всі дійсні корені рівняння

Таблиця 3.4 – Варіанти індивідуальних завдань

Варіант	Рівняння	Варіант	Рівняння
1	2		4
1	$x^3 - 2,92x^2 + 1,4355x + 0,79 = 0$		$x^3 - 8,96x^2 - 2,5x + 111,4 = 0$
3	$x^3 - 2,56x^2 - 1,32x + 4,39 = 0$		$x^3 + 5x^2 + 0,84x - 5 = 0$
5	$x^3 + 2,84x^2 - 5,60x - 14,76 = 0$		$x^3 + 2,89x^2 - 22,88x + 10 = 0$
7	$x^3 + 1,41x^2 - 5,47x - 7,38 = 0$		$x^3 - 10x^2 + 11,19x + 81,27 = 0$
9	$x^3 + 10x^2 - 2x - 100 = 0$	0	$x^3 - 3,7x^2 - 43,1x + 69,6 = 0$
11	$x^3 + x^2 - 40x + 0,016 = 0$		$x^3 - 4,7x^2 - 23,7x - 21,6 = 0$

		2	
13	$x^3 + 0,88x^2 - 10x - 10 = 0$	4	$x^3 + 2,8x^2 - 60,84x - 170 = 0$
15	$x^3 - 0,12x^2 - 1,47x + 0,19 = 0$	6	$x^3 + 12,9x^2 + 36,5x + 3,52 = 0$
17	$x^3 + 0,78x^2 - 2x + 0,146 = 0$	8	$x^3 + 5,4x^2 - 4x - 21,6 = 0$
19	$x^3 + 2,28x^2 - 1,93x - 3,907 = 0$	0	$x^3 + 3,7x^2 - 54x - 56,7 = 0$
21	$x^3 + 0,45x^2 - 12x + 0,125 = 0$	2	$x^3 + 16x^2 + 76x + 96 = 0$
23	$x^3 + 0,415x^2 - 5,6x - 3 = 0$	4	$x^3 + 19x^2 + 111x + 189 = 0$
25	$x^3 + 0,897x^2 - x - 0,5 = 0$	6	$x^3 - 3x^2 - x + 3 = 0$
27	$x^3 - 6x^2 - 9x + 54 = 0$	8	$x^3 - 4x^2 - 9x + 36 = 0$
29	$x^3 - 8,8x^2 - 47,19x + 359 = 0$	0	$x^3 - 5,36x^2 - 64x + 343 = 0$

4 ЛАБОРАТОРНА РОБОТА №4

4.1 Теоретичні відомості

Для запуску програми з головного меню Windows, виконайте наступні дії.

1. Натисніть кнопку Пуск, розташовану в нижній частині екрану.
2. У головному меню Windows оберіть команду Програми. З'явиться меню даної команди.
3. Виберіть у меню пункт Microsoft Visual Studio.
4. У вікні, що з'явилося оберіть Microsoft Visual Studio. На екрані з'явиться головне вікно Visual Studio (рис. 1.1).

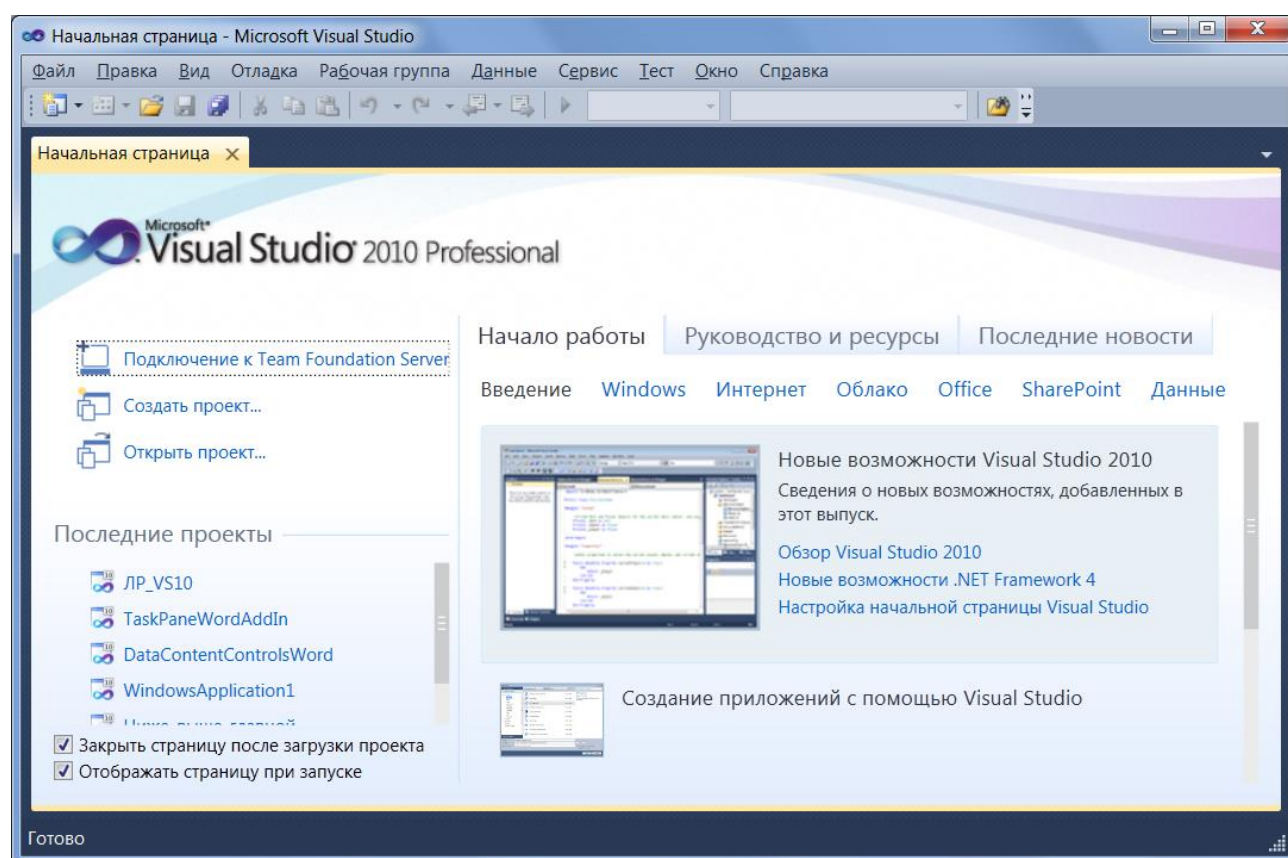


Рисунок 1.1 – Початкова сторінка середовища програмування Visual Studio 2010

Програми, які розробляються у Visual Studio, звичайно називаються проектами або рішеннями, так як вони містять не один-єдиний файл, а багато окремих компонентів. Для програм на Visual Basic .NET завжди є файл проекту (.vbproj) і файл рішення (.sln). Файл проекту містить інформацію, що відноситься до однієї програмної задачі. Файл рішення містить інформацію про один або декілька проектів. Файли рішень використовуються для управління декількома взаємопов'язаними проектами.

Після створення проекту зазвичай відкривається вся сукупність вікон та панелей середовища програмування (рис. 1.2).

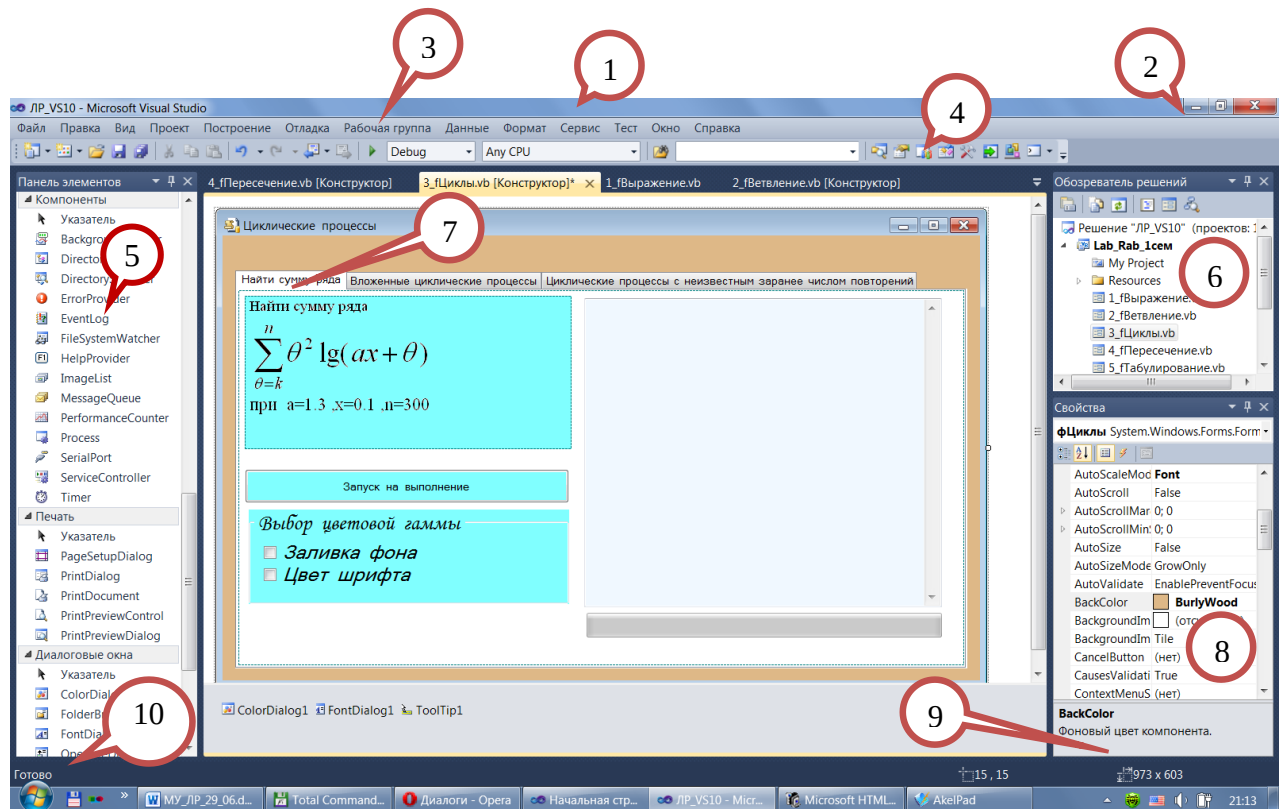


Рисунок 1.2 – Середовище програмування Visual Studio 2010

1. Заголовок головного вікна. Містить іконку Visual Studio 2010, назву поточного проекту, назву робочого середовища (Microsoft Visual Studio).

2. Кнопки головного вікна згорнути, згорнути / розгорнути у вікно й закрити.

3. Головне меню. Містить пункти, кожен з яких має свої підпункти. Забезпечує швидкий доступ до необхідних функцій.

4. Панель інструментів (ПІ). Дозволяє отримати швидкий доступ до найбільш часто використовуваних функцій, які є в меню. Для зручності вони реалізовані у вигляді значків, кожен з яких відповідає загальноприйнятим стандартам. Панель в Visual Studio 2010 плаваючого типу й складається з декількох частин, які за бажанням можна перетягувати, міняти місцями, додавати нові. Visual Studio при виборі виду проекту розміщує на панелі необхідні частини для зручної роботи з даним проектом.

5. Панель елементів. Викликається кнопкою на ПІ. Аналогічно ПІ і меню, панель елементів може містити в собі різний набір елементів, що відповідають типу відкритого проекту. Але загалом дане плаваюче вікно, яке можна переміщати, згортати і закріплювати на місці, містить у собі ряд об'єктів з стандартним кодом. Користувач може використовувати їх як шаблон, просто переміщуючи елемент на потрібне місце на формі; Visual

Studio автоматично згенерує код, що скорочує витрати часу на створення інтерфейсу користувача.

6. Оглядач рішень. Викликається кнопкою  на ПІ. Надає швидкий доступ до всіх файлів проекту в деревоподібному оформленні (як у провіднику). Тут можна безпосередньо відкривати їх, перейменовувати, видаляти. Містить і сам файл проекту, форми й каталоги проекту. Містить власну ПІ , у якій містяться основні команди для роботи з об'єктами проекту.

7. Вікно модуля. У Visual Studio може бути відкрито багато вікон модулів, в такому випадку вони відображаються у вигляді закладок. У даному випадку відкрито 1 вікно модуля. Воно може містити програмний код модуля (викликається кнопкою  у вікні оглядача рішень (6) для відповідного модуля), що представлений в зручному вигляді, з можливістю приховування розділів та функцією редагування й читання. Може бути у вигляді конструктора (викликається кнопкою  у вікні оглядача рішень (6) для відповідного модуля), як у нашому випадку. Конструктор надає можливість бачити реалізацію коду наочно, тобто у вигляді вже готового візуального об'єкта (форми).

8. Властивості. Викликається кнопкою  на ПІ. Відображає властивості виділеного в даний момент об'єкта (елемента управління, форми, тощо). Властивості надаються в згрупованому вигляді для зручності. Змінюючи поля у властивостях, Visual Studio автоматично змінить код, згідно з нововведеннями.

9. Довідка та вибраний елемент. Відображає стислу інформацію про вибраний об'єкт або його властивості, про значення слова в коді програми.

10. Рядок стану. Відображає інформацію про поточну подію у середовищі: відкриття файлу, збірка проекту, ініціалізація об'єктів. «Готово» означає, що Visual Studio виконала всі процеси й очікує дій вже з боку користувача.

Visual Studio включає підтримку мов програмування C# 4.0, Visual Basic .NET 10.0, а також мови F#, які забезпечують підтримку платформи .Net. Така різноманітність програмних засобів дозволяє обирати мову на основі особистих переваг, оскільки всі вони мають однакові можливості. Усі приклади, наведені в наступних главах, реалізовані на мові програмування Visual Basic 2010 .Net (далі VB).

VB призначений для ефективного створення типобезпечних і об'єктно-орієнтованих додатків (ООП). Програми, написані мовою VB, як і на інших мовах, призначених для Microsoft .NET Framework, відрізняються безпекою й підтримкою взаємодії. Це забезпечує можливість швидкого й ефективного створення додатків на основі .NET Framework.

Математичні обчислення у VB

Змінні

Змінна – це тимчасове місце в пам'яті для зберігання даних програми. У кодї може використовуватися одна чи декілька змінних, і вони можуть містити текст, числа, дати або властивості. Змінні корисні тим, що вони дозволяють присвоїти кожній частині даних, з якими ви працюєте, короткі імена. Змінні можуть зберігати інформацію, введену користувачем при виконанні програми, результат деякого обчислення або блок даних, який потрібно вивести у формі.

Якщо коротко, то змінні є зручними контейнерами, які ви можете використовувати для зберігання інформації довільного типу (табл. 4.1).

Таблиця 4.1 – Типи даних

Тип даних	Розмір	Діапазон значень
Integer (Ціле) Призначений для обробки цілочислових значень	4 байта	Від -2 147 483 648 до 2 147 483 647
Long (Довге ціле) Призначений для обробки цілочислових значень зі збільшеного діапазону	4 байта	Від -9,2E+18 до 9,2E+18
Single (Одинарної точності з плаваючою десятковою крапкою) Призначений для зберігання чисел в експонентній формі одинарної точності	4 байта	Від -3,402824E+38 до -1,401298E-45 для негативних значень; від 1,401298E-45 до 3,4028235E+38 для позитивних значень
Double (Подвійної точності з плаваючою десятковою крапкою) Призначений для зберігання чисел в експонентній формі подвійної точності	8 байт	Від -1,798E+308 до -4,9407E-324 для негативних значень; Від 4,9407E-324 до 1,7977E+308 для позитивних значень
String (Строковий) Призначений для зберігання строк різної довжини	залежить від платформи	До 2 мільярдів знаків в кодуванні Юнікод
Boolean (Логічний)	залежить від платформи	True / False
Date (Дата)	8 байт	Від 0:00:00 (опівночі) 1 січня 0001 року до 11:59:59 вечора 31 грудня 9999 року

Оголошення змінних

Використання змінних у програмах Visual Basic .NET вимагає деякого планування. Перш ніж ви зможете використовувати змінну, ви повинні виділити пам'ять, яку ця змінна буде використовувати.

Для оголошення змінної необхідно ввести її ім'я після оператора Dim (від англ. dimension – величина). Це оголошення при виконанні програми виділяє для змінної місце в пам'яті й дає можливість компілятору визначи-

ти тип даних, який він може зустріти надалі. Хоча це оголошення можна зробити в будь-якому місці коду програми (за умови, що воно з'являється до використання змінної), більшість програмістів оголошують змінні в одному місці на початку їх процедур обробки подій або модулів коду.

Наприклад, наступний оператор створює в програмі місце для змінної з ім'ям Змінна, яка буде зберігати текстове або рядкове значення.

```
Dim Переменная As String
```

```
Dim Переменная1, Переменная2,..., ПеременнаяN As Integer
```

Таблиця 4.2 – Приклади оголошення змінних

Тип даних	За допомогою типа даних
Ціле	Dim x As Integer
Дійсне одинарної точності	Dim x As Single
Дійсне подвійної точності з плаваючою десятковою комою	Dim x As Double

Область видимості змінних

При виконанні програми принципове значення має область видимості використовуваних змінних. Спроба використання змінних, які не діють в даному місці програми, призводить до помилки програми або до неоднозначності результатів. У Visual Basic можуть застосовуватися глобальні та локальні змінні. Глобальні змінні доступні з будь-якої частини програми. Для локальних змінних можна задавати область видимості в рамках усього модуля або окремої процедури.

Для оголошення змінної, локальної всередині модуля або форми, використовуються оператори Private або Dim в розділі [Общие / Объявления] (General / Declaration) модуля або форми. У цьому випадку оголошена змінна буде доступна для всіх процедур, які входять у форму або модуль, але в той же час виявиться недоступною в процедурах інших модулів і форм.

Змінні, локальні на рівні процедури, створюються операторами Dim або Static всередині процедури.

```
Sub Процедура
    Dim локПеременная As Integer
End Sub
```

Для створення глобальної змінної, в розділі [Общие / Объявления] (General / Declaration) головного модуля програми необхідно включити оператор Public.

```
Public глПеременная As Date
```

Присвоєння значення змінним

Після того, як змінна створена, можна задати її значення, використовуючи оператор присвоєння (=).

Наприклад, наступний оператор програми присвоює змінній < Имя > значення "Джон":

```
Имя = "Джон"
```

Присвоїти значення змінної можливо при її оголошенні:

```
Dim X As Integer = 4
```

Константи

Якщо в програмі використовується змінна, значення якої не змінюється, то краще використовувати замість змінної константу. Константа являє собою умовне ім'я, що використовується замість числа, або текстового рядка, що не підлягає зміні. Дія константи схожа з дією змінної, але її значення не може бути змінено в процесі використання програми. Константи оголошуються за допомогою ключового слова `Const`.

```
Public/ Private Const < ім'я > [ AS < ім'я типу > ] = < значення >
Const Pi=3.14159
```

Коментарі

Крім команд і виразів можна включити в методи й процедури довільний текст або коментарі. Коментарі, що пояснюють текст програми, зроблять її більш читабельною й допомагають краще орієнтуватися в програмному коді.

Для включення в текст програми коментаря необхідно ввести символ ['], який може бути першим символом рядку або перебувати в будь-якому її місці. Цей символ означає початок коментаря. Будь-який текст, розташований у рядку слідом за цим символом, буде сприйматися як коментар, а не як програмний код для трансляції.

```
Imports System.Math 'импорт пространства имен System.Math
```

Розміщення оператора на декількох рядках

У випадку, якщо оператор має велику довжину, його можна розбити на кілька рядків, використовуючи символи продовження рядка, що представляють собою пробіл, за яким слідує символ підкреслення [_].

Розміщення декількох операторів на одному рядку

Як правило, при написанні програм оператори розміщують на окремому рядку. Якщо оператори мають невелику довжину, можна їх помістити на одному рядку, розділивши двокрапкою [:].

Основні варіанти розміщення операторів подано у табл. 4.3.

Таблиця 4.3 – Приклади розміщення операторів

a = 2 b = 5 c = 10	Оператори на окремому рядку
a = 2 : b = 5 : c = 10	Оператори в один рядок
MsgBox("x =" & Str(x) & Chr(13) & "cos(x) =" & Str(Cos(x)), _ vbInformation + vbOKCancel, "Результат")	Конструкція розділена знаком переносу строки [_]

Математичні оператори. Функції

У табл. 4.4 наведені умовні позначення шести базових математичних операцій.

Результат стандартного ділення (/) завжди відноситься до типу Double, навіть у разі ділення без залишку. Результат цілочисельного ділення (\) завжди відноситься до типу Integer.

Таблиця 4.4 – Арифметичні операції

Оператор	Операція
+	Додавання
-	Віднімання та позначення негативних чисел
/	Ділення
\	Цілочисельне ділення
*	Множення
^	Піднесення до степені

Стандартні типи функцій

Функція – це оператор, що виконує певні дії та повертає результат роботи у програму. Функція може мати один або кілька аргументів, які беруться в дужки й відокремлюються між собою комами.

Круглі дужки та пріоритет операцій

При обробці складних виразів послідовність виконання операцій задається двома способами. При використанні круглих дужок вам не доведеться запам'ятовувати пріоритети різних операцій (табл. 4.5). У VB, NET, як і в багатьох мовах програмування, операції мають пріоритет, який ви-

значає послідовність їх виконання. Множення володіє більш високим пріоритетом, ніж складання; отже, вираз $3+4*5$ дорівнює 23, оскільки множення ($4*5$) виконується раніше, ніж складання.

Нижче перераховані математичні операції в порядку зменшення пріоритету.

Таблиця 4.5 – Арифметичні операції за пріоритетом

Оператори	Порядок обчислення
()	Першими завжди обчислюються значення в круглих дужках
^	Піднесення числа до степеня обчислюється в другу чергу
-	Створення негативного числа (зміна знака) йде третім
* /	Четвертими йдуть множення і ділення
\	П'яте – цілочисельне ділення
Mod	Шосте – залишок від ділення
+ -	Останні – додавання і віднімання

Якщо дві операції мають однаковий пріоритет, порядок виконання визначається порядком їх слідування у виразі (зліва направо).

Математичні функції та математичні константи

Математичні функції та математичні константи в VB. NET реалізовані у вигляді методу класу Math, що входить в .NET Framework.

Всі перераховані методи є загальними методами класу Math, тому вони повинні викликатися з префіксом Math – наприклад: `Math.Log10 (10)`, або імпортувати клас Math у модуль: `Imports System.Math`.

Таблиця 4.6 – Основні математичні функції

Математична функція	Опис
Abs(arg)	Повертає абсолютне значення (модуль) числа (arg)
Acos(arg)	Повертає кут, косинус якого дорівнює заданому числу (arg)
Asin(arg)	Повертає кут, синус якого дорівнює заданому числу (arg)
Atan(arg)	Повертає кут, тангенс якого дорівнює заданому числу (arg)
Cos(arg)	Повертає косинус заданого кута (arg)
Exp(arg)	Повертає число e ($\approx 2,71828182845905$), зведене в задану ступінь (arg)
Log(arg)	Повертає натуральний логарифм числа (arg)
Log10(arg)	Повертає десятковий логарифм числа (arg)
Log(arg, осн)	Повертає логарифм числа (arg) за заданою основою (осн)
Round(arg[, точн])	Округляє число (arg) до найближчого цілого або зазначеної кількості десятих знаків після коми (точн)
Sign(arg)	Повертає величину, що визначає знак числа (arg)
Sin(arg)	Повертає синус заданого кута (arg)
Sqrt(arg)	Повертає квадратний корінь числа (arg)
Tan(arg)	Повертає тангенс заданого кута (arg)

Функції конвертації

Таблиця 4.7 – Основні функції конвертації

Ім'я функції	Тип, що повертається	Діапазон для аргументу
CDbl	Double	Від $-1.79769313486231570E+308$ до $-4.94065645841246544E-324$ для негативних значень; від $4.94065645841246544E-324$ до $1.79769313486231570E+308$ для позитивних значень
CInt	Integer	Від -2147483648 до 2147483647 ; дробова частина округляється
CLng	Long	Від $-9,223,372,036,854,775,808$ до $9,223,372,036,854,775,807$; дробова частина округляється
CShort	Short	Від -32768 до 32767 ; дробова частина округляється
CSng	Single	Від $-3,402823E+38$ до $-1,401298E-45$ для негативних значень Від $1,401298E-45$ до $3,402823E+38$ для позитивних значень
CStr	String	Значення що повертаються функцією CStr залежать від аргументу
Val	Числовий	Повертаються числа, що містяться в строковій змінній в якості числових значень відповідного типу

Функції вводу/виводу

Для реалізації більш гнучкого інтерфейсу користувача доцільно замість жорсткого присвоювання значень змінним або об'єктам використовувати функцію вводу InputBox. Для виводу результатів або повідомлень програми, можливе використання функції виводу MsgBox.

Оператор вводу InputBox

Синтаксис:

InputBox (<повідомлення> , [<заголовок>] , [<значення>])

<повідомлення> / [<заголовок>] Обов'язковий / необов'язковий параметр типу String. довільна послідовність символів в подвійних лапках ("); текст, який міститься в згенерованому вікні InputBox в якості пояснення та до заголовку відповідно. Якщо текст складається з декількох рядків, то можна розділити рядки за допомогою знаку повернення каретки (Chr(13)), знаку переходу на новий рядок (Chr(10)) або їх комбінації між кожним рядком;

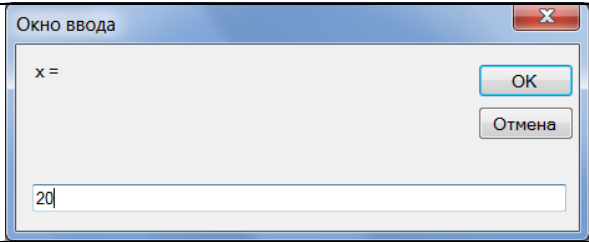
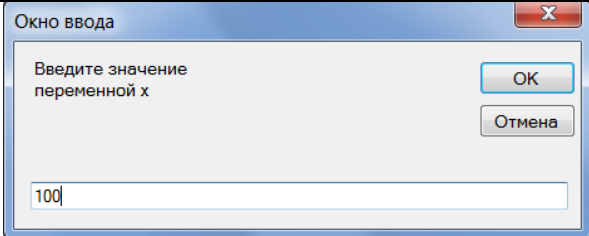
<значення> значення за замовчуванням змінної відповідного типу.

Якщо будь-який необов'язковий параметр оператора замовчується, ставляться коми.

Приклад (результат виконання подано на рис. 4.1):

```
x = InputBox("x = ", "Окно вводу", 20, 100, 500)
```

Таблиця 4.1 – Приклади використання функції введення InputBox

<pre>x = InputBox("x = ", "Окно ввода", 20)</pre>	
<pre>x = InputBox("Введите значение" & Chr(13) & "переменной x", "Окно ввода", 100)</pre>	

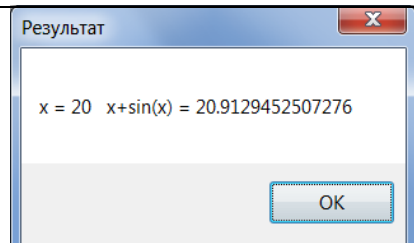
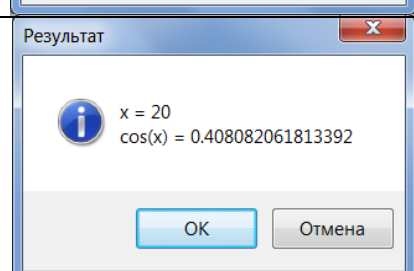
Оператор виведення даних MsgBox

Синтаксис:

`MsgBox("<СПИСОК ВИВОДУ>" [, <КНОПКИ>] [, "<ЗАГОЛОВOK>"])`

- <СПИСОК ВИВОДУ>** Обов'язковий параметр типу String. Текст, що відображається в діалоговому вікні у вигляді повідомлення. Якщо текст складається з декількох рядків, то можна розділити рядки за допомогою знаку повернення каретки (Chr(13)), знаку переходу на новий рядок (Chr(10)) або їх комбінації між кожним рядком. В якості списку виводу також зручно використовувати функцію Str(аргумент). Елементи списку виводу з'єднуються між собою знаком &;
- [<КНОПКИ>]** Необов'язковий параметр. Значення, що задають номер і тип відображуваних кнопок, стиль використовуваного значка, тип кнопки за замовчуванням і ознаку модальності вікна повідомлення. Якщо даний параметр опущений, по-замовчуванню використовується нульове значення;
- [<ЗАГОЛОВOK>]** Необов'язковий параметр типу String. Текст, який відображається в рядку заголовка діалогового вікна. Якщо параметр опущений, рядок заголовка містить ім'я додатку.

Таблиця 4.2 – Приклади використання функції виведення MsgBox

<pre>MsgBox("x =" & Str(x) & " " & "x+sin(x) =" & Str(x + Sin(x)), , "Результат")</pre>	
<pre>MsgBox("x =" & Str(x) & Chr(13) & "cos(x) =" & Str(Cos(x)), vbInformation + vbOKCancel, "Результат")</pre>	

Розв'язок задачі «Обчислення значення математичного виразу»

Умова. Обчислити значення виразу, з розробкою інтерфейсу користувача:

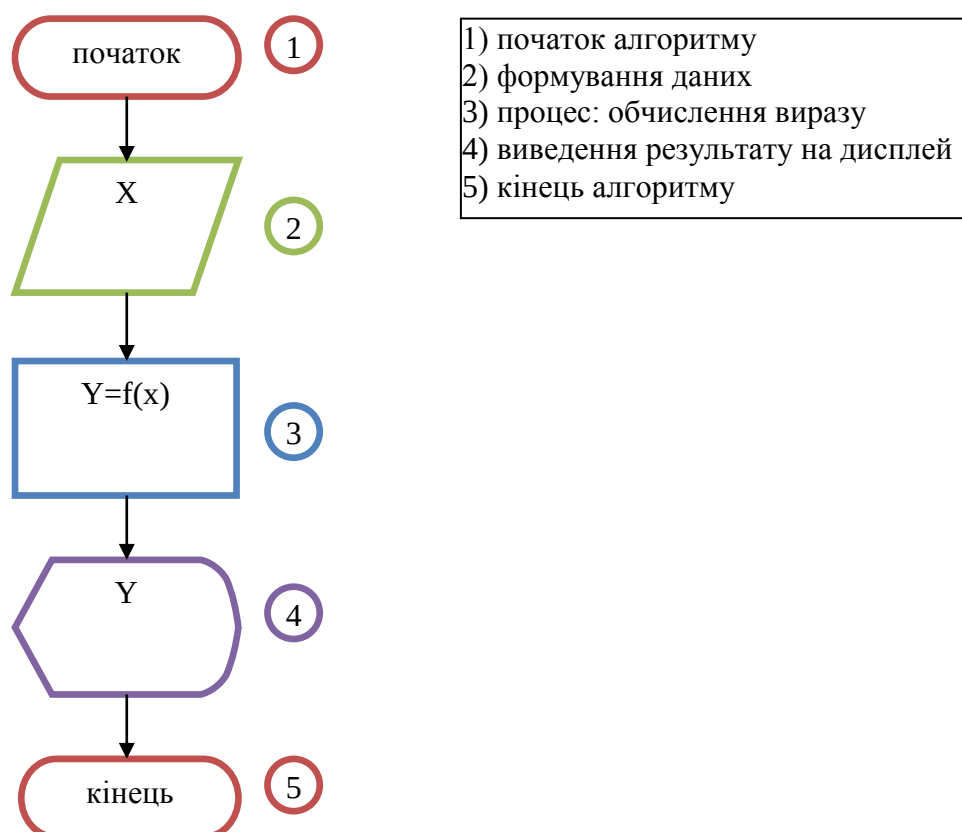
$$\left(\sin(x) - 1 + \frac{1}{\sin^2(3x^5 + 67x^2 - 6x)} \right) * \left(\frac{x^2 - x}{(2 - x)^2} \right) \quad \text{при } x = 0,5$$

Отримаємо еквівалентну математичну конструкцію згідно з даною умовою:

$$y = (\sin(x) - 1 + 1 / ((\sin(3 * x ^ 5 + 67 * x ^ 2 - 6 * x)) ^ 2)) * ((x ^ 2 - x) / ((2 - x) ^ 2))$$

Стислі відомості про елементи управління, які призначенні для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



$$f(x) = (\sin(x) - 1 + 1 / ((\sin(3 * x ^ 5 + 67 * x ^ 2 - 6 * x)) ^ 2)) * ((x ^ 2 - x) / ((2 - x) ^ 2))$$

Програмний код алгоритму

Імпортуємо системний клас Math, який служить для використання математичних функцій і математичних констант:

```
Imports System.Math ' импорт пространства имен System.Math
Public Class Form1
```

Оголошуємо змінні:

```
Dim y As Double, x As Double
Private Sub Form3_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    Label1.Text = "Вычислить, с разработкой соответствующего интерфейса пользователя: "


Label1.BackColor = Color.Gray



Label1.ForeColor = Color.WhiteSmoke



Label1.Font = New System.Drawing.Font("Arial", 10)



End Sub



Private Sub NumericUpDown1_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles NumericUpDown1.Click


```

У разі помилки програма одразу перейде до позначки 999

```
On Error GoTo 999
x = TextBox1.Text
```

В наступному блоці обчислимо вираз, згідно з умовою:

```
y = (Sin(x) - 1 + 1 / ((Sin(3 * x ^ 5 + 67 * x ^ 2 -
6 * x)) ^ 2)) * ((x ^ 2 - x) / ((2 - x) ^ 2))
TextBox2.Text = y
```

Використання математичної функції Round, яка округлює значення змінної y до встановленої точності:

```
TextBox2.Text = Round(y, CInt(NumericUpDown1.Value))
999:
End Sub
End Class
```

Тестування алгоритму

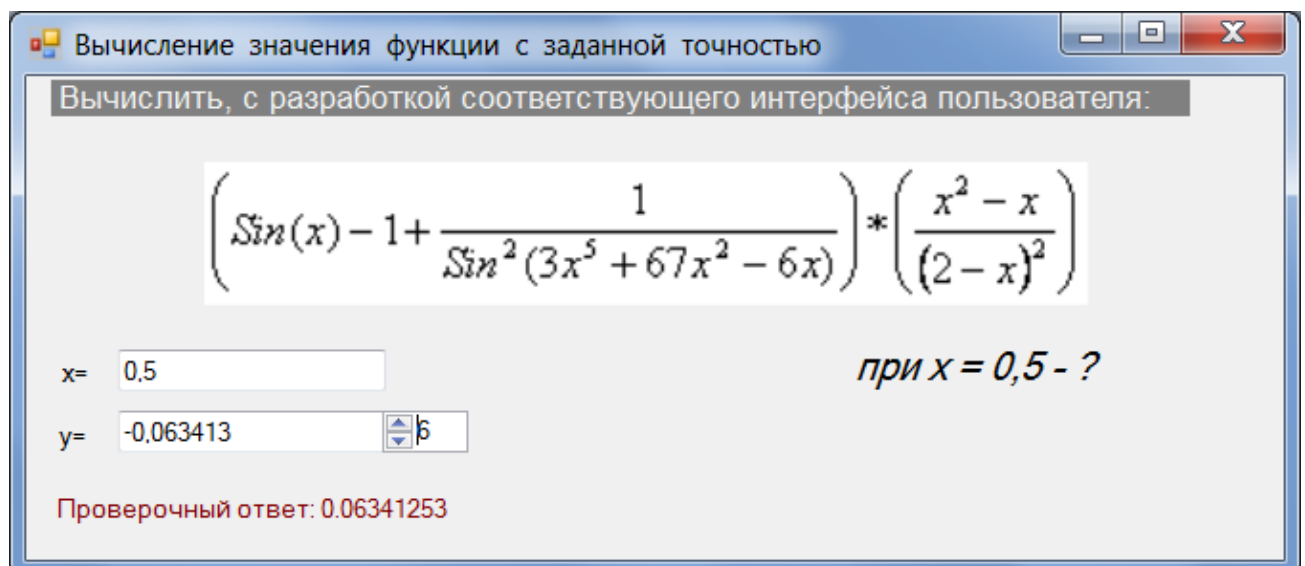


Рисунок 4.1 – Форма задачі «Обчислення виразу»

4.2 Практичне завдання

Реалізувати лінійний алгоритм, побудувати блок-схему, виконати тестування при 2 значення

1.	$\left(\sin(x) - 1 + \frac{1}{\sin^2(3x^5 + 67x^2 - 6x)} \right) * \left(\frac{x^2 - x}{(2 - x)^2} \right)$ При $x = 0,5$. Відповідь: -6,341253E-02.
2.	$\left(\frac{b}{ab - 2a^2} - \frac{2}{b^2 + b - 2ab - 2a} * \left(1 + \frac{3b + b^2}{3 + b} \right) \right) + {}^{\log_5 9} \sqrt[9]{81}$ При $a=0,1$; $b=7$ Відповідь: 35.
3.	$2 \log_4 x + 5 \log_x 4 - 11, \text{ при } x = 2$ Відповідь: 0.
4.	$\left(\sqrt{2 \operatorname{tg} x - \frac{1}{x}} + 2 \right) * (5x^2 - 14x + 8) + \frac{45}{x^3}$ При $x = \frac{4}{5}$ Відповідь: 87,89063.
5.	$ x - 10 \log_2(x - 4) + 2(x - 10) + \sqrt[5]{x^4}, \text{ при } x = 3$ Відповідь: 2,408225.
6.	$\log_{\frac{2}{3}} \left(\frac{5}{2} * 3^x - 9 * 2^{x-1} \right) - \log_{\frac{2}{3}} (3^x - 2^{x-1}) + x - \frac{45}{\left(\frac{4}{5} \right)^3}, \text{ при } x = 2,70951129135145$ Відповідь: 87,89063.
7.	$\log_4(x^2 - 4x + 2) - \log_4(x^3 - 6x + 5), \text{ при } x = 1 - \sqrt{2}$ Відповідь: -0,5
8.	$\log_2(x^2 - x - 4) - \log_2(x^2 - 3x - 2) + 1 + \sqrt{\left(\frac{69^5}{\sqrt[4]{15375}} \right)^6}, \text{ при } x = -3$ Відповідь: 9629575,61867063.
9.	$\log_2 \sqrt{2x} + 2 \log_8 x - \log_{16} x - \frac{23}{12} + \sqrt[27]{x^{x^9}}, \text{ при } x = 2$ Відповідь: 510999,715707581.
10.	$\log_3(49^{\log_7 2} - (0,9)^0) + \sqrt{5^{(\sqrt{3}-1)^2 * 25^{\sqrt{3}}}}$ Відповідь: 26.
11.	$\left(\frac{1}{4} \right)^{9 \log_4 \frac{3-\sqrt{5}}{2} + 4 \log_{\frac{3}{2}}(5+3\sqrt{5})} + \sqrt{5} + \sqrt{6\sqrt{14} - 6\sqrt{5} - 4}$ Відповідь: 3,000125
12.	$4^{3 \log_{2\sqrt{2}}(5-\sqrt{10}) - 4 \log_4(\sqrt{5}-\sqrt{2})} + \sqrt[3]{9 + \sqrt{80}} + \sqrt[3]{9 - \sqrt{80}}$

	Відповідь: 28
13.	$3(\sqrt{7}-1)^2 : (\sqrt{7}+4)^{-1} + 3^{\log_{\sqrt{3}} 4} - \log_{\sqrt{1/2}} 2$ Відповідь: 71.
14.	$\log_{1/3}(2x+3) + 4 - \log_{1/3}(5-4x) - 7^{\log_7 5}$, при $x = -\frac{5}{7}$ Відповідь: 0.
15.	$\sqrt{\frac{1}{2}(8+\sqrt{15})} \left(\frac{1}{2}(8-\sqrt{15}) \right)^{\frac{1}{2}} - \log_2 \sqrt{2} - 3^{\log_{\sqrt{3}} 2}$ Відповідь: -1.

5 ЛАБОРАТОРНА РОБОТА №5

5.1 Теоретичні відомості

Умовний оператор If...Then...Else

Умовний оператор використовується для подання розгалуженого обчислювального процесу у кодї програми мовою VB.Net. Детальна інформація про алгоритми розгалуження подана у додатку А (гл. А.1).

Умовний оператор If...Then...Else має два типи структур: рядкову та блочну.

Синтаксис рядкової структури умовного оператора:

```
If <логічний вираз> Then <оператори1> [Else <оператори2>]
```

логічний вираз результатом може бути одне з двох значень (істина) або (хибність); логічний вираз складається з операндів логічного типу, між якими містяться знаки логічних операцій і може мати одне з двох значень: True (Істина) або False (Хибність).

оператори 1 виконуються при значенні логічного виразу «істина».

оператори 2 виконуються при значенні логічного виразу «хибність».

У табл. 5.1 наведено список операторів порівняння й умов.

Таблиця 5.1 – Логічні оператори та результати

Оператор	Істина, якщо	Хибність, якщо
< (менше)	expression1 < expression2	expression1 >= expression2
<= (менше або рівно)	expression1 <= expression2	expression1 > expression2
> (більше)	expression1 > expression2	expression1 <= expression2
>= (більше або рівно)	expression1 >= expression2	expression1 < expression2
= (рівно)	expression1 = expression2	expression1 <> expression2
<> (не рівно)	expression1 <> expression2	expression1 = expression2

Синтаксис блочної форми умовного оператора:

```
If <логічний вираз 1> Then
    оператори 1
[Else if <логічний вираз 2> Then
    оператори 2]
[Else
    оператори n]
End If
```

оператори 1 довільна кількість операторів, що виконуються за умовою:

логічний вираз 1 є «істина»;

логічний вираз 2 повертає ненульове значення («істина») або нуль («хибність»);

оператори 2 довільна кількість операторів, що виконуються за умовою:

логічний вираз 2 є «істина»

оператори n довільна кількість операторів, що виконуються при інших умовах.

В наступному прикладі реалізований алгоритм визначення кількості цифр числа. Критерієм визначення є належність числа до одиниць, десятків, сотень (1 цифра, 2 цифри, 3 цифри відповідно).

```
Dim число, цифри As Integer
Dim строка As String
число = 53 ' задаем число
'в логической конструкции проверяем принадлежность числа к
одному из интервалов
If число < 10 Then
    цифри = 1 'если число меньше 10 - оно содержит 1 цифру
ElseIf число < 100 Then
    цифри = 2 'если число меньше 100 (и при этом больше
10, исходя из предыдущего условия) - оно содержит 2 цифры
Else
    цифри = 3 'если число больше, или равно 100 (следует
из предыдущих 2 условий) - оно содержит 3 цифры
End If
If цифри = 1 Then строка = "Одна цифра" Else строка = "Не-
сколько цифр"
MsgBox(строка, MsgBoxStyle.Information, "Сообщение")
```

Логічна конструкція Select Case

Конструкція Select Case дозволяє обробляти в програмі декілька умов і аналогічна блоку конструкцій If ... Then ... Else. Ця конструкція складається з аналізованого виразу й набору операторів Case на кожне можливе значення виразу.

Синтаксис конструкції Select Case наступний:

```
Select Case <змінна>
    CASE <значення 1>
        <конструкція 1>
    CASE <значення 2>
        <конструкція 2>
    ...
End Select
```

змінна
значення 1, **значення 2**,...

<конструкція 1> / <конструкція 2>

змінна для порівняння;
обов'язковий в операторі Case. Список можливих значень для параметра <змінна>, що розділяються комами, який може мати вигляд (табл. 5.2);
Один або група операторів, що реалізуються, якщо значенню <змінна> відповідає **значення 1** / **значення 2**.

Таблиця 5.2 – Конструкції для оператора Case

значення1 То значення2	вказує діапазон значень для <змінна>
Is <оператор порівняння> <значення>	Використовується з оператором порівняння (=, <, <=, > або >=) для обмеження можливих значень <змінна>
значення	Вид, у якому надається тільки значення, розглядається як особливий випадок використання Is, де оператор порівняння – знак рівності (=). Дана форма обчислюється як змінна = значення

Логіка виконання конструкції наступна: обчислюється значення заданого в конструкції виразу. Потім отримане значення порівнюється зі значеннями, що задаються в операторах CASE конструкції. Якщо знайдено шукане значення, виконуються команди, приписані даному оператору CASE. Після завершення виконання конструкцій управління буде передано конструкції, наступної за ключовим словом End Select.

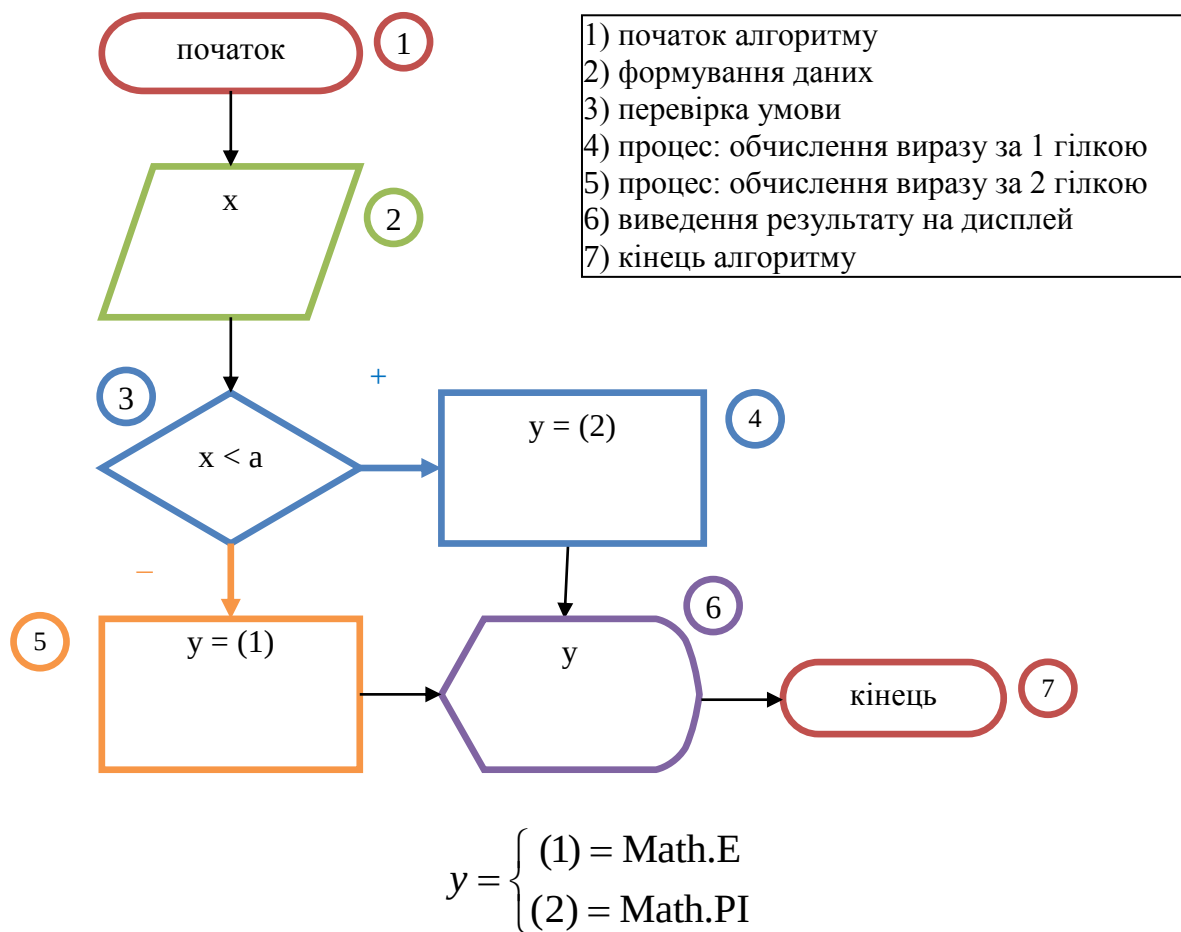
Розв’язок задачі «Алгоритм розгалуження 2 гілки» (If-рядок)

Умова. Обчислити значення кусково-аналітичної функції 2 гілки з використанням конструкції (If-рядок).

$$y = \begin{cases} e, x < a \\ \pi, x \geq a \end{cases}$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```

Dim x, y, v, a
Dim qqqr As Label
Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click

```

Якщо буде обраний RadioButton1, то значення аргументів будуть вводитися через текстові поля:

```

If RadioButton1.Checked = True Then
    a = CDb1 (TextBox1.Text)
    x = CDb1 (TextBox2.Text)

```

```
Else
```

Якщо буде обраний RadioButton2, то значення аргументів будуть вводитися через діалогові вікна, за допомогою оператора вводу InputBox:

```

a = InputBox ("Введіть значення узла a")
x = InputBox ("Введіть значення аргумента x")

```

```
End If
```

В наступному блоці реалізується алгоритм вибору гілки згідно з умовою розгалуження, представлений у вигляді строкової конструкції If...Then:


```

If x < a Then y = Math.E : v = 1 Else y = Math.PI : v = 2
MsgBox ("ОТВЕТ:  x=" & x & ";  y=" & y & Chr(13) & "Вычи-
сления прошли по  " & v & "-й ветви ", , "Ветвление: 2 ветви")
End Sub

```

Тестування алгоритму

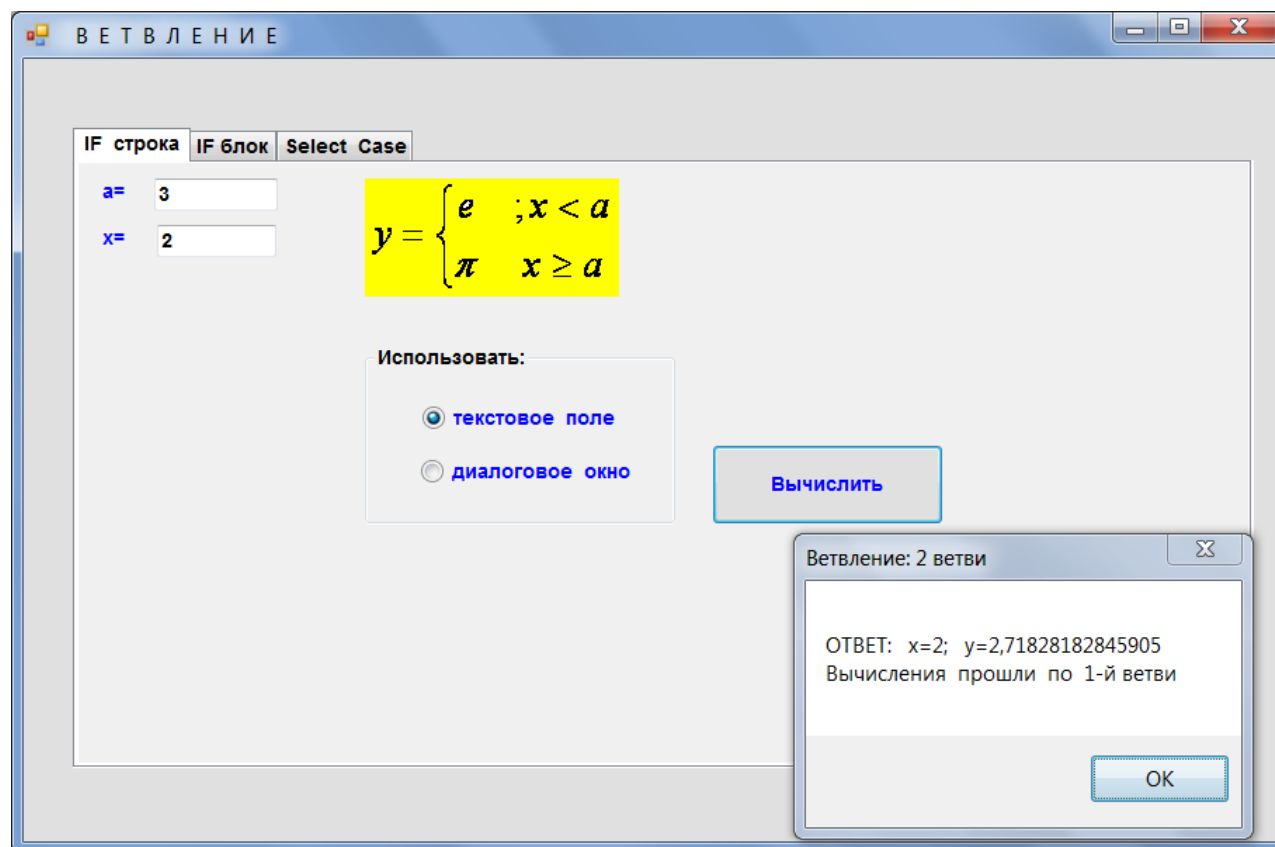


Рисунок 5.1 – Форма задачі «Алгоритм розгалуження 2 гілки»

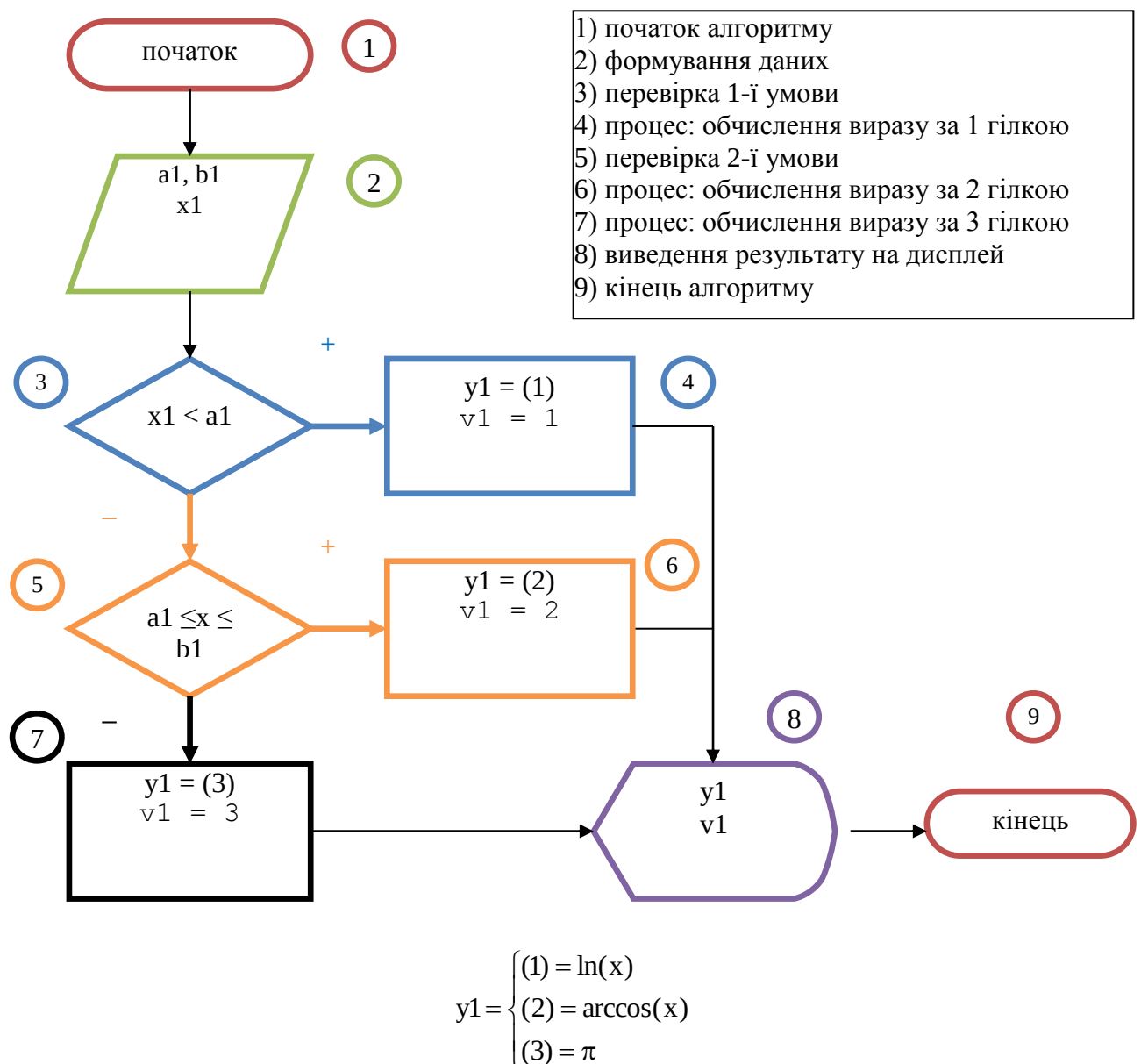
Розв'язок задачі «Алгоритм розгалуження 3 гілки» (If-блок)

Умова. Обчислити значення кусково-аналітичної функції 3 гілки з використанням конструкції (If-блок)

$$Y = \begin{cases} \ln(x), & x < a \\ \arccos(x), & a \leq x \leq b \\ \pi, & x \geq b \end{cases}$$

Стислі відомості про елементи управління, які призначенні для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```
Dim a1 As Double = -6
Dim b1 As Double = 5
Dim x1 As Double = 22
Dim Y1 As Double
Dim v1 As Double
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    ListBox1.Items.Clear()
    On Error GoTo 999 'в случае ошибки управление передается метке 999
    ComboBox1.Items.Add(ComboBox1.Text) 'Если надо добавить новое значение
```

```

a1 = ComboBox1.Text 'Задание значения узла А
ComboBox2.Items.Add(ComboBox2.Text) 'Если надо добавить
новое значение
b1 = ComboBox2.Text 'Задание значения узла В
ComboBox3.Items.Add(ComboBox3.Text) 'Если надо добавить
новое значение
x1 = ComboBox3.Text 'Задание значения узла С

```

В наступному блоці реалізується алгоритм вибору гілки згідно з умовою розгалуження, представлений у вигляді блокової конструкції If...Then:

```

If x1 < a1 Then
    Y1 = Log(x1) : v1 = 1
ElseIf x1 >= a1 And x1 <= b1 Then
    Y1 = Acos(x1) : v1 = 2
Else
    Y1 = PI : v1 = 3
End If

'Вывод результата в ListBox1
ListBox1.Items.Add("Получили значение функции:")
ListBox1.Items.Add("Y=" & Y1)
ListBox1.Items.Add("при значении аргумента:")
ListBox1.Items.Add("x=" & x1)
ListBox1.Items.Add("Вычисления прошли по " & v1 & "-й
ветви") : GoTo 1000
999:    ListBox1.Items.Add("Проверь правильность ")
        ListBox1.Items.Add("ввода данных")
1000:
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button4.Click
    'Запускающая кнопка
    If CheckBox1.Checked = True Then
        Button3.Visible = True
        Button4.Visible = False
        'Layout (пер с англ. - схема размещения)
        TableLayoutPanel1.Visible = True
    End If
    '***** IF 3 ветви (Стандартный вариант) *****

```

В наступному блоці реалізується алгоритм вибору гілки згідно з умовою розгалуження, представлений у вигляді блокової конструкції If...Then:

```

If x1 < a1 Then
    Y1 = Log(x1) : v1 = 1
ElseIf x1 >= a1 And x1 <= b1 Then
    Y1 = Acos(x1) : v1 = 2
Else
    Y1 = PI : v1 = 3
End If

Dim Ответ As Object = MsgBox("ОТВЕТ:  x=" & x1 & ";  y="
& Y1 & Chr(13) & _
    "Вычисления прошли по " & v1 & "-й ветви",
MsgBoxStyle.Exclamation + MsgBoxStyle.YesNo)
If Ответ = MsgBoxResult.No Then

```

```

Dim Ответ1 As Object = MsgBox("ПРОДОЛЖАТЬ РАБОТУ ?",
MsgBoxStyle.Question + MsgBoxStyle.YesNo)
If Ответ1 = MsgBoxResult.No Then
    TabControl1.SelectTab(2)
End If
End If
End Sub

```

Тестування алгоритму

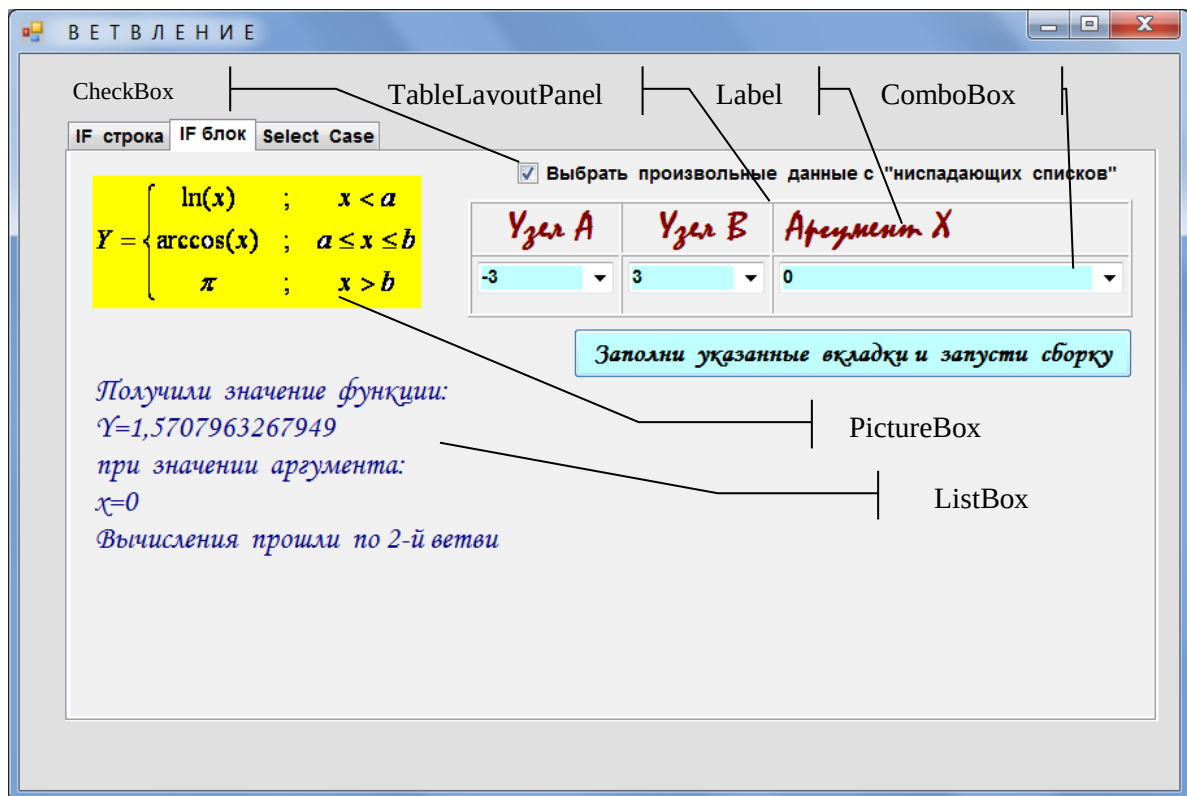


Рисунок 5.2 – Форма задачі «Алгоритм розгалуження 3 гілки»

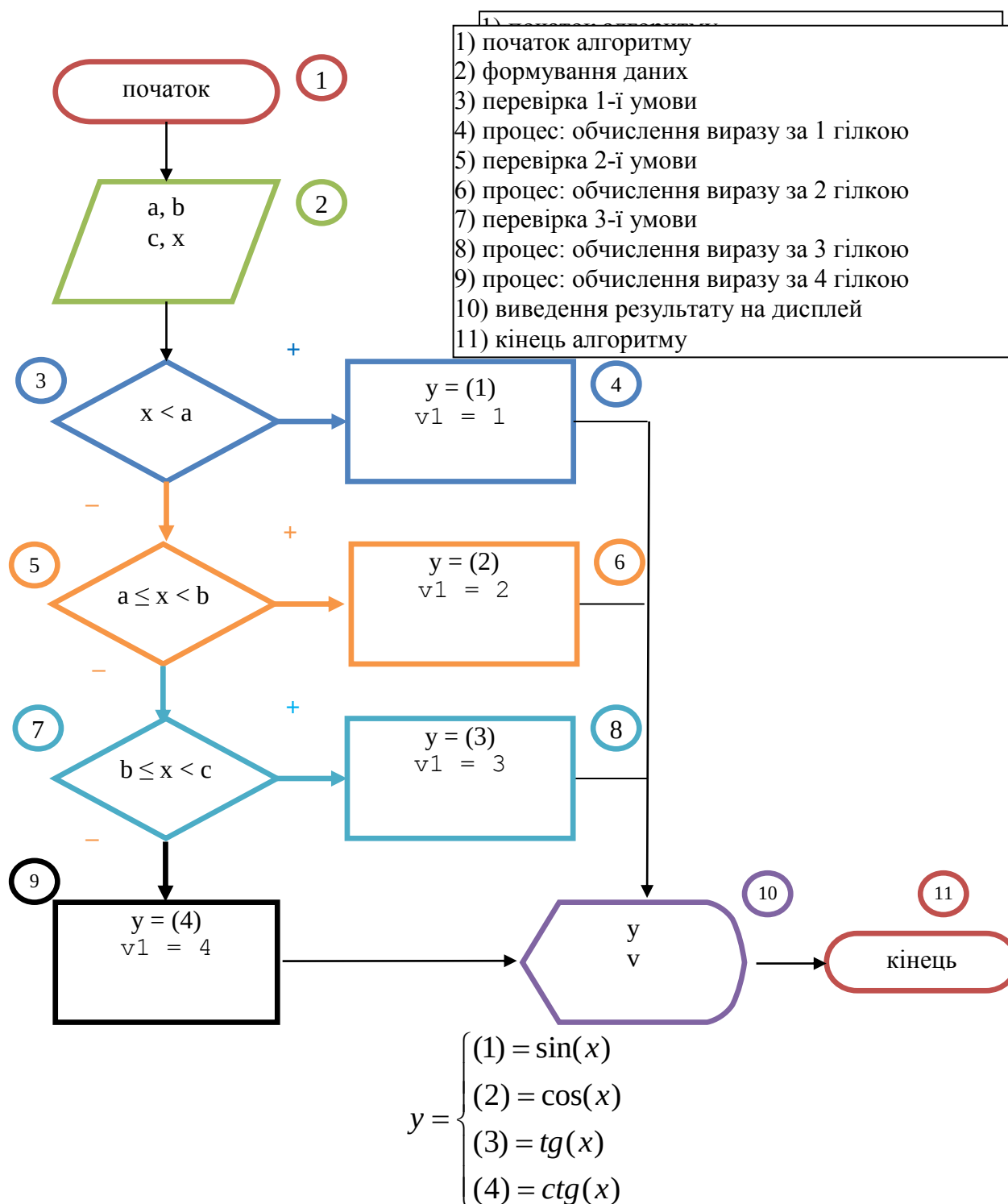
3.5 Розв'язок задачі «Алгоритм розгалуження 4 гілки» (Select Case)

Умова. Обчислити значення кусково-аналітичної функції 4 гілки з використанням конструкції (Select Case).

$$y = \begin{cases} \sin(x), & x < a \\ \cos(x), & a \leq x < b \\ tg(x), & b \leq x < c \\ ctg(x), & x \geq c \end{cases}$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Label10.Visible = False : Label11.Visible = False
    Label12.Visible = False : Label13.Visible = False
    Dim a As Double, b As Double, c As Double, v As Integer, x As Double
    a = TextBox3.Text : b = TextBox4.Text : c = TextBox5.Text
```

```

x = InputBox("Введите значение аргумента", "Окно вво-
да", -5)
Select Case x
    Case Is < a
        y = Sin(x) : v = 1 : Label10.Text = "y=" &
Round(y, 5) : Label10.Visible = True : qq = Label10
    Case a To b And x <> b
        y = Cos(x) : v = 2 : Label11.Text = "y=" &
Round(y, 3) : Label11.Visible = True : qq = Label11
    Case b To c And x <> c
        y = Tan(x) : v = 3 : Label12.Text = "y=" &
Round(y, 3) : Label12.Visible = True : qq = Label12
    Case Else
        y = 1 / Tan(x) : v = 4 : Label13.Text = "y=" &
Round(y, 5) : Label13.Visible = True : qq = Label13
End Select
End Sub
#Region "МЕПЦАЮЩЕЕ ПОЛЕ"
Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Timer1.Interval = 100
    Timer1.Enabled = True
    Timer2.Interval = 150
    Timer2.Enabled = True
    qq = Label14
    '-----
    Button3.Visible = False
    'Layout-ПЕРЕВОД:Схема размещения
    TableLayoutPanel1.Visible = False
End Sub
Sub Timer1_Tick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    With qq
        .BackColor = Color.Blue
        .ForeColor = Color.Aqua
        .Font = New System.Drawing.Font("Arial", 8)
    End With
End Sub
Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
    With qq
        .BackColor = Color.Aqua
        .ForeColor = Color.Blue
        .Font = New System.Drawing.Font("Arial Black", 8)
    End With
End Sub
#End Region

```

Тестування алгоритму

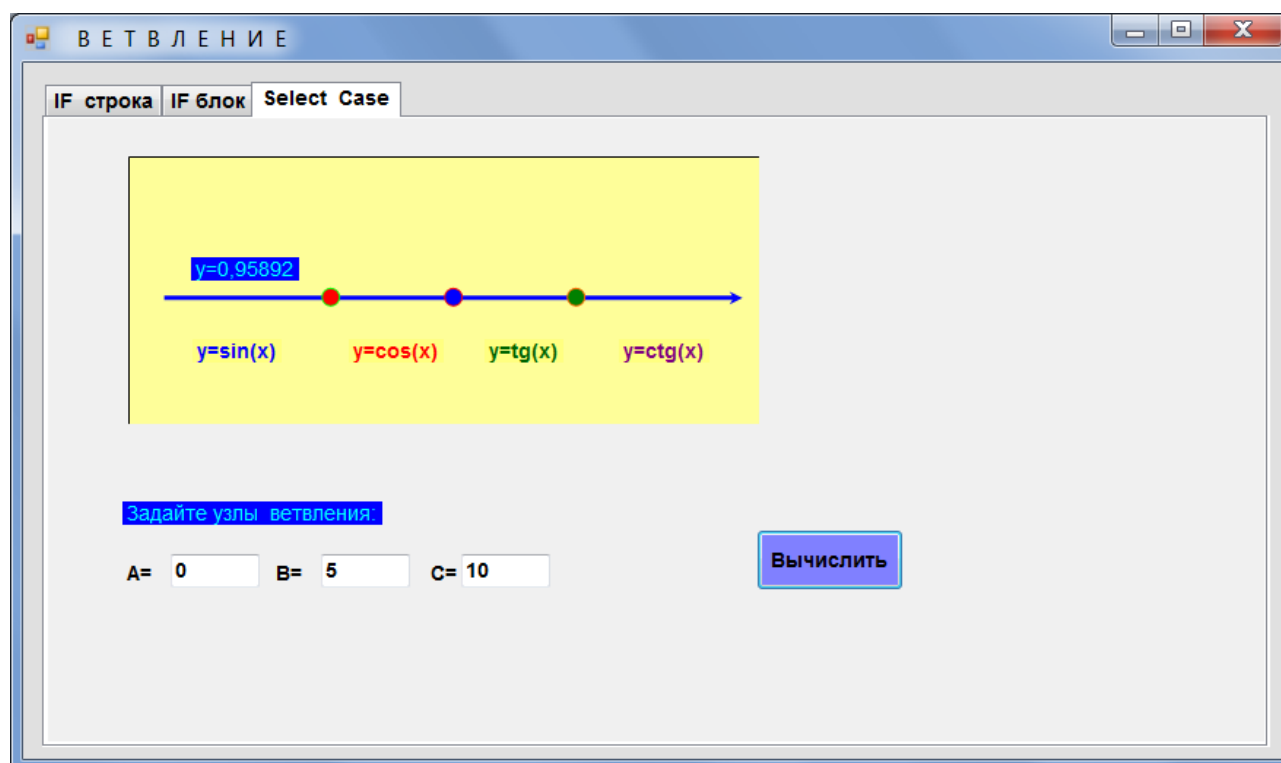


Рисунок 5.3 – Форма задачі «Алгоритм розгалуження 4 гілки»

5.2 Практичне завдання

Таблиця 5.3 – Варіанти індивідуальних завдань

1	Увести число. Якщо воно позитивне – відняти з нього 50; якщо негативне – помножити на 10, якщо нуль – додати 100.
2	Увести 2 числа. Відняти з більшого менше. Якщо числа рівні – скласти їх.
3	Увести число. Якщо воно парне – розділити його на 4, якщо при цьому число рівне 0 – додати 100; якщо непарне – помножити на 5.
4	Скласти алгоритм, що забезпечує введення трьох значень температури та перевіряючий, чи є серед них температура танення льоду
5	Скласти алгоритм, що забезпечує введення температури тіла людини та визначає, здоровий він або хворий (ідеальний стан: 36,6, здоровий: $36 < t < 37$, хворий: >37)
6	Увести 2 числа. Якщо їх добуток негативний, помножити його на -2 і вивести на екран, а якщо ні, то збільшити його в 1,5 рази та вивести на екран.
7	Скласти програму, яка запитує введення трьох значень температури та перевіряє, чи є серед них температура танення льоду
8	Дан вираз: $y = \frac{1}{\sqrt{x}}$ Скласти алгоритм, що забезпечує визначення допустимості введеного

	значення x і обчислення виразу при припустимому значенні .
9	Дан вираз: $y = \frac{1}{\ln(x)}$ Скласти алгоритм, що забезпечує визначення допустимості введеного значення x і обчислення виразу при припустимому значенні.
10	Обчислити значення кусочно-аналітичної функції 3 гілок з використанням конструкції (If-Блок) $y = \begin{cases} 2 - x, & x < 0 \\ e^{x-1}, & 0 \leq x \leq 1 \\ \cos x^2, & x \geq 1 \end{cases}$
11	Обчислити значення кусочно-аналітичної функції 3 гілок з використанням конструкції (If-Блок) $y = \begin{cases} x^2, & x < -1 \\ 2x + 1, & -1 \leq x \leq 0 \\ x^2, & x > 0 \end{cases}$
12	Обчислити значення кусочно-аналітичної функції 3 гілок з використанням конструкції (If-Блок) $y = \begin{cases} \lg(x), & x < 0 \\ \sin(2x), & 0 \leq x \leq 1 \\ \sqrt{x+3}, & x \geq 1 \end{cases}$
13	Обчислити значення кусочно-аналітичної функції 3 гілок з використанням конструкції (If-Блок) $y = \begin{cases} x^2, & x < -1 \\ 2x + 1, & -1 \leq x \leq 0 \\ x^2, & x > 0 \end{cases}$
14	Обчислити значення кусочно-аналітичної функції 3 гілок з використанням конструкції (If-Блок) $y = \begin{cases} \sqrt{x}, & x < 1 \\ x^2 + 1, & 1 \leq x \leq 2 \\ 1 - 2x, & x > 2 \end{cases}$
15	Скласти алгоритм, що забезпечує введення трьох числових значень і наступне виведення їх у порядку зростання

6 ЛАБОРАТОРНА РОБОТА №6

6.1 Теоретичні відомості

Цикли без вкладення зі заздалегідь відомою кількістю повторень
(For...Next)

Цикл For...Next використовується, коли наперед відоме початкове та кінцеве значення лічильника. Повторює виконання групи інструкцій зазначене число раз. Синтаксис запису циклу For...Next має вигляд:

```
For <лічильник> [As <тип даних>] = <початок> To <кінець>
[Step <крок>]
    [інструкції]
[Exit For]
    [інструкції]
Next [лічильник]
```

лічильник обов'язковий. Числова змінна, використовувана в якості лічильника циклу;
початок обов'язковий. Початкове значення змінної лічильника;
кінець обов'язковий. Кінцеве значення змінної лічильника;
крок необов'язковий. Значення, на яке змінюється лічильник при кожному виконанні тіла циклу. Якщо це значення не задане, за замовченням крок дорівнює одиниці;
інструкції необов'язковий. Одна або декілька інструкцій між For і Next, що виконуються зазначене число раз.

Приклад використання циклу For...Next без вкладень:

```
Dim число As Integer
Dim сума As Integer
For число = 0 To 10
    сума = число + 1
Next число
MsgBox(число, MsgBoxStyle.Information, "Сообщение")
```

Цикли з вкладенням зі заздалегідь відомою кількістю повторень
(For...Next)

Допускається організація вкладених циклів For... Next (один цикл For... Next розташовується в середині іншого). Лічильник кожного циклу повинен мати унікальне ім'я.

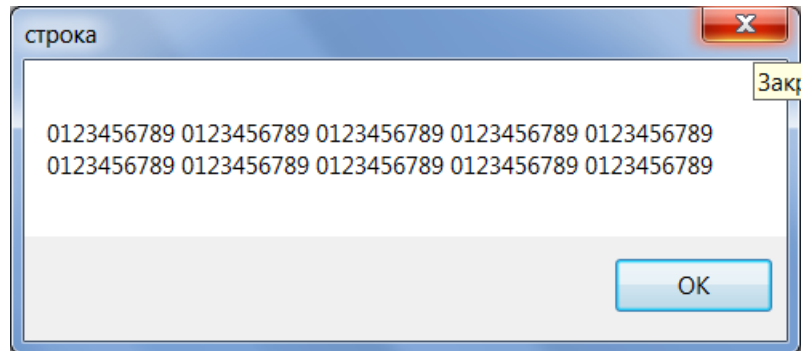
В наступному прикладі реалізований алгоритм формування та виведення на екран строк, які містять цифри від 0 до 9. Алгоритм заснований на застосуванні вкладених циклічних конструкцій For... Next.

```
Dim слова, цифра As Integer
Dim строка As String = ""
For слова = 10 To 1 Step -1
    For цифра = 0 To 9
        строка &= CStr(цифра) '
    
```

```

Next цифра
  строка &= " "
Next слова
MsgBox (строка, , "строка")

```



Цикли з невідомою заздалегідь кількістю повторень (While ... End While)

Оператор While завжди перевіряє умову до виконання циклу. Виконання циклу продовжується доки значення умови залишається True (Істина). Якщо умова дорівнює False (Хибність) при першому вході в цикл, вона не виконається жодного разу. Синтаксис запису має вигляд:

```

While <умова>
  [<інструкції>]
  [ Exit While ]
  [<інструкції>]
End While

```

Приклад використання циклу While ... End While. Алгоритм реалізує накопичення значень у змінній <сметчик>, та виведення на екран відповідно цьому інформації про кількість ітерацій.

```

Dim сметчик As Integer = 0
While сметчик < 20
  сметчик = сметчик + 1
End While
MsgBox ("Цикл прошел " & CStr(сметчик) & "раз")

```

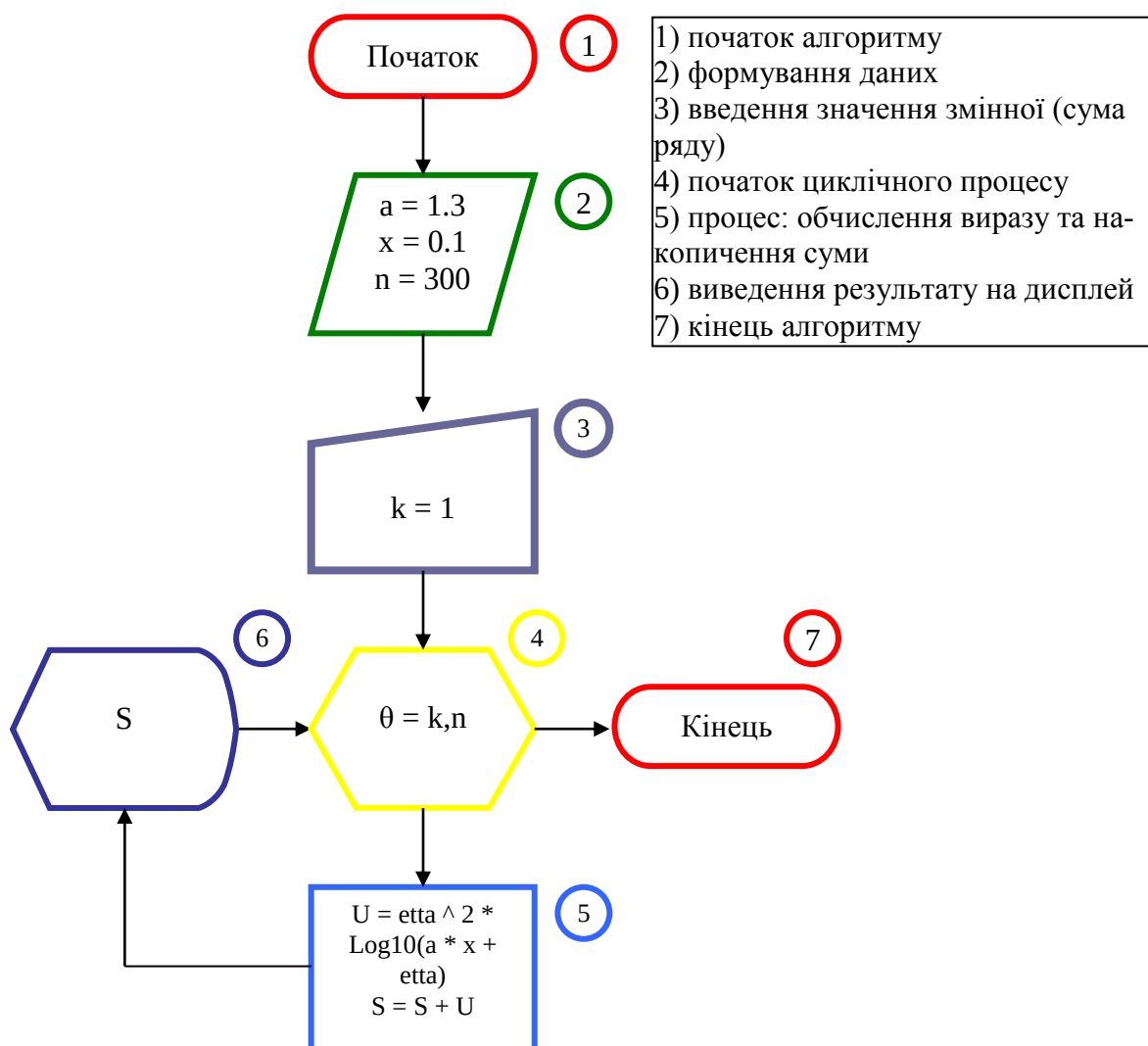
Розв'язок задачі «Одинарна сума / добуток числового ряду» (конструкція For...Next без вкладення)

Умова

$$\sum_{\theta=k}^n \theta^2 \lg(ax + \theta) \quad a = 1.3 \quad x = 0.1 \quad n = 300$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```
Dim etta As Integer
Dim a As Double = 1.3
Dim x As Double = 0.1
Dim n As Integer = 300 ' смотри (.Maximum)
Dim k As Integer = 1, S As Double, U As Double
tbxInf0.Text = ""
tbxInf0.Visible = True
ProgressBar1.Visible = True
tbxInf0.ScrollBars = ScrollBars.Horizontal
tbxInf0.ScrollBars = ScrollBars.Vertical
tbxInf0.Multiline = True ' многоуровневое поле
' _____ Работаем с ProgressBar _____
With ProgressBar1
    .Maximum = n ' ВНИМАНИЕ: (.Maximum)=реальному кол-ву
    итераций
    .Minimum = 0
```

```

        .Step = 1
        .Value = 0
    End With
    k = InputBox("Введите начальное значение счетчика",
"Определяет нижний индекс суммирования", 1)

```

В наступному блоці реалізується циклічний алгоритм без вкладення, обчислення членів ряду U , накопичення їх суми, та виведення результату в текстове поле:

```

        For etta = k To n
            ProgressBar1.PerformStep() 'Step-Приращение, на кото-
рое каждый вызов
            'метода PerformStep увеличивает текущее положение ин-
дикатора выполнения
            tbxInf0.Text = tbxInf0.Text & vbCrLf & "№" & etta &
":  U=" & U & "  S=" & S
            U = etta ^ 2 * Log10(a * x + etta)
            S = S + U
        Next

```

В наступному блоці реалізується функціональність елементу управління ColorDialog, який задає колір заливання та шрифту текстового поля tbxInf0:

```

        If cbxBGground.Checked = True Then
            If ColorDialog1.ShowDialog() = DialogResult.OK Then
                tbxInf0.BackColor = ColorDialog1.Color
            End If
        End If
        ' _____ Работа с ColorDialog1 _____
        If cbxColorFont.Checked = True Then
            If ColorDialog1.ShowDialog() = DialogResult.OK Then
                tbxInf0.ForeColor = ColorDialog1.Color
            End If
        End If

```

Тестування алгоритму

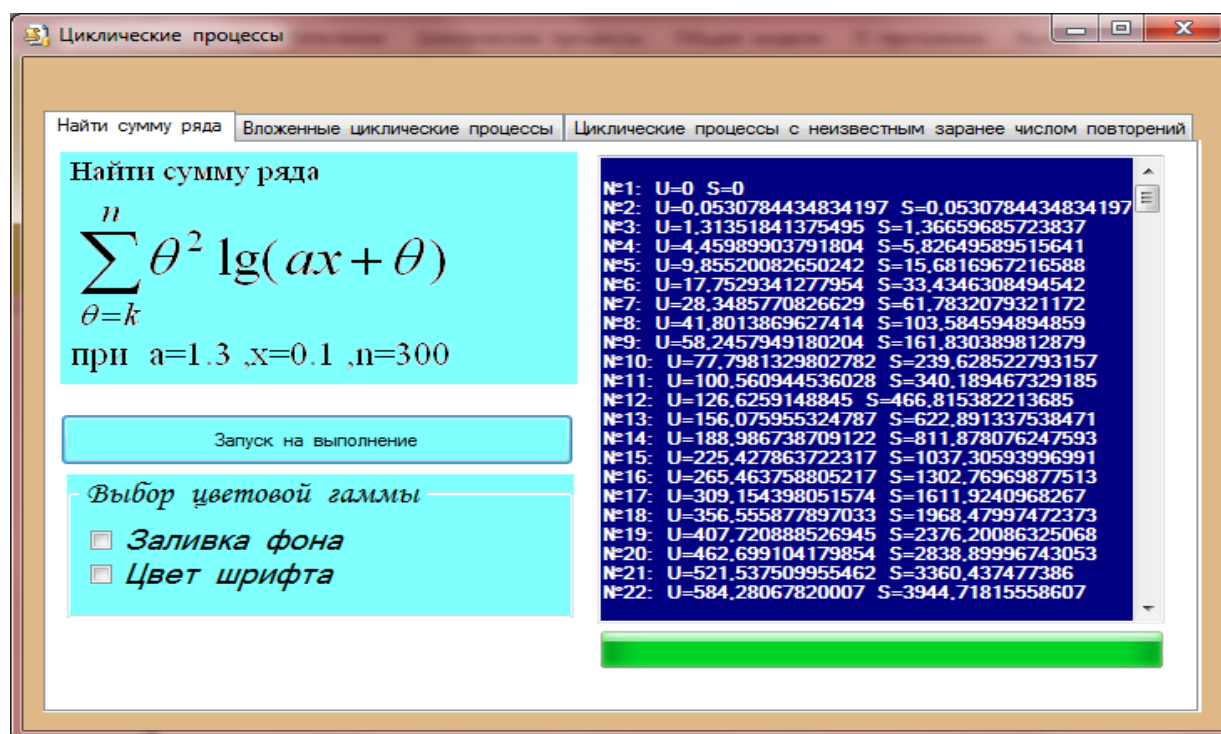


Рисунок 6.1 – Форма задачі «Пошук суми кінцевого ряду»

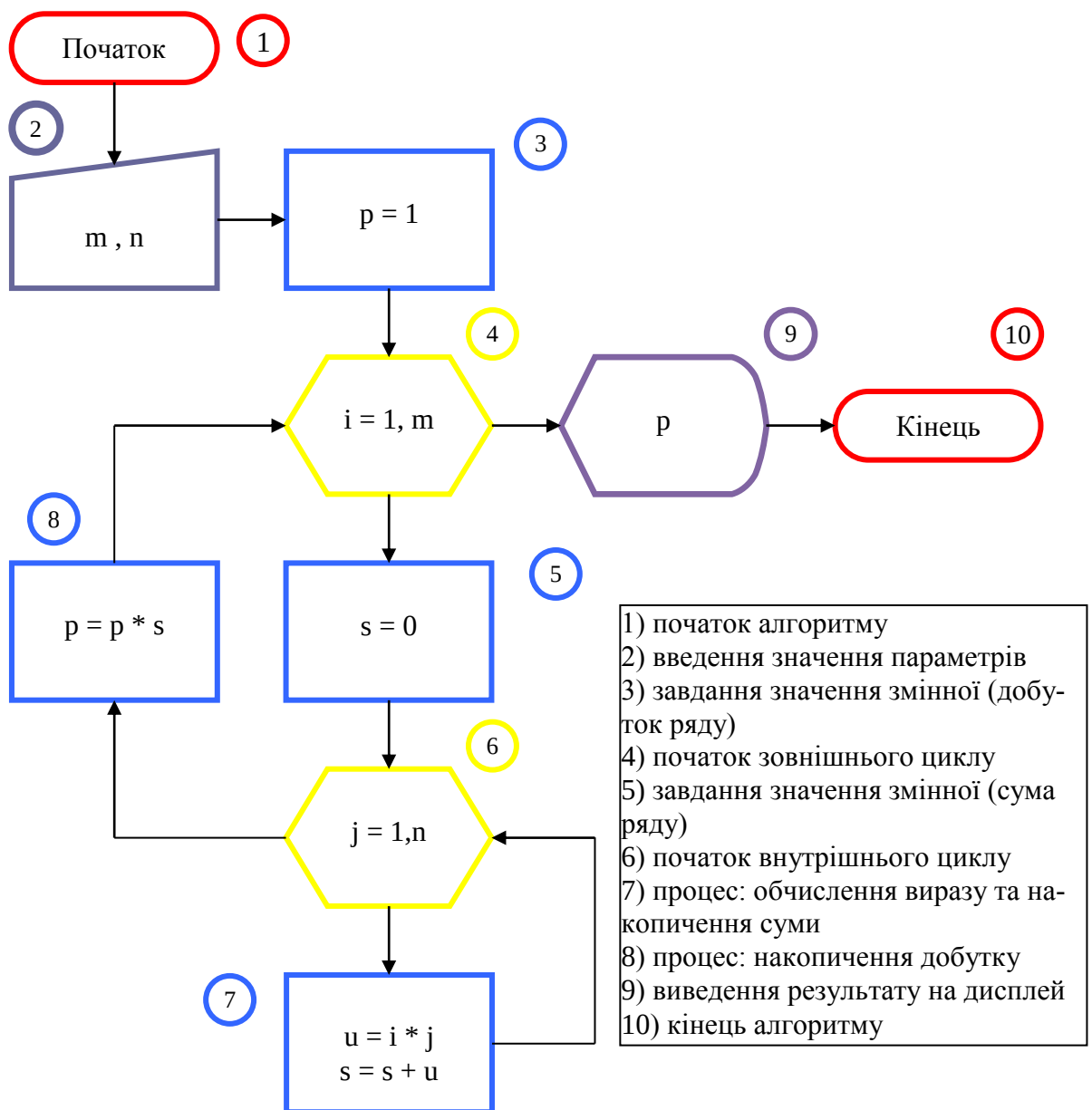
Розв'язок задачі «Подвійна сума / добуток числового ряду» (конструкція For...Next з вкладенням)

Умова

$$\prod_{i=1}^2 \sum_{j=1}^3 ij$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```

Dim i As Integer, m As Integer = tbxIn.Text, n As Integer =
tbxJn.Text
Dim j As Integer, u As Double, s As Double, p As Double = 1
With tbxInf1
    .ScrollBars = ScrollBars.Both
    .Multiline = True
    .Font = New System.Drawing.Font("Times New Roman", 14)
    .Text = "Тест внутрішнього цикла:"
End With
With tbxInf2
    .Text = "Тест зовнішнього цикла:"
    .ScrollBars = ScrollBars.Both
    .Multiline = True
  
```

```
.Font = New System.Drawing.Font("Times New Roman", 14)
End With
```

В наступному блоці реалізується циклічний алгоритм із вкладенням. У внутрішньому циклі (за змінної j) організовано обчислення членів ряду U , накопичення їх суми, та виведення результату в текстове поле. У зовнішньому циклі (за змінної i) організовано накопичення добутку суми членів ряду, та виведення результату в текстове поле:

```
For i = 1 To m : s = 0
    For j = 1 To n
        u = i * j : s = s + u
        tbxInf1.Text = tbxInf1.Text & vbNewLine & "U(" & j
& ")" & u
    Next
    tbxInf1.Text = tbxInf1.Text & vbNewLine &
"
    tbxInf1.Text = tbxInf1.Text & vbNewLine & " ЧАСТИЧНАЯ
СУММА: S(" & i & ")" & s
    p = p * s
    tbxInf2.Text = tbxInf2.Text & vbNewLine & "S(" & i &
")=" & s
Next
tbxInf2.Text = tbxInf2.Text & vbNewLine &
"
    tbxInf2.Text = tbxInf2.Text & vbNewLine & "P( ИТОГОВОЕ )" & p
    If cbxFont.Checked = True Then
        If FontDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Then
            tbxInf1.Font = FontDialog1.Font
            tbxInf2.Font = tbxInf1.Font
        End If
```

Тестування алгоритму

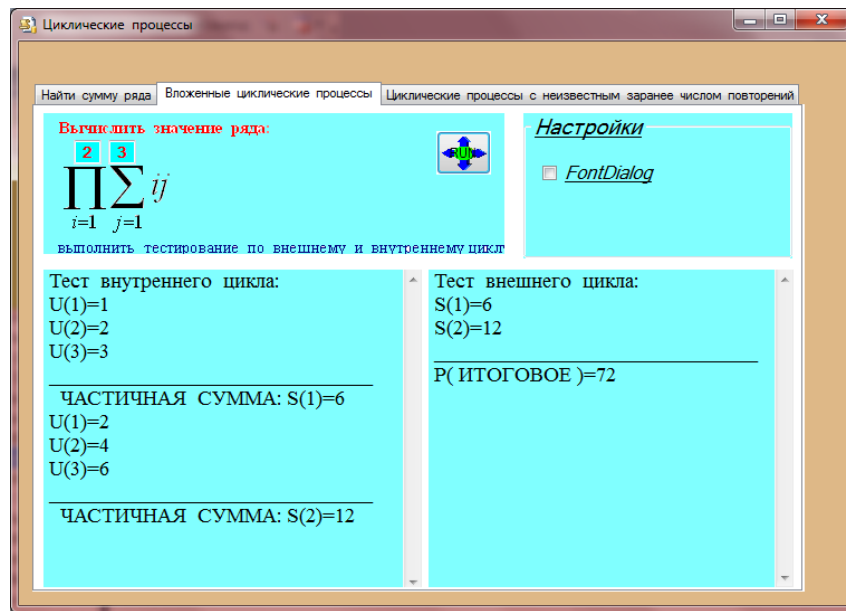


Рисунок 6.2 – Форма задачі «Пошук добутку суми кінцевого ряду»

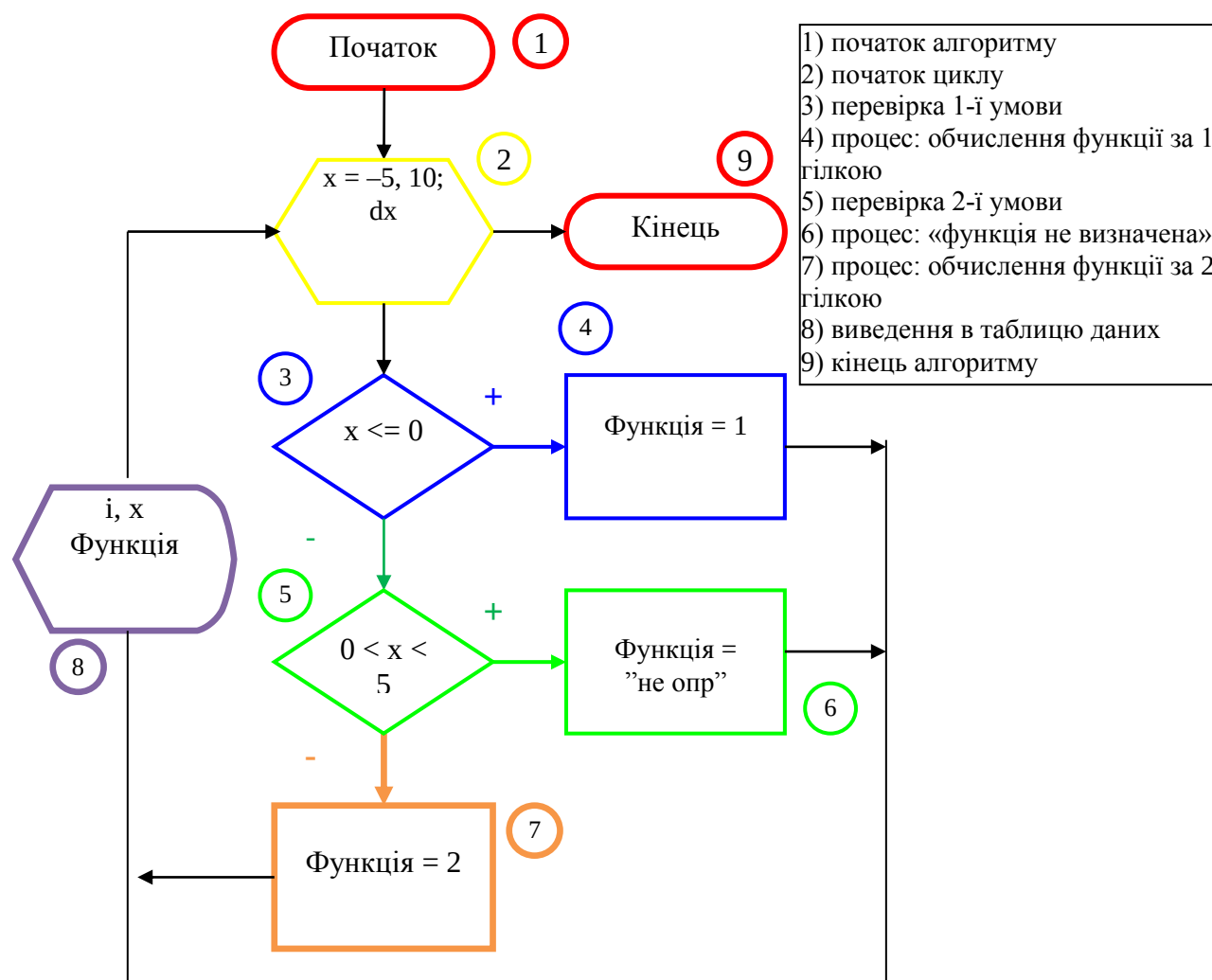
Розв’язок задачі «Таблювання функції з проколотою областю»

Умова. Протабулювати функцію з проколотою областю

$$y = \begin{cases} 1, & x \leq 0 \\ 2, & x \geq 5 \end{cases}$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```

Dim Функція
Dim шаг As Object, dx As Double, x As Double, i As Integer
шаг = MsgBox("Хотите ввести шаг ?", vbQuestion +
vbYesNo, "Відповідь_на_питання")
If шаг = vbYes Then
    dx = InputBox("Уведіть крок зміни аргументу", "Введен-
ня кроку", 0.51)
Else
    dx = 0.51
End If
DataGridView1.Rows.Add()

For x = -5 To 10 Step dx
    i = i + 1
    Select Case x
        Case Is <= 0
            Функція = 1
        Case 0 To 5 And x <> 5
            Функція = "не определена"
    
```

```

        Case Else
            Функція = 2
        End Select
        DataGridView1.Rows.Add()
        DataGridView1.Item(0, i).Value = i
        DataGridView1.Item(1, i).Value = Math.Round(x, 3)
        DataGridView1.Item(2, i).Value = Функція
    Next x

```

Тестування алгоритму

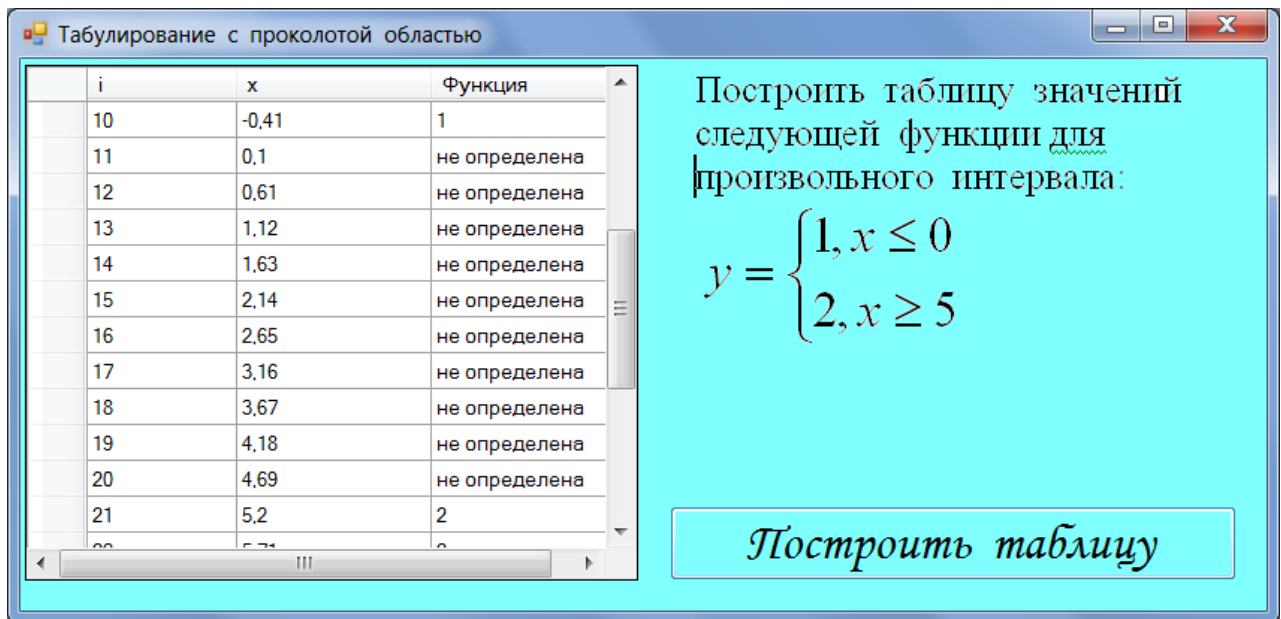


Рисунок 6.3 – Форма задачі «Табуювання функції з проколотою областю»

6.2 Практичне завдання

Завдання: згідно варіанту (табл. 6.1), реалізувати алгоритм табулювання функції, побудувати блок-схему, виконати тестування за гілками.

Таблица 6.1 – Варіанти індивідуальних завдань

1	$R = \begin{cases} \lg x, & x < 3 \\ \operatorname{ctg}(x), & x > 5 \end{cases}$	$-2 \leq x \leq 6 \quad \Delta x = 0,1$
2	$Q = \begin{cases} \operatorname{ctg}(x), & x < 0 \\ \operatorname{tg}(x), & x > 1 \end{cases}$	$-2 \leq x \leq 2 \quad \Delta x = 0,4$
3	$P = \begin{cases} \sqrt{x}, & x < 1 \\ \sqrt[3]{x}, & x > 2 \end{cases}$	$0 \leq x \leq 4 \quad \Delta x = 0,2$

4	$R = \begin{cases} \ln x, & x < 3 \\ \lg x, & x > 4 \end{cases}$	$2 \leq x \leq 5 \quad \Delta x = 0,1$
5	$P = \begin{cases} \sin^2(x), & x > 0 \\ \cos(x^3), & x > 1 \end{cases}$	$-1 \leq x \leq 2,3 \quad \Delta x = 0,2$
6	$R = \begin{cases} \ln(x+1), & x < 1 \\ \lg(x), & x > 2 \end{cases}$	$0 \leq x \leq 3 \quad \Delta x = 0,2$
7	$R = \begin{cases} e^{2x-1}, & x < 1 \\ \sqrt[3]{x}, & x > 2 \end{cases}$	$-1 \leq x \leq 5 \quad \Delta x = 0,2,$
8	$Q = \begin{cases} \ln x, & x < 2 \\ \lg x, & x > 3 \end{cases}$	$1 \leq x \leq 5 \quad \Delta x = 0,3$
9	$P = \begin{cases} \lg x, & x < 3 \\ \sqrt[3]{x}, & x > 6 \end{cases}$	$1 \leq x \leq 0 \quad \Delta x = 0,3$
10	$P = \begin{cases} \sin^2(x), & x < 1 \\ \cos(x^3), & x > 2 \end{cases}$	$0 \leq x \leq 3 \quad \Delta x = 0,1$
11	$Q = \begin{cases} \sin(x^3), & x < -2 \\ \cos^2(x), & x > 2 \end{cases}$	$-3 \leq x \leq 3 \quad \Delta x = 0,3$
12	$R = \begin{cases} \operatorname{ctg}(x), & x < 1 \\ \sqrt[3]{x}, & x > 2 \end{cases}$	$0,5 \leq x \leq 3 \quad \Delta x = 0,1$
13	$R = \begin{cases} \operatorname{tg}(3x) & x < 1 \\ \operatorname{ctg}(x^3) & x > 2 \end{cases}$	$0 \leq x \leq 3 \quad \Delta x = 0,1$
14	$Q = \begin{cases} \operatorname{ctg}(x^2) & x < 2 \\ \operatorname{tg}^3(x) & x > 3 \end{cases}$	$-1 \leq x \leq 5 \quad \Delta x = 0,3$
15	$R = \begin{cases} \sqrt[5]{x} & x < -1 \\ e^{1-x} & x > 1 \end{cases}$	$-2 \leq x \leq 2 \quad \Delta x = 0,1$
16	$P = \begin{cases} 3^{1-x} & x < 1 \\ e^{x+1} & x > 2 \end{cases}$	$0 \leq x \leq 3 \quad \Delta x = 0,2$

17	$P = \begin{cases} e^{2x} & x < 0 \\ 3^{2x} & x > 1 \end{cases}$	$-1 \leq x \leq 2$	$\Delta x = 0,2$
18	$P = \begin{cases} \sqrt[3]{x} & x < 0 \\ \sqrt{x} & x > 1 \end{cases}$	$-2 \leq x \leq 2$	$\Delta x = 0,2$
19	$R = \begin{cases} \sin^3(x) & x < -1 \\ \operatorname{tg}(2x+1) & x > 0 \end{cases}$	$-2 \leq x \leq 2$	$\Delta x = 0,2$
20	$Q = \begin{cases} \lg x & x < 2 \\ \sqrt{x} & x > 3 \end{cases}$	$1 \leq x \leq 4$	$\Delta x = 0,2$
21	$R = \begin{cases} e^{x-1} & x < 0 \\ 2^{x+1} & x > 1 \end{cases}$	$-2 \leq x \leq 3$	$\Delta x = 0,1$
22	$P = \begin{cases} \ln(x+1) & x < 3 \\ \lg(3x+1) & x > 4 \end{cases}$	$0 \leq x \leq 5$	$\Delta x = 0,5$
23	$R = \begin{cases} \lg x & x < 2 \\ \ln x & x > 3 \end{cases}$	$1 \leq x \leq 4$	$\Delta x = 0,3$
24	$Q = \begin{cases} \cos^2(x) & x < -1 \\ \sin(x^3) & x > 1 \end{cases}$	$-2 \leq x \leq 2$	$\Delta x = 0,2$
25	$R = \begin{cases} \log_3 x & x < 4 \\ \operatorname{ctg}(x), & x > 5 \end{cases}$	$-2 \leq x \leq 6$	$\Delta x = 0,1$
26	$Q = \begin{cases} \operatorname{arcctg}(x), & x < 0 \\ x \cdot \operatorname{arctg}(x), & x > 1 \end{cases}$	$-2 \leq x \leq 2$	$\Delta x = 0,4$
27	$P = \begin{cases} \operatorname{tg}(x), & x < 1,5 \\ \sqrt[3]{x}, & x > 2,5 \end{cases}$	$0 \leq x \leq 4$	$\Delta x = 0,2$
28	$R = \begin{cases} \cos(x) \cdot \ln x, & x < 3 \\ \operatorname{arcctg}(x) \cdot \lg x, & x > 4 \end{cases}$	$2 \leq x \leq 5$	$\Delta x = 0,1$
29	$P = \begin{cases} \sin^2(x^3), & x < 0 \\ \cos^3(x^2), & x > 1 \end{cases}$	$-1 \leq x \leq 2,3$	$\Delta x = 0,2$
30	$R = \begin{cases} \ln^3(x+1), & x < 1 \\ \lg^3(x+1), & x > 2 \end{cases}$	$0 \leq x \leq 3$	$\Delta x = 0,2$

ДОДАТОК А

А.1 Алгоритм. Види алгоритмів. Блок-схеми

Алгоритм – це послідовність дій над початковими даними та проміжними результатами необхідних для одержання кінцевого результату.

Алгоритм повинен мати наступні властивості:

- дискретність – процес рішення подається як послідовність кроків (етапів) і відбувається в часі дискретно;
- детермінованість (визначеність) – кожен крок повинен бути чітко визначеним і не повинен допускати довільного тлумачення;
- результативність – алгоритм повинен приводити до рішення задачі за скінчене число кроків;
- масовість – алгоритм рішення задачі розробляється в загальному виді, таким чином, щоб він міг бути застосованим для цілого класу задач, що різняться лише початковими даними.

Розрізняють:

- лінійний алгоритм;
- алгоритм з розгалуженням (за порядком розгалуження);
- циклічний алгоритм.

Алгоритм розв'язку будь-якої задачі, яка піддається формалізації, може бути поданий як комбінація цих трьох вище названих.

Алгоритм можна формалізувати словесно, у вигляді графічної схеми чи програмного коду на одній з мов програмування.

Побудова блок-схеми алгоритму дозволяє суттєво спростити подальший процес його формалізації у формі програмного коду. Основні графічні примітиви (блоки) з яких складається блок-схема та їх формати наведені на рис. А.1.

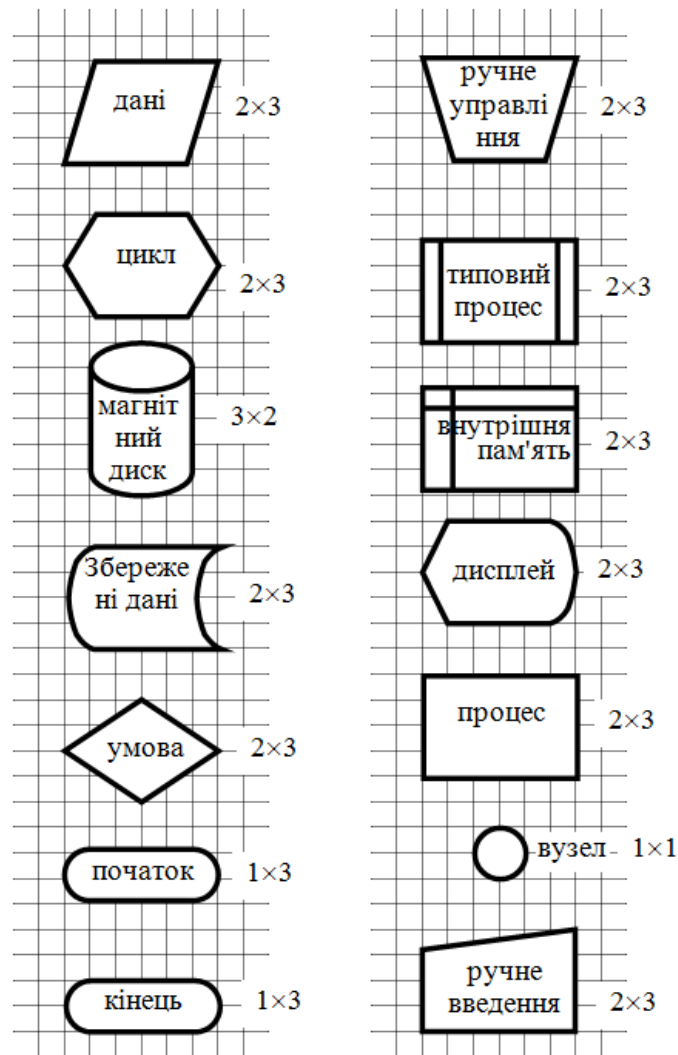


Рисунок А.1 – Формат основних блоків схем

Розглянемо більш детально кожний з типів алгоритмів.

1. Лінійний алгоритм характеризується однонаправленим послідовним переходом від блоку до блоку (від оператора до оператора) по мірі їх виконання.

2. Розгалужений алгоритм має місце в тому разі, коли в залежності від виконання чи невиконання умови виконується одна чи друга дія.

3. Множинний вибір є узагальненням розгалуження, коли в залежності від значення змінної виконується одна з багатьох дій.

4. Циклічний алгоритм передбачає багаторазове повторення програмних дій (обчислень, введення даних, виведення результатів, тощо) при різних початкових умовах.

ДОДАТОК Б

Б.1 Клас Windows Forms. Відомості про об'єктно-орієнтоване програмування

У Windows Forms термін «форма» – синонім вікна верхнього рівня. Головне вікно програми – форма. Будь-які інші вікна верхнього рівня, які має додаток – також форми. Вікна діалогу також вважаються формами. Незважаючи на назву, додатки, що використовують Windows Forms, не виглядають як форми. Подібно традиційним Windows-додаткам програми здійснюють повний контроль над подіями у власних вікнах.

Програми, що використовують Windows Forms використовують класи System.Windows.Forms. Цей розділ включає такі класи, як Form, який моделює поведінку вікон або форм; Menu, який представляє меню; Clipboard, який дає можливість додаткам Windows Forms використовувати буфер обміну. Він також містить численні класи, які надають засоби управління, наприклад: Button, TextBox, ListView, MonthCalendar і т. д. Ці класи можуть бути включені в додаток: з використанням тільки імені класу, або з використанням повного імені, наприклад: System.WinForms.Button.

В основі майже кожної програми, написаної зі застосуванням Windows Forms, – похідний клас від System.Windows.Forms. Зразок цього класу представляє головне вікно програми. System.Windows.Forms має безліч властивостей і методів, які мають багатий програмний інтерфейс до форм.

Додатки, засновані на Windows Forms, які використовують кнопки, списки та інші типи компонентів Windows, використовують класи управління System.Windows.Forms, що значно спрощує програмування інтерфейсу.

Основні поняття об'єктно-орієнтованого програмування

До появи персональних комп'ютерів технологія створення комп'ютерних програм базувалася на процедурному програмуванні, в якому основою були функції й процедури, тобто дії. Програміст визначав, які дії й обчислення потрібні для вирішення поставленої задачі, потім описував ці дії у вигляді процедур і функцій та об'єднував їх в програму. Створена в такий спосіб комп'ютерна програма відрізнялася чітким алгоритмом роботи – послідовністю дій для досягнення поставленої мети.

Об'єктно-орієнтоване програмування (ООП) зародилося з появою ПК в мовах програмування Pascal, Ада, Smalltalk, С. У ООП головною відповідною точкою при проектуванні програми є не процедура, не дія, а об'єкт. Такий підхід є досить зрозумілим, оскільки в реальному світі нас оточують саме об'єкти, які взаємодіють один з одним.

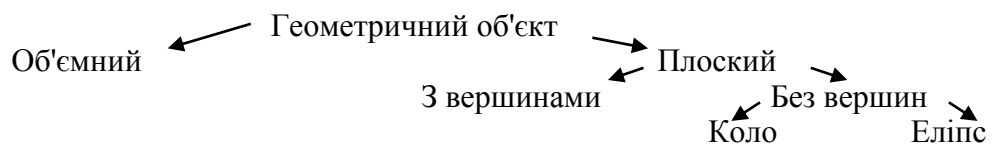
Об'єкти

Назвемо об'єктом поняття, абстракцію або будь-який предмет з чітко окресленими межами, що має сенс у контексті розглянутої прикладної проблеми. Всі об'єкти можна ідентифікувати, між об'єктами можна встановити відношення тотожності (два Сидорових Івана – це різні люди, але вони студенти однієї групи).

Класи об'єктів

Класом називають особливу структуру, яка може мати у своєму складі поля, методи та властивості. Клас виступає в якості об'єктного типу даних, а об'єкт – це конкретний екземпляр класу. Наприклад, два Сидорових Івана належать одному й тому ж класу об'єктів, вони – студенти групи. Саме з цим пов'язана їх однаковість (однаковий шифр групи, один розклад занять і т. д.).

Кожен конкретний клас має свої особливості поведінки і характеристики, що визначають цей клас.



Найвищий рівень – загальний і найпростіший, кожний наступний рівень більш специфічний і менш загальний. На самому останньому рівні можна визначити колір, стиль заповнення, величину радіусу кола і т. п.

Якщо характеристика вже одного разу визначена для більш високого рівня, то всі рівні, розташовані нижче мають ту ж характеристику (якщо вже визначена окружність, зрозуміло, що вершин у неї немає).

Таким чином, класи-спадкоємці можуть успадковувати характеристики класів-батьків.

Властивості

Властивості – перелік параметрів об'єкта, які визначають зовнішній вигляд і поведінку об'єкта, виділяють унікальні особливості кожного екземпляра. До властивостей відносяться: ім'я, тип, значення, колір, розмір та ін. Стан – сукупність всіх властивостей даного об'єкта.

Метод

Метод – це деяка дія (операція), яку можна виконувати над даним об'єктом. У результаті цієї дії змінюється об'єкт (наприклад, місце розташування, колір та ін.) Іншими словами, методом називається команда, яку може виконувати об'єкт. Для кожного класу об'єктів є свій перелік методів, які можна до нього застосувати або які він може виконати. Наприклад, об'єкт можна видалити з екрану, перемістити в інше місце.

Події

Кожен об'єкт здатний реагувати на певні події – це різновид властивостей об'єкта. При виникненні події проводиться її обробка.

Події – сигнали, що формуються зовнішнім середовищем, на які об'єкт повинен відреагувати відповідним чином.

Події настають в результаті дій користувача – переміщення курсору, натискання кнопок миші або клавіш на клавіатурі, а також у результаті роботи самих об'єктів. Для кожного об'єкта визначено безліч подій, на які він може реагувати. Для конкретних екземплярів об'єкта можуть бути визначені обробники якихось із цих подій, які й визначають реакцію даного екземпляра об'єкта.

Об'єкт можна визначити як сукупність властивостей і методів, а також подій, на які він може реагувати.

Зовнішнє управління об'єктом здійснюється через обробник подій. Ці обробники звертаються до методів та властивостей об'єкта.

Б.2 Загальні властивості для основних елементів управління

Таблиця Б.1 – Елементи управління VB

Тип об'єкта	Призначення	Префікс
Label	Напис	lbl
TextBox	Текстове поле	txt
CommandButton	Кнопка	cmd
CheckBox	Прапорець	chk
OptionButton	Перемикач	opt
Frame	Рамка	fra
ListBox	Список	lst
ComboBox	Поле зі списком	cbo
PictureBox	Графічний фрейм	pic
Form	Форма	frm

Таблиця Б.2 – Загальні властивості елементів управління VB

Властивості	Описання
BackColor	Задає фоновий колір об'єкта. Колір можна задати, вибравши його з палітри кольорів або вказавши прив'язку до поточної колірної схеми операційної системи
Backstyle	Управляє відображенням фону області виведення тексту. Область виводу тексту може бути зафарбована (Opaque) або бути прозорою (Transparent).
Font	Шрифт, який використовується для відображення тексту на об'єкті
Text	Напис, що розміщується на об'єкті
Name	Вказує ім'я, яке використовується для ідентифікації об'єкта в програмі
Left	Відстань від лівої межі об'єкта до лівої границі форми
Top	Відстань від верхньої межі об'єкта до верхньої межі форми
Height	Висота компонента
Width	Ширина компонента
Visible	Дозволяє приховати компонент (значення властивості – False) або зробити його видимим (значення властивості – True)
Enabled	Дозволяє активувати об'єкт

1. Button

Об'єкт управління **Button**(кнопка):  **Button** .

За допомогою клікання мишею на кнопці можна задати виконання якої-небудь дії.

2. CheckBox

Об'єкт управління **CheckBox** :  **CheckBox** .

Для введення в форму даних, які можуть мати тільки одне з двох допустимих значень, призначені елементи управління називаються прапорцями.

Прапорці, наприклад, дозволяють користувачу дати відповідь на поставлене питання. У разі позитивної відповіді користувач встановлює прапорець, і він набуває вигляд квадрата, в якому розміщена галочка: .

Таблиця Б.3 – Властивості елемента управління CheckBox

Властивості	Описання
Name Caption	Текст, який знаходиться праворуч від прапорця
Checked	Стан перемикача: прапорець встановлено (у елементі є «галочка»)
Unchecked	Прапорець скинутий (нема «галочки»)

3. Panel

Об'єкт управління **Panel**(панель):  **Panel** .

Елемент управління **Panel** являє собою панель і використовується в формах в основному для об'єднання елементів в групи, наприклад, для розміщення у формі двох і більше груп перемикачів. Елемент, що розташовується на елементі управління **Panel**, автоматично стає його елементом.

Таблиця Б.4 – Властивості елемента управління Panel

Властивості	Описання
AutoScroll	Керує появленням смуг прокрутки

4. Label

Об'єкт управління **Label**:  **Label** .

Компонент **Label** призначений для виведення тексту на поверхню форми. Властивості компонента визначають вид і розташування тексту на поверхні форми.

Таблиця Б.5 – Властивості елемента управління Label

Властивості	Описання
Caption	Відображуваний текст
AutoSize	Ознака того, що розмір поля визначається його вмістом
Wordwrap	Ознака того, що слова, які не вміщуються у поточному рядку, автоматично переносяться на наступний рядок

5. Об'єкт управління ComboBox: ComboBox.

Списки типу ComboBox називають полями, що розкриваються або полями зі списками. Для вибору значення із списку спочатку необхідно список відкрити, натиснувши кнопку зі стрілкою, розташовану з правого боку поля вводу: .

ComboBox поєднує функції списку ListBox і поля введення TextBox. Використання списків ComboBox дозволяє представляти більший обсяг інформації, заощаджуючи при цьому місце у формі.

Події.

Для поля зі списком важливу роль відіграють події як поля введення, так і списку. Основні з них – Click, яка використовується для вибору елемента списку, та Change – зміна запису в полі введення тексту.

Таблиця Б.6 – Властивості елемента управління ComboBox

Властивості	Описання
Caption	Відображуваний текст
List	Елементи списку – масив рядків
ListCount	Кількість елементів списку
Index	Номер елемента, вибраного у списку. Якщо жоден з елементів списку не був обраний, то значення властивості дорівнює –1. Нумерація елементів починається з нуля
Style	Стиль (вид) списку. Якщо значення властивості рівне DropdownCombo (0), то дані в полі редагування можна ввести з клавіатури або вибрати зі списку (щоб отримати доступ до списку, його треба розкрити). Якщо значення властивості рівне Simple-Combo (1), то дані можна ввести в полі редагування з клавіатури або вибрати зі списку, причому список доступний завжди. Якщо значення властивості рівне DropDownList (2), то дані в полі редагування можна ввести лише шляхом вибору зі списку
Text	Вміст поля редагування (дані, введенні користувачем з клавіатури або виділені зі списку)
Sorted	Ознака необхідності автоматичного сортування списку після додавання чергового елемента (значення властивості – True)
Locked	Блокування списку. Визначає заборону вибору елемента зі списку (рядок введення також блокується)

6. Timer

Об'єкт управління Timer:  Timer.

Використання таймера є гарним способом управління програмою. За допомогою таймера можна запускати або завершувати процеси додатки в певні моменти часу. Під час проектування таймер розміщується на формі, під час виконання програми його на формі не видно.

Таймер має єдину подію – Timer, яка викликається після закінчення встановленого тимчасового інтервалу.

Таблиця Б.7 – Властивості елемента управління Timer

Властивості	Описання
Name	Ім'я компоненту. Використовується для доступу до компонента та його властивостей
Interval	Період генерації події Timer. Задається в мілісекундах
Enabled	Дозвіл генерації (значення властивості – True) або блокування (значення властивості – False) події Timer

7. ProgressBar

Об'єкт управління ProgressBar:  **ProgressBar**.

Відображення процесу виконання операцій. У Windows цей елемент відображає перебіг процесу копіювання, переміщення, завантаження й т. п.

Таблиця Б.8 – Властивості елемента управління ProgressBar

Властивості	Описання
Min	Нижня межа
Max	Верхня межа
Value	Поточне значення
Orientation	Орієнтація елементів: – Вертикальна (ccOrientationVertical) – Горизонтальна (ccOrientationHorizontal)
Scrolling	Спосіб відображення процесу: – Безперервний (ccScrollingSmooth) – Сегментарний (ccScrollingStandard)

8. TrackBar

Об'єкт управління TrackBar:  **TrackBar**.

За допомогою TrackBar можна задавати значення змінних з діапазону значень.

Таблиця Б.9 – Властивості елемента управління TrackBar

Властивості	Описання
Min	Мінімальне значення для бігунка, коли він знаходиться на лівому краю
Max	Максимальне значення для бігунка, коли він знаходиться на правому краю
Value	Поточне положення бігунка
SmallChange	Мала зміна. Зрушення бігунка на одну позицію. Крок бігунка
LargeChange	Велика зміна. Зрушення бігунка при подвійному натисканні на ньому

9. TextBox

Об'єкт управління TextBox:  **TextBox**.

Компонент TextBox являє собою поле введення-редагування тексту. Події.

При використанні текстового поля викликає інтерес декілька подій.

Насамперед, подія Change, яка викликається при зміні змісту текстового поля, ця подія відбувається кожен раз при введенні, знищенні або зміні символу. Наприклад, при введенні в текстове поле слова «так», подія

Change відбувається три рази – по одному разу для кожної літери. Для аналізу введеного в поле тексту найкраще всього підходить подія LostFocus. Ця подія викликається після того, як текстове поле зробиться неактивним (після передачі фокуса іншому елементу, коли користувач закінчить введення). Однак, якщо це поле є єдиним елементом управління в формі, то воно не може втратити фокус.

Для того, щоб видалити або ініціалізувати вміст текстового вікна, використовується подія GotFocus. Вона викликається тоді, коли користувач «входить» в текстове вікно.

Таблиця Б.10 – Властивості елемента управління TextBox

Властивості	Описання
Text	Текст, що знаходиться в полі редагування
Locked	Використовується для обмеження можливості змінити текст в полі редагування. Якщо значення властивості рівне False, то текст в полі редагування змінити не можна
MultiLine	Робить можливим багаторядкове відображення тексту
ScrollBars	Управляє відображенням смуг прокрутки: Vertical – тільки смуга вертикальної прокрутки; Horizontal – тільки смуга горизонтального прокрутки; Both – вертикальна і горизонтальна смуги прокрутки; None – без смуг прокрутки

10. ListBox

Об'єкт управління ListBox :  ListBox

Компонент ListBox являє собою список, в якому можна вибрати потрібний елемент.

Події.

Основна подія списку – Click. Ця подія викликається, якщо користувач за допомогою мишки або клавіш управління вибере елемент зі списку.

Методи.

Вікно списку – це перший з елементів управління, для якого важливу роль відіграють методи. Методи списку необхідні для обробки елементів списку – додавання або видалення. Для додавання нових елементів використовується метод AddItem: ListBox.AddItem Елемент [Індекс].

Для видалення елемента зі списку використовується метод RemoveItem, якому, як параметр передається індекс елемента, який видаляється. Індексція елементів починається з нуля (0):

ListBox.RemoveItem – Індекс елемента

Для видалення усіх елементів використовується метод Clear:

ListBox.Clear

Таблиця Б.11 – Властивості елемента управління ListBox

Властивості	Описання
ForeColor	Колір, що використовується для відображення елементів списку
List	Елементи списку – масив рядків
ListCount	Кількість елементів списку
Sorted	Ознака необхідності автоматичного сортування списку після додавання чергового елемента
Listindex	Номер вибраного елемента списку (нумерація елементів списку починається з нуля)
MultiSelect	Дозволяє зробити можливим множинний вибір зі списку. Якщо значення цієї властивості – None (0), то вибрати кілька елементів зі списку не можна. При значенні, рівному Simple (1), кожен елемент, на який клікнули, стає обраним. Скасування вибору виробляється за допомогою повторного клікання миші або за допомогою клавіші-пробіл. Якщо значення властивості рівне Extended (2), то множинний вибір здійснюється за допомогою миші й клавіші <Shift> або <Ctrl> (клацання кнопкою миші на елементі списку, утримуючи клавішу <Shift> позначає елемент як обраний, повторне клацання скасовує вибір)
Style	Стиль (вид) списку. Якщо значення властивості дорівнює CheckBox (1), то перед кожним елементом списку відображається компонент Checkbox. При цьому для вибору елемента із списку потрібно встановити відповідний прапорець. При значенні властивості, рівному Standard (0), список має стандартний вигляд

11. PictureBox

Об'єкт управління PictureBox:  PictureBox .

Компонент PictureBox забезпечує відображення графіки. Зображення можна завантажити з файлу або сформувати з графічних примітивів (намалювати) під час роботи програми.

Таблиця Б.12 – Властивості елемента управління PictureBox

Властивості	Описання
Picture	Зображення (об'єкт Bitmap), яке відображається у полі компонента. Завантажити картинку можна під час розробки форми або з файлу під час роботи програми (функція LoadPicture)
AutoSize	Властивість дозволяє (True) або забороняє (False) автоматичну зміну розміру компонента (області виведення ілюстрації) відповідно до розміру картинки, завантаженої в компонент
BorderStyle	Стиль межі компонента. Якщо значення властивості – FixedSingle (1), то межа стандартна (тонка лінія), якщо – None (0), то межа не відображається
ForeColor	Шрифт, яким метод Print виводить текст. Для методу Print задає колір символів, для методів креслення графічних примітивів (об'єктів) – колір ліній
FillColor	Встановлює колір зафарбовування внутрішніх областей графічних примітивів (об'єктів), викреслює в полі (на поверхні) компонента
DrawStyle	Вид контуру графічних об'єктів, викреслює в полі компонента відповідними методами: Solid (0) – суцільна лінія; Dash (1) – пунктирна лінія; Dot (2) – лінія з точок; Dash – Dot (3) – лінія виду "точка -тире "; Dash – Dot – Dot (4) – лінія виду «тире – крапка – крапка»; Transparent (5) – «прозора» лінія

Закінчення таблиці Б.12

DrawWidth	Товщина ліній для графічних об'єктів
Sorted	Ознака необхідності автоматичного сортування списку після додавання чергового елемента
Listindex	Номер вибраного елемента списку (нумерація елементів списку починається з нуля)
MultiSelect	Дозволяє зробити можливим множинний вибір зі списку. Якщо значення цієї властивості – None (0), то вибрати кілька елементів зі списку не можна. При значенні, рівному Simple (1), кожен елемент, на який клікнули, стає обраним. Скасування вибору виробляється за допомогою повторного клікання миші або за допомогою клавіші пробіл. Якщо значення властивості рівне Extended (2), то множинний вибір здійснюється за допомогою миші й клавіші <Shift> або <Ctrl> (клацання кнопкою миші на елементі списку, утримуючи клавішу <Shift> позначає елемент як обраний, повторне клацання скасовує вибір)
Style	Стиль (вид) списку. Якщо значення властивості рівне CheckBox (1), то перед кожним елементом списку відображається компонент Checkbox. При цьому для вибору елемента із списку потрібно встановити відповідний прапорець. При значенні властивості, рівній Standard (0), список має стандартний вигляд

12. DataGridView

Об'єкт управління DataGridView:  DataGridView .

DataGridView надає потужний і гнучкий спосіб відображення даних в табличному форматі. DataGridView можна використовувати для подання в режимі тільки читання невеликих об'єктів даних; можна розширити цей елемент для представлення великих обсягів даних в режимі редагування.

Таблиця Б.13 – Властивості елемента управління PictureBox

Член, властивість, метод	Описання
DataGridView.Item (ИндСтолбца, ИндСтроки) значение = Str(dataGridView1(Столб, Стр).Value) 'получить значение ячейки для ячейки в столбце Столб, строке Стр значение = Str(dataGridView1("ИмяСт", Стр).Value) 'получить значение ячейки для ячейки в столбце "ИмяСт", строке Стр	Надає індексатор для отримання або задання клітинки, розташованої на перетині стовпця (ИндСтолбца) і рядки (ИндСтроки) з заданими індексами
DataGridView1.Rows.Add()	Додає новий рядок в колекцію
DataGridView1.RowCount = число_строк	Повертає або задає кількість рядків, що відображаються в об'єкті DataGridView
DataGridView1.ColumnCount = число_столбцов	Повертає або задає число стовпців, які відображаються в об'єкті DataGridView
DataGridView1.ColumnHeadersVisible = True (False)	Повертає або задає значення, яке вказує, чи відображається рядок заголовків стовпців
DataGridView1.RowHeadersVisible = True (False)	Повертає або задає значення, яке вказує, чи відображається стовпець, що містить заголовки рядків

ДОДАТОК В СТРУКТУРНІ ЕЛЕМЕНТИ КУРСОВОЇ РОБОТИ

Приклад титульного листа

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ
АВТОМОБІЛЬНО-ДОРОЖНІЙ ІНСТИТУТ
ДЕРЖАВНОГО ВИЩОГО НАВЧАЛЬНОГО ЗАКЛАДУ
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»

Факультет ЕУ
Кафедра ПМІ

КУРСОВА РОБОТА

на тему:

СТВОРЕННЯ ПРОЕКТІВ, РОБОТА З ЗОВНІШНІМИ ФАЙЛАМИ,
МАСИВИ, МОДУЛІ, ГРАФІКА

Виконав: ст. гр. МО-***а
Іванов О.О.
№ з/к: ##-###

Перевірив: _____

Горлівка 20__

Приклад написання реферату (анотації)

РЕФЕРАТ

Сторінок -38, рисунків -12, джерел -5.

Об'єктом дослідження цієї курсової роботи є побудова проекту у Microsoft Visual Studio, а саме з використанням мови програмування Visual Basic. Net. До проекту входять роботи з масивами, матрицями, використання модулів та розробка графіку.

Мета даної курсової роботи полягає в вивченні мови програмування Visual Basic. Net, створенні проекту за допомогою цієї мови, застосування таблиць, формул, рисунків. Окрім цього у реферативній частині ми розглянемо окремі функції та можливості Word, Excel, Access. Наприклад, ми розглянемо границі та заливання у Word як стосовно усієї сторінки, так і окремих абзаців та таблиць. Також розглянемо побудову різних типів діаграм за допомогою Excel та відредагуємо їх. І нарешті ми дізнаємось, що таке база даних, які вони бувають та чим ними керувати.

Отже, в процесі виконання даної курсової роботи необхідно вивчити мову програмування Visual Basic.Net, навчитись працювати з файлами, таблицями, графіками, процедурами, процедурами функціями. Результатом роботи є вільне володіння мовою Visual Basic.Net та вміння використовувати Microsoft Visual Studio у створюванні проектів.

АЛГОРИТМ, ВЕКТОР, ПРОЦЕДУРА, ПРОЦЕДУРА-ФУНКЦІЯ, ГРАФІКА,
ОПЕРАТОР, МАСИВ, ТАБУЛЮВАННЯ, ЗОВНІШНІ ФАЙЛИ.

Приклад структури змісту

ЗМІСТ

ВСТУП.....	7
1. РЕФЕРАТИВНІ ПОВІДОМЛЕННЯ	
1.1 Створення Web-сторінок	8
1.2 Призначення макросу кнопки	11
1.3 Настроювання запиту	13
1.4 Виділення файлів. Перегляд файлів. “Оцінити ступінь стискання”	14
1.5 Комп’ютерний вірус. Антивірусна програма	16
2. ПРАКТИЧНА ЧАСТИНА	
2.1 Масиви	
2.1.1 Вектор	20
2.1.2 Матриця 1	22
2.1.3 Матриця 2	25
2.2 Модулі	
2.2.1 Процедура	28
2.2.2 Процедура-функція	31
2.3 Графіка	
2.3.1 Графік функції	33
ВИСНОВОК	37
ПЕРЕЛІК ЛІТЕРАТУРИ	38

ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Абрамян М. Э. Практикум по программированию на языках С# и VB.NET. 2-е изд. / М. Э. Абрамян. – Ростов н/Д.: «ЦВВР», 2007. – 514 с.
2. Балена Ф. Современная практика программирования на Microsoft Visual Basic и Visual C# / Ф. Балена, Дж. Димауро. – М.: Русская редакция, 2006. – 970 с.
3. Дубовцев А. В. Microsoft .NET в подлиннике / А. В. Дубовцев. – СПб.: БХВ-Петербург, 2004. – 850 с.
4. Нортроп Т. Основы разработки приложений на платформе Microsoft .NET Framework / Т. Нортроп, Ш. Уилдермьюс, Б. Райан. – М.: Русская редакция, 2007. – 730 с.
5. Рихтер Дж. Программирование на платформе Microsoft .NET Framework. Мастер-класс. 3-е изд. / Дж. Рихтер. – М.: Русская редакция, 2005. – 565 с.
6. Microsoft Corporation. Разработка Windows-приложений на Microsoft Visual Basic .NET и Microsoft Visual C# .NET. – М.: Русская редакция, 2003. – 645 с.
7. Зиборов В. В. Visual Basic 2010 на примерах / В. В. Зиборов. – СПб.: БХВ-Петербург, 2010. – 336 с.
8. Методичні вказівки до вивчення дисципліни «Економічна інформатика» (для студентів напряму 6.030601 «Менеджмент» усіх форм навчання) [Електронний ресурс] / укладачі М. Є. Корольов, Р. С. Кравченко. – Електрон. дані. – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2012, 76 с. [МВ № 22/14-2012-11]
9. Методичні вказівки до навчальної комп'ютерної практики. Частина 1. «Електронні таблиці MS Excel» (для студентів напрямів підготовки 6.030601 «Менеджмент», 6.070101 «Транспортні технології (автомобільний транспорт)» усіх форм навчання) [Електронний ресурс] / укладачі: М. Є. Корольов, Р. С. Кравченко. – Електрон. дані. – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2012, 77 с. [МВ № 22/16-2012-15].
10. ДСТУ 3008-95 «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення».
11. Держстандарт 2.105-95 «Загальні вимоги до текстових документів».
12. Програмний мультимедійний навчальний комплекс «АРМ викладач-студент» (Електронний носій).