Національний університет
водного господарства
та природокористування

Міністерство освіти і науки України
Національний університет водного господарства та природокористування

Кафедра комп'ютерних наук

04-05-15

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт
з навчальної дисципліни «Програмування»
(Частина 1. Основи мови програмування C#)
для здобувачів вищої освіти першого
(бакалаврського) рівня за спеціальністями
126 «Інформаційні системи та технології»
015.10 «Професійна освіта. Комп'ютерні технології»
денної та заочної форм навчання

Рекомендовано методичними комісіями зі спеціальностей
126 «Інформаційні системи та технології»
Протокол № 8 від 07.03.2019 р.
015.10 «Професійна освіта. Комп'ютерні технології»
Протокол № 6 від 07.03.2019 р.

Рівне – 2019

Національний університет
водного господарства
та природокористування

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Програмування» (Частина 1. Основи мови програмування C#) для здобувачів вищої освіти першого (бакалаврського) рівня за спеціальностями 126 «Інформаційні системи та технології», 015.10 «Професійна освіта. Комп'ютерні технології» денної та заочної форм навчання / Шевченко І. М. – Рівне : НУВГП, 2019. – 72 с.

Укладач:
Шевченко І. М., ст. викладач кафедри комп'ютерних наук.

Відповідальний за випуск: Тулашвілі Ю. Й., доктор педагогічних наук, професор, завідувач кафедри комп'ютерних наук.

Зміст

Зміст……2
Передмова ……...3
Лабораторна робота № 1. Основи мови програмування C#.
Консольний режим програмування на C#. ……..4
1.1.Опорні теоретичні відомості 4
1.2. Приклади виконання завдань 37
1.3. Індивідуальні завдання 44
1.4. Зміст звіту: 54
1.5. Питання для самоперевірки 54
Лабораторна робота № 2. Організація циклів в програмах мовою C#.
……56
2.1. Опорні теоретичні відомості 56
2.2. Приклади виконання завдань 59
2.3. Індивідуальні завдання 65
2.4. Зміст звіту: 69
2.5. Питання для самоперевірки 70
Література ……70

©Шевченко І. М., 2019
© Національний університет
водного господарства та
природокористування, 2019

Національний університет
водного господарства
та природокористування

Передмова

Мета цих методичних вказівок – сформувати навички написання програм мовою програмування C# під платформу .NET, які надалі можуть бути застосовані у процесі обробки складних структур даних, у тому числі і динамічних, а також для створення Windows-додатків.

У методичних вказівках вміщено завдання, реалізуючи які читач зможе опанувати базові прийоми написання програм мовою програмування C#, реалізовувати алгоритмічні структури розгалуження та повторення для вирішення посталених задач.

Методичні вказівки можуть бути використані для опанування дисципліни «Програмування» студентами спеціальності 126 «Інформаційні системи і технології».

Вказівки містять лабораторні роботи за двома темами. В кожній лабораторній роботі описано мету роботи, опорні теоретичні відомості, приклади виконання завдань, завдання для самостійного індивідуального виконання, структуру змісту звіту та перелік контрольних запитань за матеріалами теми та отриманими результатами виконання поставлених задач.

Звіт до кожної виконаної лабораторної роботи виконується відповідно до вимог, що висуваються до оформлення документації у ВНЗ, і повинен містити:

- титульний аркуш;
- формулювання мети роботи;
- постановку задачі відповідного до варіанту завдання;
- результати роботи;
- висновки.

Національний університет
водного господарства
та природокористування

Лабораторна робота № 1. Основи мови програмування C#. Консольний режим програмування на C#.

Мета роботи: набути навички програмування консольних додатків в середовищі Visual Studio.

Послідовність виконання роботи:

1. Ознайомитись із теоретичними відомостями. (*Актуалізація опорних знань*).
2. Виконати програмування програм за поданими прикладами. Результат представити викладачу (Застосування набутих *знань*).
3. Виконати варіант самостійної роботи. (*Закріплення набутих знань*).
4. Оформити звіт на виконану роботу. (*Узагальнення та систематизація набутих знань*).
5. Захист звітів, відповіді на запитання.

Національний університет
водного господарства
та природокористування

1.1.Опорні теоретичні відомості

Короткі відомості про платформу .NET. Платформа .NET Framework визначає середовище для підтримки, створення й виконання платформонезалежних гетерогенних додатків. Особливостями даної платформи є незалежне від мови середовище виконання (Common Language Runtime, CLR) та бібліотека класів .NET.

Платформа .NET містить величезну бібліотеку класів, які можна використовувати при програмуванні будь-якою мовоюі .NET. Загальна структура бібліотеки приведена на рис. 1.1. Бібліотека має кілька рівнів. На самому нижньому знаходяться базові класи середовища, які використовуються при створенні будь-якої програми:

Національний університет
водного господарства
та природокористування

класи введення-виведення, обробки рядків, управління безпекою, графічного інтерфейсу, зберігання даних тощо.

Над цим шаром знаходиться набір класів, що дозволяє працювати з базами даних і XML. Класи самого верхнього рівня підтримують розробку розподілених додатків, а також веб-і Windows-додатків. Програма може використовувати класи будь-якого рівня. Бібліотека класів разом з CLR утворюють каркас (framework), тобто основу платформи.

Рис. 1.1. Основні компоненти платформи .NET.

Основними базовими поняттями .NET є:

NET компілятор, який переводить вихідний код в CIL (Common Intermediate Language);

збірка – двійковий файл, який підтримує версії, самоописує себе і обслуговується оточенням CLR (Common Language Runtime) Окрім CIL коду в збірці міститься різноманітна службова інформація, наприклад: інформація про зовнішні збірки (залежності), структура включених в неї типів тощо. Збірка може складатися з декількох модулів. У будь-якому випадку вона являє собою програму, яка готова для встановлення і не вимагає для цього ні додаткової інформації, ні складної послідовності дій. Кожна збірка має унікальне ім'я;

Національний університет
водного господарства
та природокористування

JIT компілятор – виконує перетворення коду MSIL в машинні команди по мірі виклику підпрограм (фрагменти). На виході утворюється так званий керований код, який потім виконується в середовищі CLR;

керований код – це такий, що містить метадані, які дозволяють CLR виконувати перевірку безпеки, контроль прав, контроль версій, збір мусору тощо.

На рис. 1.2 представлено схему компіляції в CLR.

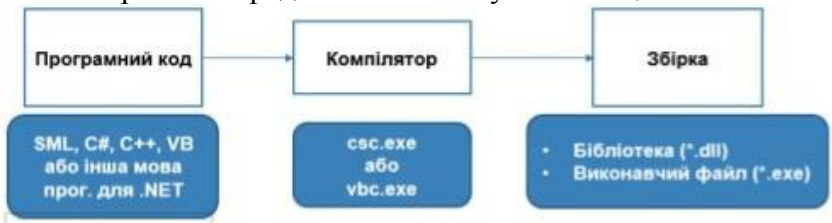


Рис. 1.2. Схема компіляції в CLR.

Під час виконання процедури трансляції вихідний текст програми (написаний на SML, C#, Visual Basic, C++ або іншому з безлічі інших мов програмування, який підтримується .NET) перетвориться компілятором в так звану збірку (Assembly) і зберігається у вигляді файлу бібліотеки, що динамічно приєднується (Dynamically Linked Library, DLL), або виконуваного файлу (Executable, EXE). Природно, що для кожного компілятора (будь то компілятор мови C#, csc.exe або Visual Basic, vbc.exe) середовищем часу виконання проводиться необхідне відображення використовуваних типів в типи CTS, а програмного коду – в код «абстрактної машини» .NET – MSIL (Microsoft Intermediate Language).

В результаті програмний проект формується у вигляді збірки – самодостатнього компонента для розгортання, тиражування і повторного використання. Збірка ідентифікується цифровим підписом автора і унікальним номером версії.

Національний університет
водного господарства
та природокористування

Отже, після компіляції створюється не виконуваний файл написаний у машинних кодах, а набір команд записаних проміжною мовою MSIL (Microsoft Intermediate Language). При запуску цього додатку в операційній системі його код написаний у MSIL транслюється JIT-компілятором (Justin- Time compiler – оперативний компілятор) в машинні коди, зрозумілі процесору комп'ютера (рис.1.3).

Припустимо, що компоненти проекту написані на двох мовах програмування мові C# та C++, яка характеризується можливістю написання потенційно небезпечного коду (зокрема, динамічного розподілу пам'яті). Вихідні тексти компонент проекту транслюються відповідно компіляторами з мов C# й C++ в уніфікований MSIL-код і зберігаються в файлах у вигляді збірок.

В ході компонування і виконання програмного проекту Just-In-Time (JIT) компілятор середовища CLR виконує проект з використанням в міру необхідності відтрансльованого проміжного коду збірок. Істотно, що потенційно небезпечний код мовою C++ принципово не може бути виконаний саме JIT-компілятором, але виконується за допомогою сервісів операційної системи. Відповідальність за працездатність програми і безпеку коду в цьому випадку лежить вже не на середовищі проектування і розробки програмного забезпечення .NET, а на програмістові-розробникові.

Як не важко здогадатися, такий додаток буде працювати лише в операційних системах з установленою платформою .NET Framework відповідної версії.

Національний університет
водного господарства
та природокористування

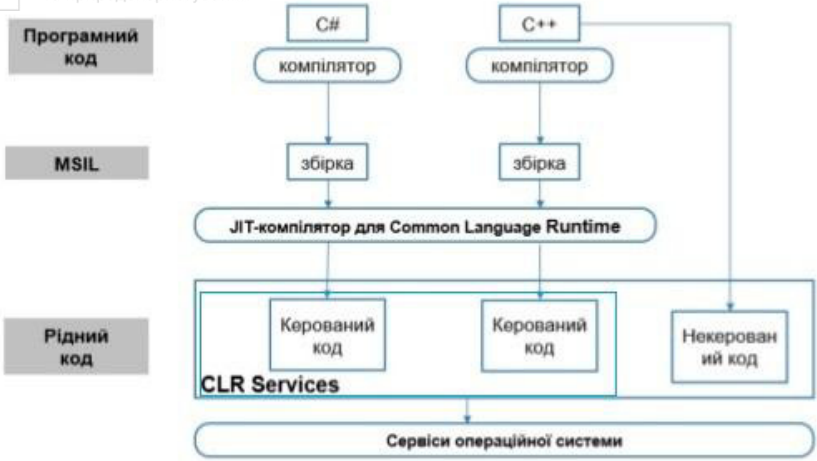


Рис. 1.3. Схема виконання в CLR.

Система типів .NET. Усі мови програмування оперують певними даними. Дані можуть бути різних типів. Можна, наприклад, зберігати у пам'яті комп'ютера число, літеру, рядок літер, таблицю чисел тощо. Платформа .NET використовує різні типи даних система типів Microsoft .NET, які утворють ієрархію із зростанням спільності знизу вгору, в якій явно виділяються дві великі групи типів, а саме, типи-посилання і типи-значення. Різниця між останніми визначається особливостями виклику в процедурах: по імені або за значенням (call-by-name, CBN) і по засланні (call-by-reference, CBR). Зауважимо також, що система типізації Microsoft .NET крім розвиненої ієрархії визначених типів дозволяє користувачеві створювати власні типи (як тіпи-посилання, так і типи-значення) на основі вже існуючих (рис 1.4).

Національний університет
водного господарства
та природокористування

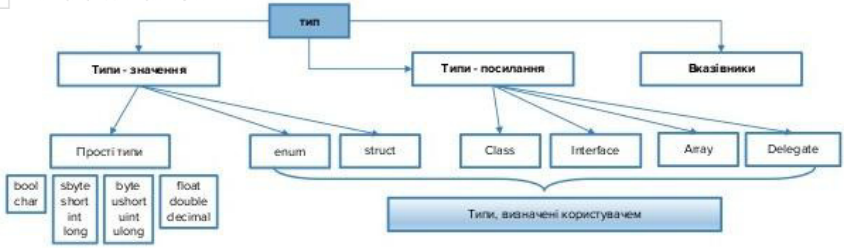


Рис.1.4. Універсальна система типів (UTS)

В ході компіляції програма на .NET-сумісній мові програмування трансформується відповідно до раніше заданої узагальненої специфікації мови Common Type System (CTS). Система типів CTS повністю описує всі типи даних, підтримувані середовищем виконання, визначає їх взаємозв'язкb і зберігає їх відображення в систему типів .NET (рис. 1.5).

CTS	VB. NET	C#	Managed Extensions for C++
System.Byte	Byte	byte	unsigned char
System.SByte	SByte	sbyte	signed char
System.Int16	Short	short	short
System.Int32	Integer	int	int or long
System.Int64	Long	long	__int64
System.Uint16	UShort	ushort	unsigned short
System.Uint32	UInteger	uint	unsigned int **або** unsigned long
System.Object	**Object**	**object**	**Object^**
System.Uint64	ULong	ulong	unsigned __int64
System.Single	Single	float	Float
System.Double	Double	double	Double
System.Char	Char	char	wchar_t^
System.String	String	string	String^
System.Decimal	Decimal	decimal	Decimal
System.Boolean	Boolean	bool	Bool

Рис. 1.5. Вбудовані типи даних CTS.

Система типізації Microsoft .NET являє собою частково впорядковану множину, яка на якісному рівні може розумітися як ISA-ієрархія (ISA походить від англійських слів "is a", які означають «є одним з»). Так, наприклад, висловлювання STUDENT ISA PERSON

Національний університет
водного господарства
та природокористування
означає, що тип STUDENT є підтипом типу PERSON (тут цілком доречна аналогія з множинами і цілком точна аналогія з доменами).

Середовище Visual Studio.NET. Середовище розробки Visual Studio.NET надає потужні і зручні засоби написання, коригування, компіляції, налагодження і запуску додатків, що використовують .NET-сумісні мови. Корпорація Microsoft включила в платформу засоби розробки для чотирьох мов: C#, VB.NET, C++ і J#.

Платформа .NET є відкритим середовищем. Це означає, що компілятори для неї можуть поставлятися і сторонніми розробниками. До теперішнього часу розроблені десятки компіляторів для .NET, наприклад, Ada, COBOL, Delphi, Eiffel, Lisp, Oberon, Perl, Python.

Все .NET сумісні мови повинні відповідати вимогам загальномовної специфікації (Common Language Specification, CLS), в якій описується набір загальних для всіх мов характеристик. Це дозволяє використовувати для розробки програми кілька мов програмування і вести повноцінне міжмовне налагодження. Всі програми незалежно від мови використовують ті ж базові класи бібліотеки .NET.

Додаток в процесі розробки називається проектом. Проект об’єднує все необхідне для створення програми: файли, папки, посилання і інші ресурси. Середовище Visual Studio.NET дозволяє створювати проекти різних типів, наприклад:

- Windows-додаток використовує елементи інтерфейсу Windows, включаючи форми, кнопки, прапорці та ін .;
- консольний додаток виконує висновок на екран «на консоль», тобто в вікно командного процесора;
- бібліотека класів об'єднує класи, які призначені для використання в інших додатках;

Національний університет
водного господарства
та природокористування

– веб-додаток – це додаток, доступ до якого здійснюється через браузер (наприклад, Internet Explorer) і яке за запитом формує веб-сторінку і відправляє її клієнту через мережу;
– веб-сервіс - компонент, методи якого можуть викликатися через Інтернет.

Консольні додатки. Середовище Visual Studio.NET працює на платформі Windows і орієнтована на створення Windows- і веб-додатків, однак розробники передбачили роботу і з консольними додатками. При запуску консольного застосування операційна система створює так зване консольне вікно, через яке йде введення-виведення програм. Зовні це нагадує роботу в операційній системі в режимі командного рядка, коли введення-виведення є потоком символів.

Консольні додатки найкращим чином підходять для вивчення мови, так як в них не використовується безліч стандартних об'єктів, необхідних для створення графічного інтерфейсу.

Створення консольного додатку.

Вікно середовища MS Visual Studio .NET (рис. 1.5) складається з меню (1), стандартної панелі інструментів (2), списку раніш виконуваних проектів (3), кнопок для відкриття та створення проектів (4), панелі для виведення повідомлень (5).

Національний університет
водного господарства
та природокористування

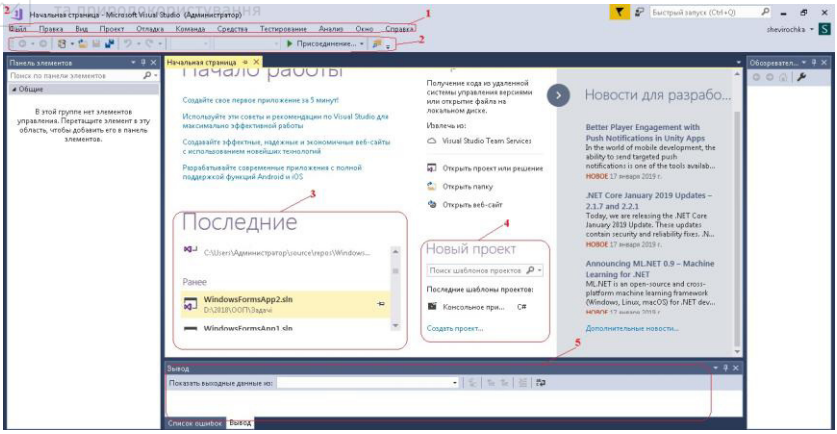


Рис.1.5. Вікно середовища MS Visual Studio .NET.

Створимо новий проект. Для цього виконаємо команду: [Файл→Создать→Проект…] (рис. 1.6).

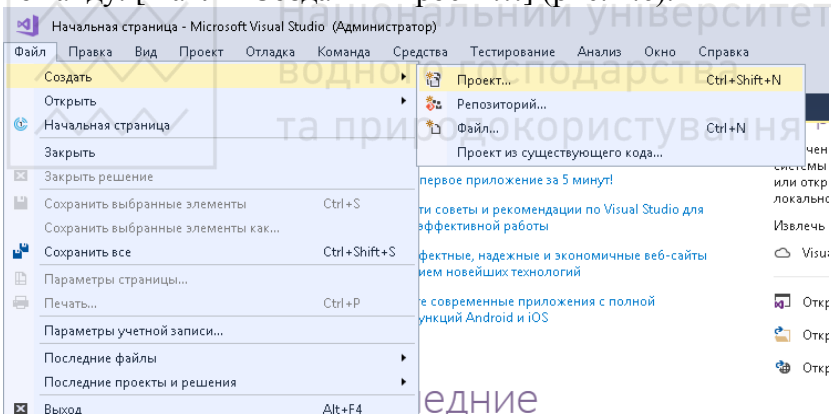


Рис. 1.6. Створення нового проекту

Ліворуч у вікні Создание проекта оберіть Visual C#, праворуч – Консольное приложение (.NET Framework) (рис. 1.7).

Національний університет
водного господарства
та природокористування

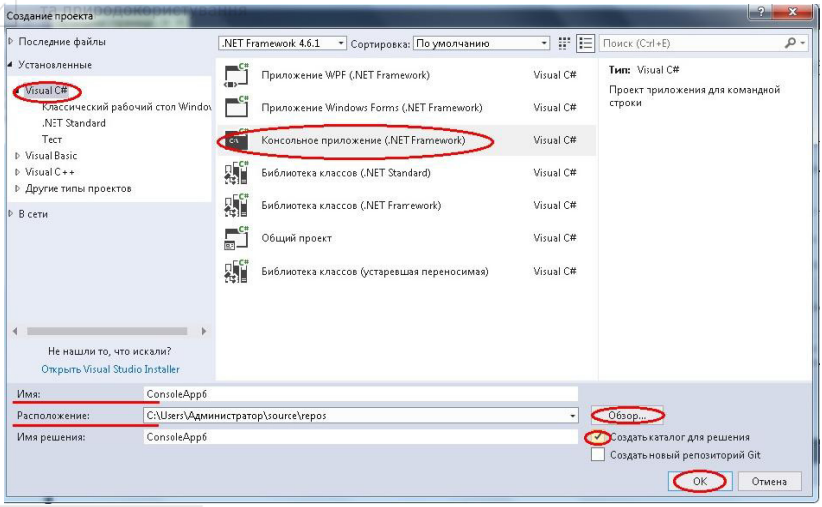


Рис. 1.7. Створення консольного додатку C#.

В полі [Имя] можна надати ім’я проекту (система за замовчуванням надає ім’я ConsoleAppN, де N – номер додатку).

В полі [Расположение] за допомогою кнопки [Обзор…] можна задати шлях, де потрібно розмістити проект, якщо не місце, запропоноване середовищем.

Встановити прапорець [Создать каталог для решения] та натиснути [ОК].

В результаті вищеописаних дії відкриється файл Program.cs, де автоматично буде написаний початковий код програми, який в подальшому може змінити користувач.

Національний університет
водного господарства
та природокористування

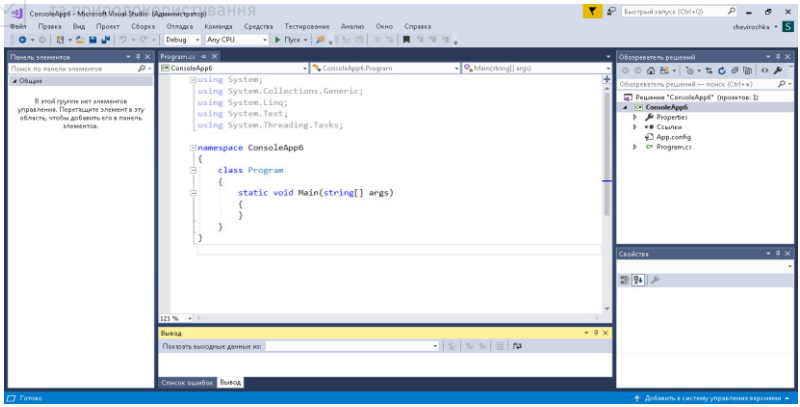


Рис. 1.8. Вікно консольного додатку Program.cs

Заготовка консольної програми. Розглянемо структуру програми мовою C#, яку представлено на рис. 1.9.

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace ConsoleApp6
8   {
9       class Program
10      {
11          static void Main(string[] args)
12          {
13          }
14      }
15  }
16
```

Рис. 1.9. Заготовка консольної програми.

Перші п’ять рядків містять команди для підключення модулів System, System.Collections.Generic,

Національний університет
водного господарства
та природокористування

System.Linq, System.Text, System.Threading.Tasks. Модуль – це спеціальна бібліотека з нформаією про об'єкти, які використовуються у програмі. Підключені в цій програмі модулі входять до системних і є частиною операційної системи. Програміст може сторювати власні модулі та підключати їх до програми директивою ***using***. Модуль ***System*** дозволяє використовувати імена стандартних класів з простору імен ***System*** безпосередньо без зазначення простору.

У сьомому рядку ключове слово ***namespace*** створює для проекту власний простір імен, який названий за замовчуванням ***ConsoleApp6***. Це зроблено для того, щоб можна було давати програмним об'єктам імена, не піклуючись про те, що вони можуть збігтися з іменами в інших просторах імен.

Символи «{» та «}» починають і закінчують блок команд, їх називають операторними дужками. Призначенням операторних дужок є обмежування блоку команд.

У мові *C#* програма являє собою сукупність взаємодіючих між собою класів. У програмі автоматично створений клас з ім'ям ***Program***. Усередині класу (рядок 11) – метод ***Main ()***, який зобов'язаний бути присутнім в будь-якому додатку – з нього починається виконання програми. ***Main*** – це стандартна назва основної програми (ще говорять – точка входу). У круглих дужках вказуються параметри, які передаються в програму з командного рядку, якщо вони є. Слово ***void*** вказує, що програма не повертає результат обчислень в програму, яка її викликала. Слово ***static*** є обов'язковим. Код програми записується між дужками в рядках 12 і 13.

Введення та виведення інформації в програму. Введення та виведення інформації здійснюється за допомогою команд:

Національний університет
водного господарства
та природокористування

Write () – виведення інформації в консоль;

WriteLine () – виведення інформації в консоль та перехід на новий рядок;

Read () – читання одного символу з консолі (клавіатури);

ReadLine () – читання одного символу з консолі та перехід на новий рядок.

Створимо програму, яка виводить повідомлення на екран.

Приклад 1.1. Виведення на екран повідомлення "ПРИВІТ! Вивчаємо мову C#!".

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Hello
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("ПРИВІТ! Вивчаємо мову C#!");
      Console.Read();
    }
  }
}
```

Національний університет
водного господарства
та природокористування

Запустити на виконання програму можна одним з способів:

– натиснути клавішу F5;
– натиснути кнопку
– виконати команду [Отладка→Начать отладку]

Результат роботи програми відобразиться у вікні консолі (рис. 1.10).

Національний університет
водного господарства
та природокористування

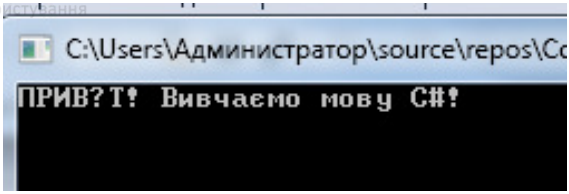


Рис. 1.10. Вікно виведення результату роботи консольного додатку.

Для збереження проекту потрібно виконати команду [Файл→Сохраніть все…]. Для проекту буде створено окрему папку, в яку буде збережено всі файли проекту.

Основи мови програмування C#. Всі конструкції мови будуються за допомогою символяв, які складають алфавіт. У C# використовується кодування символів Unicode. Кодування Unicode дозволяє представити символи всіх існуючих алфавітів одночасно, що докорінно покращує переносимість текстів. Кожному символу відповідає свій унікальний код. Природно, що при цьому для зберігання кожного символу потрібно більше пам'яті. Перші 128 Unicode-символів відповідають першій частині кодової таблиці ANSI.

Алфавіт C# включає:

- літери (латинські та національних алфавітів) і символ підкреслення (_), який вживається поряд з буквами;
- цифри;
- спеціальні символи, наприклад +, *, {, & тощо;
- пробільні символи (пробіл і символи табуляції);
- символи переведення рядка.

З символів складаються більші будівельні блоки: лексеми, директиви препроцесора і коментарі.

Лексема (token) - це мінімальна одиниця мови, що має самостійний сенс. Існують наступні види лексем:

- імена (ідентифікатори);

Національний університет
водного господарства
та природокористування

– ключові слова;
– знаки операцій;
– роздільники;
– літерали (константи).

Лексеми мови програмування аналогічні словами природної мови.

Директиви препроцесора прийшли в C# з його попередника – мови C++. Препроцесором називається попередня стадія компіляції, на якій формується остаточний вигляд початкового тексту програми. Наприклад, за допомогою директив (інструкцій, команд) препроцесора можна включити або виключити з процесу компіляції фрагменти коду.

Коментарі призначені для запису пояснень до програми і формування документації. З лексем складаються вирази й оператори.

Вираз задає правило обчислення деякого значення. Наприклад, вираз *a* + *b* задає правило обчислення суми двох величин.

Оператор задає закінчений опис деякої дії, даних або елемента програми. Наприклад: int a; Це оператор опису цілочисельний змінної a.

Компілятор виконує переклад програми в виконуваний файл на мові IL за дві послідовні фази: лексичного і синтаксичного аналізу. При лексичному аналізі з потоку символів, які складають вихідний текст програми, виділяються лексеми (за іншими лексемами, розділювачами, пробільним символами). На етапі синтаксичного аналізу програма переводиться в виконуваний код. Крім того, компілятор формує повідомлення про синтаксичних помилках.

Імена в програмах переслідують ту ж мету, що і імена в світі людей, – щоб звертатися до програмних об'єктів і розрізняти їх, тобто ідентифікувати. Тому імена

Національний університет
водного господарства
та природокористування

також називають ідентифікаторами. Мова *C#* є чутливою до регістра. Це означає, що два ідентифікатора F1 та f1 різні. Всі ключові слова мови *C#* пишуться в нижньому регістрі. Основні правила створення імен змінних, констант та інших об'єктів програми мовою *C#:*

1. першим символом може бути будь-яка літера латинського алфавіту, знак підкреслення «_», символ «@»;
2. наступними символами імені можуть бути довільні символи латиниці, знак підкреслення, цифри. символи можуть бути великими та малими;
3. для імені не можна використовувати ключові слова мови програмування.

Ключові слова – це зарезервовані ідентифікатори, які мають спеціальне значення для компілятора. Їх можна використовувати тільки в тому сенсі, в якому вони визначені. Список ключових слів C # наведено в таблиці 1.1.

Національний університет
водного господарства
та природокористування

Таблиця 1.1.

Ключові слова C#

break	checked	default	ehum
finally	goto	interface	namespace
out	public	sealed	string
true	unchecked	void.bool	char
decimal	else	false	foreach
int	long	operator	protected
sbyte	static	throw	ulong
virtual	base	catch	continue
double	extern	for	in
lock	object	private	return
stackalloc	this	uint	using as
case	const	do	explicit
float	implicit	is	null
params	ref	sizeof	switch

Національний університет
водного господарства
та природокористування

typeof	ushort	while	abstract
byte	class	delegate	event
fixed	if	internal	new
override	readonly	short	struct
try	unsafe	volatile	

Знак операції – це один або більше символів, що визначають дію над операндами. Усередині знака операції пропуски не допускаються. Наприклад, у виразі *a* += *b* знак += є знаком операції, а *a* і *b* - операндами. Символи, що є ом операцій, можуть бути як спеціальними, наприклад, &&, | і <, так і літерними, такими як as або new.

Операції поділяються на унарні, бінарні і тернарного за кількістю операндів, що беруть в них участь. Один і той же знак може інтерпретуватися по-різному в залежності від контексту. Всі знаки операцій, за винятком [], () і ? :, являють собою окремі лексеми.

Роздільники використовуються для розділення або, навпаки, групування елементів. Приклади роздільників: дужки, крапка, кома. Нижче перераховані всі знаки операцій і роздільники, що використовуються в C#:

{ } [ ] ( ). ,:; + - * /% & | ^! ~ =
<>? ++ - && << >> ==! = <=> = + == * = / =% =
& = | = ^ = << = >> = ->

Літералами, або константами, називають незмінні величини. У C# є логічні (true, false), цілі, дійсні, символьні і рядкові константи, а також константа ***null***. Компілятор, виділивши константу в якості лексеми, відносить її до одного з типів даних по її зовнішньому вигляду.

Символьною константою може бути будь-який символ в кодуванні Unicode. Символьні константи записуються в одній з чотирьох форм:

Національний університет
водного господарства
та природокористування

– «звичайний» символ, який має графічне представлення (крім апострофа і символу перекладу рядка) – 'A', 'ю', '*';
– керуюча послідовність - '\ 0', '\ n';
– символ у вигляді шістнадцятирічного коду – '\xF', '\x74';
– символ у вигляді escape-послідовності Unicode – '\uA81B'.

Керуючою послідовністю, або простий escape-послідовністю, називають певний символ, що передує зворотною косою межею. Керуюча послідовність інтерпретується як одиночний символ і використовується для подання:

– кодів, які не мають графічного зображення (наприклад, \n - перехід в початок наступного рядка);
– символів, що мають спеціальне значення в строкових і символьних літералах, наприклад, апострофа '.

У таблиці 1.2 наведені допустимі значення послідовностей. Якщо безпосередньо за зворотною косою рискою слідує символ, не передбачений таблицею, виникає помилка компіляції.

Таблиця 1.2.

Допустимі значення escape-послідовністей

\a-звуковой сигнал	\b-возврат на шаг
\f-перевод страницы (формата)	\n-перевод строки
\r-возврат каретки	\t-горизонтальная табуляция
\v-вертикальная табуляция	\\ - обратная косая черта
\’-апостроф	\”-кавычка
\0-нуль-символ	

Керуючі послідовності можуть використовуватися і в строкових константах. Наприклад, якщо потрібно вивести кілька рядків їх об'єднують в один літерал,

Національний університет
водного господарства
та природокористування
відокремивши один рядок від іншої керуючої послідовністю \n:

"Ніхто не задоволений своєю \n зовнішністю, але кожен задоволений \n своїм розумом"

Якщо ж всередині рядка потрібно використовувати лапки, їх випереджають косою рискою:

"Видавничий дім \" Пітер \""

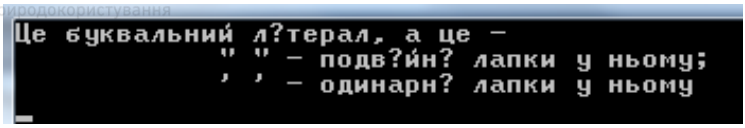
Буквальні літерали. Цікавий ще один нюанс, який відрізняє текстові літерали мов C++ та C#. Якщо перед текстовим літералом стоїть знак @, то такий літерал називається буквальним і при зображенні на екрані виглядає дослівно так само, як у подвійних лапках. Таким чином зникає потреба у використанні знаків табуляції, нового рядку тощо. Єдиний виняток проти «буквальності» – сам знак подвійних лапок, якщо він зустрічається у літералі, має бути подвоєним. Нижче наведений приклад використання буквального (virbatim) літералу:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Virbatim_Literal
{
 class Program
 {
  static void Main()
  {
   Console.WriteLine(@"Це буквальний літерал, а це -
   "" "" - подвійні лапки у ньому;
   ' ' - одинарні лапки у ньому ");
   Console.Read();
  }
 }
}
```

Після запуску побачимо на екрані наступний текст:

Національний університет
водного господарства
та природокористування



Коментарі. У *C#* існує 3 види коментарів – однорядкові, багаторядкові і XML коментарі. Однорядковий коментар починається з // і закінчується символом переходу на новий рядок. Складний коментар полягає в / * * / і може займати частину рядка, цілий рядок або кілька рядків. Формування документації в форматі XML використовують коментарі, що починаються з ///. Компілятор витягує ці коментарі з програми, перевіряє на відповідність правилам і записує в окремий файл.

Типи даних C#. Мова C# – це мова жорстокої типізації. Це означає, що кожний об'єкт у програмі має відноситись до одного з визначених типів. Тип даних однозначно визначає:

– внутрішнє подання даних;
– безліч їх значень;
– допустимі дії над даними (операції і функції).

Всі типи даних, що використовуються в C#, діляться на 2 категорії: типи-значень (value types) та типи-посилання (reference types). Про відмінності між ними говорилось раніше, а зараз розглянемо основні вбудовані типи мови C#. Вони представлені у табл. 1.3.

Таблиця 1.3.

Вбудовані типи мови C#

Тип даних	**Ключове слово**	**Тип .NET**	**Диапазон**	**Опис**	**Розмір, біт**
Логічний тип	bool	Boolean	true, false		
Цілі типи	sbyte	SByte	-128..127	зі знаком	8
	byte	Byte	0..255	без	8

Національний університет
водного господарства
та природокористування

				знаку	
	short	Int16	-32768..32767	зі знаком	16
	ushort	UInt16	0..65535	без знаку	16
	int	Int32	$-2*10^9..2*10^9$	зі знаком	32
	uint	UInt32	$0..4*10^9$	без знаку	32
	long	Int64	$-9*10^{18}..9*10^{18}$	зі знаком	64
	ulong	UInt64	$0..18*10^{18}$	без знаку	64
Символьний тип	char	Char	U+0000..U+ffff	Unicode-символ	16
Дійсні типи	float	Single	$1.5*10^{-45}..3.4*10^{38}$	7 цифр	32
	double	Double	$5.0*10^{-324}..1.7*10^{308}$	15-16 цифр	64
Фінансовий тип	decimal	Decimal	$1.0*10^{-28}..7.9*10^{28}$	28-29 цифр	128
Рядковий тип	string	String	Довжина обмежена об’ємом доступної пам’яті	Рядок з Unicode-символів	
Тип object	object	Object	Можна зберігати що завгодно	Базовий для всіх інших типів	

Для того, щоб визначити змінну одного із стандартних типів досить вказати її тип та ідентифікатор.

Національний університет
водного господарства
та природокористування

Можлива її ініціалізація в момент визначення константним значенням або значення виразу.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Variable_Declarations
{
 class Program
 {
  static void Main()
  {
   double f = 1.5F;
   // визначення змінної f з ініціалізацією
   char c;  // просто визначення змінної c
   int i = 0;
   bool b = false;
   decimal d = 1.55555555555555555555555M;
   // визначення змінної x з динамічною ініціалізацією
   double x = Math.Sin(Math.PI / 3);
   c = 'A'; // надання змінної c значення
   Console.WriteLine("f = {0} i = {1} x = {2} d ={3}
                     c={4}",f, i, x, d, c);
   Console.WriteLine(b.ToString());
   Console.Read();
  }
 }
}
```

Розглянемо команду виведення на консоль:

```
Console.WriteLine("f = {0} i = {1} x = {2}  d ={3} c={4}",f,i,x,d,c);
```

Ця команда демонструє можливість форматування виведення інформації на консоль, тобто представлення її в зручному для сприйняття вигляді. Для цього у текстову константу `"f = {0} i = {1} x = {2}  d ={3}`

Національний університет
водного господарства
та природокористування

`c={4}",`– аргумент методу Console.WriteLine – треба помістити так звані плейсхолдери з номером потрібного елементу списку виводу. На місці кожного з таких плейсхолдерів на екрані з'явиться відповідне значення. Більше того, можливо використання форматів виводу. Загальний вигляд плейсхолдеру такий {n:fk}. Тут n означає порядковий номер елементу у списку виводу (нумерація починається з нуля), f – це символ специфікації формату, число k задає точність. Основні символи специфікацій формату такі:

- F або f – для виводу дійсних значень у форматі з фіксованою точністю;
- E або e – для виводу дійсних значень у експоненціальному форматі;
- G або g – загальний формат для виводу дійсних значень або у форматі з фіксованою точністю або у експоненціальному форматі;
- N або n – формат для виводу дійсних значень з відокремленням трійок розрядів пробілами;
- C або c – грошовий формат;
- X або x – шістнадцятковий формат для виводу цілих типів.

Приведення типів. При обчисленні виразів, які містять операнди різних типів, всі вони приводяться (звісно, якщо типи сумісні між собою) до найбільш широкого типу. Таке перетворення виконується неявним чином при дотриманні низки правил «просування по сходинках типів». При звуженні потужності типу завжди потрібне явне приведення типу. Така операція записується інструкцією:

```
(тип_до_якого_приводимо_вираз)вираз;
```

Правила просування є такими:

- якщо один із операндів має тип decimal, то і другий буде приводитись до такого типу (але якщо другий

Національний університет
водного господарства
та природокористування

операнд мав тип float або double , результат буде помилковим);

- якщо один із операндів має тип double, то і другий буде приводитись до такого типу double;
- якщо один із операндів має тип float, то і другий буде приводитись до типу float;
- якщо один із операндів має тип ulong, то і другий буде приводитись до типу ulong (але якщо другий операнд мав цілий знаковий тип, результат буде помилковим);
- якщо один із операндів має тип long, то і другий буде приводитись до типу long;
- якщо один із операндів має тип uint, а другий має тип sbyte, short або int , то обидва операнди будуть приведені до типу long;
- якщо один із операндів має тип uint, то і другий буде приводитись до типу uint;
- інакше обидва операнди перетворюються до типу `int`

Операції і вирази. Вираз – це правило обчислення значення. У виразу беруть участь опе¬ранди, об'єднані знаками операцій. Операндами найпростішого виразу можуть бути константи, змінні і виклики функцій.

Таблиця 1.4.

Операції C#

Категорія	**Знак операції**	**Назва**
Первинні	**.**	Доступ до елементу
	X()	Виклик методу або делегата
	X[ ]	Доступ до елементу
	X++	Постфіксний інкремент
	new	Виділення пам'яті

Національний університет
водного господарства
та природокористування

	X--	Постфіксний декремент
	typeof	Отримання типу
	checked	Код, який перевіряється
	unchecked	Код, який не перевіряється
Унарні	**+**	Унарний плюс
	-	Унарний минус (арифметичне заперечення)
	!	Логічне заперечення
	~	Поразрядне заперечення
	++X	Префіксный інкремент
	--X	Префіксный декремент
	(тип)x	Перетворення типу
Мультиплікативні (Типу множення)	*****	Множення
	/	Ділення
	%	Остача від ділення
Аддитивні (типу додавання)	**+**	Додавання
	-	Віднімання
Зсуву	**<<**	Зсув ліворуч
	>>	Зсув праворуч
Відношення і перевірка типу	**<**	Менше
	>	Більше
	<=	Менше або дорівнює
	>=	Більше або дорівнює
	is	Перевірка чи належить типу
	as	Приведення типу

Національний університет
водного господарства
та природокористування

Перевірка на рівність	**==**	Дорівнює
	!=	Не дорівнює
Порозрядні логічні	**&**	Порозрядне І (кон'юнкція)
	^	Порозрядне виключне АБО
	|	Порозрядне АБО (диз'юнкція)
Условные логические	**&&**	Логічне І
	||	Логічне АБО
Тернарна операція	**?:**	Умовна операція
Присвоювання	**=**	Присвоювання
	***=**	Множення з присвоюванням
	/=	Ділення з присвоюванням
	%=	Остача від ділення з присвоюванням
	+=	Додавання з присвоюванням
	-=	Віднімання з присвоюванням
	<<=	Зсув ліворуч з присвоюванням
	>>=	Зсув праворуч з присвоюванням
	&=	Порозрядне І з присвоюванням
	^=	Порозрядне виключне АБО з присвоюванням
	|=	Порозрядне АБО з присвоюванням

Національний університет
водного господарства
та природокористування

Національний університет
водного господарства
та природокористування

Операції у виразі виконуються в певному порядку відповідно до пріоритетів, як і в математиці. У наведеній вище таблиці 1.4 операції розташовані за спаданням пріоритетів.

Результат обчислення виразу характеризується значенням і типом.

Форматоване введення та виведення інформації. Будь-яка програма при введенні вихідних даних і виведення результатів взаємодіє із зовнішніми пристроями. Сукупність стандартних пристроїв введення та виведення, тобто клавіатури і екрану, називається консоллю.

У мові C#, як і в багатьох інших, немає операторів введення і виведення. Замість них для обміну з зовнішніми пристроями застосовуються стандартні об'єкти. Для роботи з консоллю в C# застосовується клас ***Console***, визначений у просторі імен ***System***. Методи цього класу ***Write*** і ***WriteLine*** вже використовувалися в програмах.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp6
{
  class Class1
  {
    static void Main(string[] args)
    {
      int i = 3;
      double y = 4.12;
      decimal d = 600m;
      string s = "Вася";
      Console.WriteLine("i = " + i); //строка 1
      Console.WriteLine("y = {0} \n d = {1}", y, d);
```

Національний університет
водного господарства
та природокористування

```
        Console.WriteLine("s = " + s);
        Console.Read();
      }
    }
}
```

Перетворення в рядок визначено у всіх стандартних класах C # – для цього служить метод ***ToString ().*** Викликається він явним способом:

`Console.WriteLine ( "i =" + i.ToString ());` // що аналогічно рядку 1 в прикладі.

В рядку :

`Console.WriteLine("y = {0} \n d = {1}", y, d);`

застосовується перевантажений метод WriteLine (), що має параметри в фігурних дужках, а також керуючі послідовності. Параметри нумеруються з нуля, перед виведенням вони замінюються значеннями змінних в списку виведення. Наприклад, нульовий параметр – y, перший – d і т.д.

Введення з клавіатури. Перетворення рядків у числа

У класі ***Console*** визначені методи введення рядка або символу, але немає методів, які дозволяють зчитувати з клавіатури числа. Введення чисел відбувається в 2 етапи:

1. Числа вводяться з клавіатури як рядкові константи
2. Виконується перетворення з рядка в змінну відповідного типу.

Перетворення можна виконати, використовуючи:

- клас ***Convert***, що має методи перетворення в будь-який стандартний тип C# – ***Convert.ToInt32*** (строкова константа), ***Convert.ToDouble*** (строкова константа) тощо
- метод ***Parse*** (рядкова константа) – метод, який існує у будь-якого стандартного типу – ***double.Parse (s)***

Національний університет
водного господарства
та природокористування

Приклад введення даних з клавіатури

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
 class Class1
 {
  static void Main ()
   {
   // ----------------------------------
   Console.WriteLine("Введіть рядок");
   string s = Console.ReadLine();
   Console.WriteLine("s =" + s);
   // ----------------------------------------------
   Console.WriteLine("Введіть символ");
   char c = (char)Console.Read();
// зчитування залишку рядка, немає очищення буфера
   Console.ReadLine();
   Console.WriteLine("c =" + c);
   // ----------------------------------------------
   string buf;
   Console.WriteLine("Введіть ціле число");
   buf = Console.ReadLine();
   int i = Convert.ToInt32(buf);
   Console.WriteLine(i);
   // -----------------------------------------------
   Console.WriteLine("Введіть дійсне число");
   buf = Console.ReadLine();
   double x = Convert.ToDouble(buf);
   Console.WriteLine(x);
   // -----------------------------------------------
   Console.WriteLine("Введіть дійсне число");
   buf = Console.ReadLine();
   double y = double.Parse(buf);
   Console.WriteLine(y);
   // -----------------------------------------------
   Console.WriteLine("Введіть дійсне число");
```

Національний університет
водного господарства
та природокористування

```
    buf = Console.ReadLine();
    decimal z = decimal.Parse(buf);
    Console.WriteLine(z);
   }
  }
}
```

Математичні функції. Клас Math. Всі математичні функції реалізовані як методи класу Math, основні з яких представлені в таблиці 1.5.

Таблиця 1.5.

Основны поля і методи класу Math

Имя метода	Описание	Пояснения
Abs	Модуль	|x|, записується як Abs(x)
Acos	Арккосинус	Acos(double x), кут задається в радіанах
Asin	Арксинус	Asin(double x)
Atan	Арктангенс	Atan(double x)
Atan2	Арктангенс	Atan2(double x,double y) –кут, тангенс якого є результат ділення y на x
BigMul	Добуток	BigMul(int x,int y)
Celling	Округлення до більшого цілого	Celling(double x)
Cos	Косинус	Cos(double x)
Cosh	Гіперболічний косинус	Cosh(double x)
DivRem	Розподіл і залишок	DivRem(x,y,rem)
E	Число e	2,71828182845905
Exp	Експонента	Exp(x)
Floor	Округлення до меншого цілого	Floor(double x)

Національний університет
водного господарства
та природокористування

IEEERemainder	Остача від ділення	IEEERemainder(double x, double y)
Log	Натуральний логарифм	Log(x)
Log10	Десятковий логарифм	Log10(x)
Max	Максимум з 2-х чисел	Max(x,y)
Min	Мінімум з 2-х чисел	Min(x,y)
PI	Значення числа Пі	3,14159265358979
Pow	Піднесення до ступіня	Pow(x,y) – x в степени y
Round	Округлення	Round(3.1)=3 Round(3.8)=4
Sign	Знак числа	
Sin	Синус	Sin(double x)
Sinh	Гіперболічний синус	Sinh(double x)
Sqrt	Квадратний корінь	Sqrt(x)
Tan	Тангенс	Tan(double x)
Tanh	Гіперболічний тангенс	Tanh(double x)

Основними інструкціями керування мови C# є оператори розгалуження та цикли.

Розгалуження у мові C#. Для реалізації галужень (одного із трьох основних елементів блок-схем програм) у мові C# використовуються конструкції ***if*** , ***switch*** та тернарна умовна операція.

Умовна операція має таку синтаксичну форму:

```
<вираз_1> ? < вираз_2> : < вираз_3>
```

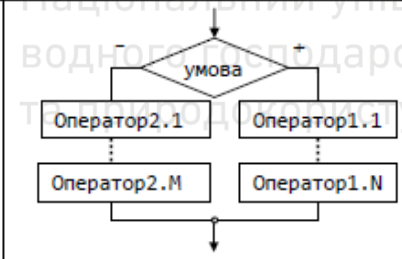
Національний університет
водного господарства
та природокористування

Перший вираз повинен мати тип ***bool***. Якщо його значення рівне ***true***, то обчислюється значення другого виразу, яке і визначає всю тернарну операцію. Якщо ж значення першого виразу ***false***, то обчислюється значення третього виразу, і його значення стає значенням всього тернарного виразу. Наприклад, у наступному фрагменті коду обчислюється максимум двох змінних: цілого типу:

```
int x = 3, y = 4;
int z = (x > y) ? x : y;
Console.WriteLine("найбільше з чисел
{0} та {1} є {2}",x, y, z);
```

Конструкція ***if*** може бути використана у двох формах. Повна форма:

Програмна структура	Аналог на мові блок-схем	Приклад

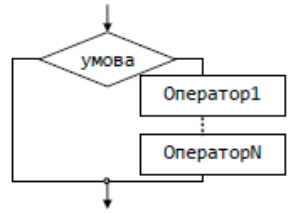
if (<умова>)
{
<оператор1.1>;
. . . . . .
<оператор1.N>;
}
else
{
<оператор2.1>;
. . . . . .
<оператор2.M>;
}
		if(x>y)
		{
		max=x;
		min=y;
		}
		else
		{
		max=y;
		min=x;
		}

<Умовою> є вираз з результатом типу ***bool***. Порядок виконання цієї конструкції такий: якщо вираз набуває значення ***true*** («Істина»), то виконується перша з інструкцій, якщо ж вираз набуває значення є ***false*** («Хибність»), то друга.

Частина ***else*** не є обов'язковою і може бути пропущена. Тоді при істинності умови єдина інструкція ***if*** виконується, а при хибності – управління виконанням програми передається наступній конструкції програми. Скорочена форма:

Національний університет
водного господарства
та природокористування

Програмна структура	Аналог на мові блок-схем	Приклад

if (<умова>)
{
<оператор1>;
. . . . . .
<операторN>;
}
		if(x>0)
		{
		z=1/x;
		l=y/x;
		}

Застосуємо конструкцію ***if*** для визначення найбільшого значення двох змінних цілого типу:

```
int x = 3, y = 4;
if (x > y)
Console.WriteLine("найбільше з чисел {0} та
{1} є  {2}",x, y, x);
else
Console.WriteLine("найбільше з чисел {0} та
{1} є  {2}",x, y, y);
```

В цьому фрагменті використовується повна форма конструкції ***if.***

Для більш зручної та компактної реалізації багатоальтернативних галужень в мові *C#* використовується конструкція ***switch***. Вона має такий синтаксис:

switch (<селектор вибору>)
{
case <конст1>: <опер1>; break;
case <конст2>: <опер2>; break;
case <конст3>: <опер3>; break;
……
case <констN>: <операторN>; break;
default: <операторN+1>; break;
}

Тут `<селектор вибору>` повинен мати цілий тип (або тип char), кожна з констант є унікальною і сумісною за типом із `<виразом>`. Після обчислення

Національний університет
водного господарства
та природокористування

<вираз> виконується та гілка конструкції ***switch***, яка містить константу рівну значенню <виразу>. Якщо жодна з констант не співпадає із значенням <виразу>, виконуються інструкції гілки `default`. Втім, остання не є обов'язковою і в разі її відсутності у випадку, коли жодна з констант не співпадає із значенням <виразу>, жодна інструкція конструкції ***switch*** не виконується взагалі.

1.2. Приклади виконання завдань

Приклад 1.2.1. Обчислити значення арифметичного виразу:

$$P = \frac{sin^3 x + ln(2y + 3x)}{t^e + \sqrt{x}}$$

Значення змінних вводиться з клавіатури.

Розвя'язок:

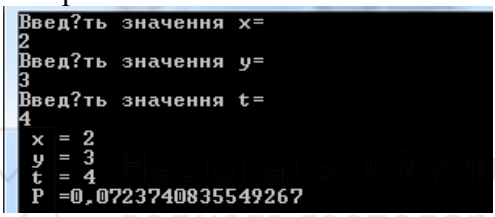
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Variable_Declarations
{
  class Program
  {
    static void Main()
    {
      double P, x, y, t;
      string buf;
      Console.WriteLine("Введіть значення x=");
      buf = Console.ReadLine();
      x = Convert.ToDouble(buf);
      Console.WriteLine("Введіть значення y=");
      buf = Console.ReadLine();
      y= Convert.ToDouble(buf);
```

Національний університет
водного господарства
та природокористування

```
        Console.WriteLine("Введіть значення t=");
        buf = Console.ReadLine();
        t = Convert.ToDouble(buf);
        P = (Math.Pow(Math.Sin(x), 3) + Math.Log(2 * y +
        3 *x)) / (Math.Pow(t, Math.E) + Math.Sqrt(x));
        Console.WriteLine(" x = {0} \n y = {1}\n t = {2}
                          \n P ={3}", x, y, t, P);
        Console.Read();
      }
    }
}
```

Результат роботи:



Приклад 1.2.2. Обчислити значення функції.

$$y = cos^2 b + b\,cos^4 a^2,\ a = \sqrt[3]{|b+c|},\ b = x + c^2.$$

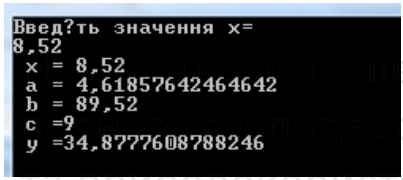
При $x = 8{,}52$ та $c = 9$. Значення x вводиться з клавіатури, значення іншого параметру задається як константа.

Розвя'язок:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Variable_Declarations
{
 class Program
 {
  static void Main()
  {
   double y, x, a, b, c=9;
```

```
    string buf;
    Console.WriteLine("Введіть значення x=");
    buf = Console.ReadLine();
    x = Convert.ToDouble(buf);
    b = x + c * c;
    a = Math.Pow(Math.Abs(b + c), 1.0 / 3);
    y = Math.Pow(Math.Cos(b), 2) + b *
        Math.Pow(Math.Cos(a * a), 4);
    Console.WriteLine(" x = {0} \n a = {1}\n b = {2} \n
                      c ={3} \n y ={4}", x, a, b, c, y);
    Console.Read();
  }
 }
}
```

Результат роботи:



Приклад 1.2.3. Обчислити значення логічного виразу:

1. $(A \vee \neg A \wedge B) \vee C$
2. $\neg A \vee A \wedge (B \vee C)$
3. $(A \vee B \wedge \neg C) \wedge C)$

де $\neg$ – заперечення, $\wedge$ – кон'юнкція, $\vee$ – диз'юнкція.

При таких значеннях змінних: $A=1$, $B=0$, $C=0$.

Розвя'язок:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Variable_Declarations
{
  class Program
```

Національний університет
водного господарства
та природокористування

```
{
    static void Main()
    {
      bool A=true, B=false, C= false, D;
      D = (A | !A & B) | C;
      Console.WriteLine("D = ({0} | !{1} & {2})|{3}
                        ={4}", A, A, B, C, D);
      D = !A | A & (B | C);
      Console.WriteLine("D = !{0} | {1} & ( {2} | {3})
                        ={4}", A, A, B, C, D);
      D = (A | B & !C) & C;
      Console.WriteLine("D = ({0} | {1} & !{2}) & {3}
                       ={4}", A, B, C, C, D);
      Console.Read();
    }
  }
}
```

Результат роботи:

```
D = (True | !True & False)|False =True
D = !True | True & ( False | False) =False
D = (True | False & !False) & False =False
```

Приклад 1.2.4. Обчислити значення y, що задане формулою:

$$y = \begin{cases} 1, \text{ якщо } 2x^2 - 3 = 0, \\ 2x, \text{ якщо } 2x^2 - 3 > 0, \\ x, \text{ якщо } 2x^2 - 3 < 0. \end{cases}$$

Розв'язок:

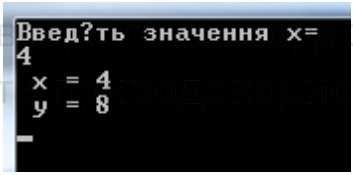
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Program1
{
  class Program
```

Національний університет
водного господарства
та природокористування

```
{
  static void Main()
  {
    double x, y;
    string buf;
    Console.WriteLine("Введіть значення x=");
    buf = Console.ReadLine();
    x = Convert.ToDouble(buf);
    if (2 * x * x - 3 == 0) y = 1;
    else
      if (2 * x * x - 3 > 0) y = 2 * x;
      else y = x;
    Console.WriteLine(" x = {0} \n y = {1}", x, y);
    Console.Read();
  }
}
}
```

Результат роботи:



Приклад 1.2.5. Використовуючи оператор вибору варіанту ***switch*** обчислити значення функції $y(x)$ для всіх варіантів вхідних параметрів. Значення x вводиться з клавіатури. Вивести значення функції для всіх варіантів.

Функція має вигляд:

$$y = \begin{cases} sin(e^{a+b}) + x^2, & якщо \quad a+b > x, \\ arctg(abc) + \sqrt[3]{x}, & якщо \quad a+b = x, \\ arcsin(cos^2(\sqrt{|x|})), & якщо \quad a+b < x. \end{cases}$$

Значення параметрів:

Національний університет
водного господарства
та природокористування

1. $a = 7.2;\ b = -1.3;\ c = 2.5$
2. $a = 1.47;\ b = 3{,}81\ c = 2.8$
3. $a = 4.8;\ b = 10{,}6;\ c = 2.7$

Розв'язок:

Для розв'язку задачі використатимемо оператор вибору варіанту ***switch***

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Program1
{
 class Program
 {
  static double x, y;
  static double a, b, c;
  static void Main()
  {
   string buf;
   Console.WriteLine("Введіть значення x=");
   buf = Console.ReadLine();
   x = Convert.ToDouble(buf);
   Console.WriteLine("Введіть номер варіанту
          обчислення (число від 1-3)");
   buf = Console.ReadLine();
   int i = Convert.ToInt32(buf);
   switch (i)
   {
    case 1: a = 7.2; b = -1.3; c = 2.5; break;
    case 2: a = 1.47; b = 3.81; c = 2.8; break;
    case 3: a = 4.8; b = 10.6; c = -2.7; break;
    default: Console.WriteLine("Такого варіанту
         немає"); break;
   }
   if (i==1|i==2|i==3)
   {
```

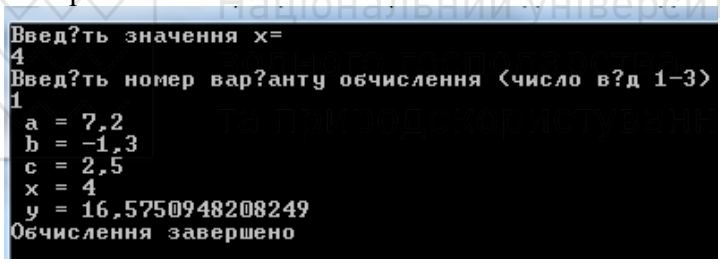
Національний університет
водного господарства
та природокористування

Національний університет
водного господарства
та природокористування

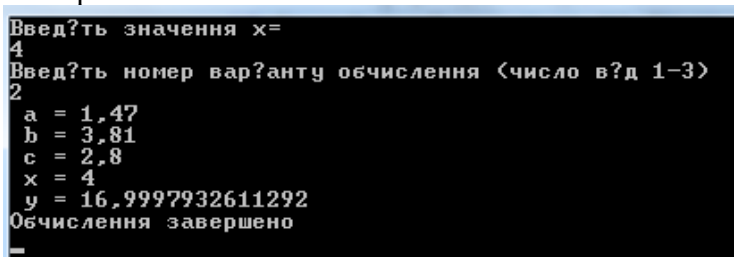
```
if (a + b > x) y = Math.Sin(Math.Exp(a + b)) + x * x;
    else if (a + b == x) y = Math.Atan(a * b * c) +
           Math.Pow(x, 1.0 / 3);
      else
           y = Math.Asin(Math.Pow(Math.Cos(Math.Sqrt
           (Math.Abs(x))), 2));
    Console.WriteLine(" a = {0} \n b = {1} \n c = {2}
              \n x = {3}\n y = {4}", a, b, c, x, y);
   }
   Console.WriteLine("Обчислення завершено");
   Console.Read();
  }
 }
}
```

Результат роботи:

Варіант 1



Варіант 2



Варіант 3

Національний університет
водного господарства
та природокористування

```
Введ?ть значення x=
4
Введ?ть номер вар?анту обчислення (число в?д 1-3)
3
 a = 4,8
 b = 10,6
 c = -2,7
 x = 4
 y = 15,7651683232153
Обчислення завершено
```

Варіант 4.

```
Введ?ть значення x=
4
Введ?ть номер вар?анту обчислення (число в?д 1-3)
4
Такого вар?анту немає
Обчислення завершено
```

1.3. Індивідуальні завдання

Варіант кожного завдання обирається за номером студента в журналі.

Завдання 1. Обчислити значення арифметичного виразу. Значення змінних вводиться з клавіатури.

№ вар.	Вираз
1.	$L = \dfrac{\sqrt{e^x - \cos^4(x^2 a^5)} + arctg^4(a - x^5)}{e\sqrt{\lvert a + xc^4 \rvert}}$
2.	$P = \dfrac{\sin^3 x + \ln(2y + 3x)}{t^e + \sqrt{x}}$
3.	$L = ctg^2 c + \dfrac{2x^2 + 5}{\sqrt{c + t}}$
4.	$T = \dfrac{\sqrt{x + b - a} + \ln y}{arctg(b + a)}$

Національний університет
водного господарства
та природокористування

5.	$A = \dfrac{tg(y^3 - h^4) + h^2}{\sin^3 h + y}$
6.	$s = \dfrac{4{,}351y^2 + 2r\ln r}{\sqrt{\cos 2y + 4{,}351}}$
7.	$F = \dfrac{\sqrt{(2+y)^2 + \sqrt[7]{\sin(y+5)}}}{\ln(x+1) - y^3}$
8.	$D = \dfrac{K^{-ax} - a\sqrt{6} - \cos(3ab)}{\sin^2(a - \arcsin x + \ln y)}$
9.	$G = \dfrac{tg(x^4 - 6) - \cos^3(z + xy)}{\cos^4 x^3 c^2}$
10.	$U = \dfrac{tg^3 y + \sin^5 x\sqrt{b-c}}{\sqrt{a-b+c}}$
11.	$K = \dfrac{\sqrt{x+b-a} + \ln y}{arctg(b+a)}$
12.	$N = \dfrac{\sqrt[5]{z + \sqrt{zx}}}{e^z + a^5 arctgx}$
13.	$D = \dfrac{\cos(x^3 + 6) - \sin(y - a)}{\ln x^4 - 2\sin^5 x}$
14.	$F = \cos(x^2 + 2) + \dfrac{3{,}5x^2 + 1}{\cos^2 y}$
15.	$P = \dfrac{a + \sin^4(y - c)}{\sin^3(x+y) + \lvert x - y\rvert}$
16.	$F = \dfrac{\sqrt{\lvert x\rvert + \cos^3 x + z^4}}{\ln x - \arcsin(bx - a)}$

Національний університет
водного господарства
та природокористування

17.	$R = \dfrac{\cos^3 y + 2^x d}{e^y + \ln(\sin^2 x + 7{,}4)}$
18.	$F = \dfrac{\cos^7 bx^5 - (\sin a^2 + \cos(x^3 + z^5 - a^2))}{\arcsin a^2 + \arccos(x^7 - a^2)}$
19.	$U = \dfrac{e^{x^3} + \cos^2(x - 4)}{arctgx + 5{,}2y}$
20.	$J = \dfrac{ctg^3 a^3 + arctg^2 a}{\sqrt{y^{tgx}}}$
21.	$I = \dfrac{2{,}33 \ln\sqrt{1 + \cos^2 y}}{e^y + \sin^2 x}$
22.	$U = \dfrac{\ln(x^3 + y) - y^4}{e^y + 5{,}4k^3}$
23.	$G = \dfrac{\cos^3\lvert y + x\rvert - (x + y)}{arctg^4(x + a)x^5}$
24.	$P = \dfrac{a^5 + \arccos(a + x^3) - \sin^4(y - c)}{\sin^3(x + y) + \lvert x - y\rvert}$
25.	$R = \dfrac{a}{x - a} + \dfrac{b^2 \cos^3 x}{lg^3 a + 4{,}5}$
26.	$G = \dfrac{tg(x^4 - 6) - \cos^{3z}(z + x^3 y)}{\cos^2 x^3 c^2}$
27.	$R = \dfrac{\sin(x^2 + 4)^3 + 4{,}3}{\sin^3 x^4}$
28.	$R = \dfrac{\cos^2 y + 2{,}4d}{e^y + \ln(\sin^2 x + 6)}$

Національний університет
водного господарства
та природокористування

29.	$N = \dfrac{m^2 + 2{,}8m + 0{,}355}{\cos 2y + 3{,}6}$
30.	$K = \dfrac{\sqrt{(3+x)^6 - \ln x}}{e^x + \arcsin 6x^2}$

Завдання 2. Обчислити значення функції. Значення x вводиться з клавіатури, значення іншого параметру задається як константа.

№ вар.	**Функція**	**Значення параметрів**
1.	$y = e^{2x} + 9{,}7^t$; $c = a^2 + \sqrt{bx}$; $a = lg\ x$	$x = 3{,}5$ $b = 0{,}4$
2.	$y = xa^3 + b^2$; $b = e^{2k} + ax$; $a = ln\lvert x\rvert$	$x = 1{,}3$ $k = 4$
3.	$y = a^3 / b^2$; $b = \sin p^2 + ax$; $a = e^{\sqrt{\lvert x\rvert}}$	$x = 2{,}1$ $p = 1$
4.	$y = tg^3 x^2 + kt$; $x = pt^2 + \sqrt{k}$; $k = \sqrt{pt}$	$t = 4{,}1$ $p = 3$
5.	$y = \sqrt[5]{\lvert a - bx\rvert}$; $a = lg\ x$; $b = \sqrt{x^2 + t^2}$	$x = 2{,}7$ $t = -6$
6.	$y = \sin^2(a + tg^3 b)$;	$x = 1{,}1$

Національний університет
водного господарства
та природокористування

	$a = \sqrt{\lvert x \rvert}$; $b = x^4 + m^2$	$m = 2$
7.	$y = lg^3 a + cos^5 x$; $a = c^4 + k^3$; $c = \sqrt{\lvert x \rvert}$	$x = 5$ $k = 8{,}2$
8.	$y = 7e^{\sqrt{\lvert x \rvert}} + cos^4 x$; $x = a + (b + c)^3$; $a = sinb$	$b = 2{,}2$ $c = 3{,}7$
9.	$y = cos^2 b + b\,cos^4 a^2$; $a = \sqrt[3]{\lvert b + c \rvert}$; $b = x + c^2$	$x = 8{,}52$ $c = 9$
10.	$y = cos^3(a + b^3)$; $a = tx + \lvert\sqrt{b}\rvert$; $b = lg^2 \lvert x \rvert$	$x = 0{,}9$ $t = 2$
11.	$y = a^3 / cos\,c$; $c = lg\,a + b^2$; $b = \sqrt{\lvert x \rvert} + e^{\sqrt{b}}$	$x = -8$ $b = 9{,}5$
12.	$y = xp^2 + t^5$; $p = x^2 - \sqrt{\lvert x \rvert}$; $t = x^2 + a^2$	$x = 4$ $a = 3{,}7$
13.	$y = sin^3 ax + \sqrt{b}\,cos\,x^2$; $a = ln\lvert x \rvert$; $b = x^4 + lg\,p^3$	$x = 1{,}4$ $p = 1{,}6$

Національний університет
водного господарства
та природокористування

Національний університет
водного господарства
та природокористування

14.	$y = lg^4\|x+t\|$; $x = p^2 + t$; $t = sinm^3$	$m = 5{,}7$ $p = 4$
15.	$y = \ln a / lg\, b^3$; $a = sin(x^2 + b^2)$; $x = n^b + b^2$	$n = 8{,}1$ $b = 1{,}6$
16.	$y = \ln x^{a+b} + \frac{a^2}{a+t}$; $a = lg\|tx + b^2\|$; $t = xb^2 + \sqrt{x}$	$x = 5{,}2$ $b = 0{,}3$

Національний університет
водного господарства
та природокористування

Завдання 3. . Обчислити значення логічного виразу, де ¬ – заперечення, ∧ – кон'юнкція, ∨ – диз'юнкція.

1. Обчислити значення логічних виразів при наступних значеннях логічних змінних A = 1, B = 0, C = 0

а) $(A \vee \neg A \wedge B) \vee X$ б) $\neg A \vee A \wedge (B \vee X)$ в) $(A \vee B \wedge \neg X) \wedge X$.

2. Обчислити значення логічних виразів при наступних значеннях логічних змінних X = 0, Y = 1, Z = 0

а) $X \wedge \neg(Z \vee Y) \vee \neg Z$ б) $\neg X \vee X \wedge (Y \vee Z)$

в) $(X \vee Y \wedge \neg Z) \wedge Z$

3. Обчислити значення логічного виразу

а) $(\neg(x \times y < 0)) \wedge (y > x)$ при x=2, y=1;

б) $(x \geq 2) \vee (y\ 2 \neq 5)$ при x=2, y=-2.

4. Записати умову, яка набуває значення «істина», якщо ціле число A кратно двом або трьом.

5. Записати умову, яка набуває значення «істина», якщо кожне з цілих чисел A і B непарне.

Національний університет
водного господарства
та природокористування

6. Обчислити значення логічних виразів при таких значеннях логічних змінних X = 0, Y = 0, Z = 1

а) $X \vee Y \wedge \neg Z$ б) $\neg X \wedge \neg Y$ в) $\neg(X \wedge Z) \vee Y$.

7. Записати логічне вираз, що описує область визначення функції

а) y=2tg x; б) y=3/(x-1).

8. Обчислити значення логічного виразу

а) $(x \geq 0) \wedge (y 2 \neq 4)$ при x = 1, y = 2;

б) $(x \times y \neq 0) \vee (y > x)$ при x = 2, y = 1.

9. Обчислити значення логічного виразу

а) $(\neg(x \times y < 0)) \vee ($
$y > x)$ при x=1, y=2;

б) $(x \times y \neq 0) \wedge (y < x)$ при x=2, y=1.

10. Обчислити значення логічних виразів при наступних значеннях логічних змінних A = 1, B = 0, C = 1

а) $A \vee \neg(A \wedge B) \vee C$; б) $\neg A \vee C \wedge \neg(B \vee C)$; в) $(A \vee B \wedge \neg C) \vee A$

11. Обчислити значення логічного виразу

а) $(\neg(x \times y < 1)) \wedge (y > x)$ при x=1, y=2;

б) $(x \times y \neq 0) \vee (y < x)$ при x=2, y=1.

12. Записати логічне вираз, який визначає, що число A є тризначним.

13. Записати умову, яка набуває значення «істина», якщо тільки одне з чисел A, B і C менше 45.

14. Записати умову, яка набуває значення «істина», якщо ціле число A не кратне трьом і закінчується нулем.

15. Записати логічне вираз, який визначає, чи належить число A інтервалу від -137 до -51 або інтервалу від 123 до 55.

16. Записати умову, яка набуває значення «істина», якщо тільки одне з чисел X, Y і Z кратне п'яти.

Національний університет
водного господарства
та природокористування

Завдання 4. Обчислити значення y, що задане формулою.

№ вар.	Завдання	№ вар.	Завдання
1	$y=\begin{cases}\cos(A+C+N), \text{якщо } A=C=N,\\ \cos(A\cdot C\cdot N), \text{якщо } A<C=N,\\ \cos((A+C)\cdot N), \text{якщо } A<C<N,\\ 0, \quad \text{в інших випадках.}\end{cases}$	2	$y=\begin{cases}ln\lvert x\rvert - n, \text{якщо } x<n,\\ ln\lvert x\rvert - n, \text{якщо } x=n,\\ cos(nx), \text{якщо } x>n.\end{cases}$
3	$y=\begin{cases}cos(x^2+ln(x)), \text{якщо } x^2+ln(x)>0\\ \dfrac{1}{(x^2+ln(x))}, \text{якщо } x^2+ln(x)<0,\\ cos(x), \quad \text{якщо } x^2+ln(x)=0.\end{cases}$	4	$y=\begin{cases}1, \text{якщо } a<b<c,\\ 2, \text{якщо } a=b=c,\\ 3, \text{якщо } b<a<c,\\ 4, \text{якщо } c<a<b,\\ 0, \text{в інших випадках.}\end{cases}$
5	$z=\begin{cases}ye^x, \text{якщо } y<x,\\ y\cdot x, \text{якщо } y=x,\\ x\cdot e^y, \text{якщо } y>x.\end{cases}$	6	$y=\begin{cases}1, \text{якщо } 0\le x<5,\\ 2, \text{якщо } 5\le x<8,\\ 3, \quad \text{якщо } x<0,\\ 4, \quad \text{якщо } x\ge 8.\end{cases}$
7	$y=\begin{cases}1, \text{якщо } 2x^2-x-3=0,\\ 2, \text{якщо } 2x^2-x-3>0,\\ 0, \text{якщо } 2x^2-x-3<0.\end{cases}$	8	$y=\begin{cases}0, \text{якщо } 0\le n<5,\\ 1, \text{якщо } 5\le n<10,\\ 2, \text{якщо } 10\le n<15,\\ 3, \text{в інших випадках.}\end{cases}$
9	$y=\begin{cases}e^t\cdot sin(t), \text{якщо } t<0\\ e^t\cdot cos(t), \text{якщо } t\ge 0\end{cases}$ $t=\begin{cases}cos(x), \text{якщо } x<2,\\ ln(x), \text{якщо } x\ge 2.\end{cases}$	10	$z=\begin{cases}ln(\quad)-tg(\quad), \text{якщо } 1\le x\le 3,\\ tg(\quad), \text{якщо } 3<x\le 4,\\ 0, \text{ в ыших випадках}\end{cases}$
11	$y=\begin{cases}cos(\quad), \text{якщо } n=1,\\ x^n, \quad \text{якщо } n=2,\\ n^x, \quad \text{якщо } n=3,\\ 0, \text{в інших випадках.}\end{cases}$	12	$y=\begin{cases}0, \text{якщо } n=1,2,3,4,\\ 1, \text{якщо } n=10,11,12,\\ 2, \text{якщо } n=15,19,\\ 3, \text{в інших випадках.}\end{cases}$

Національний університет
водного господарства
та природокористування

13	$y = \begin{cases} 1, \text{ якщо } 2x^2 - 3 = 0, \\ 2x, \text{ якщо } 2x^2 - 3 > 0, \\ x, \text{ якщо } 2x^2 - 3 < 0. \end{cases}$	14	$z = \begin{cases} \sin(n), \text{ якщо } n = 10, \\ x^n, \quad \text{якщо } n = 22, \\ n^x, \quad \text{якщо } n = 3, \\ 1, \text{в інших випадках}. \end{cases}$

Завдання 5. Використовуючи оператор ***switch*** розв'язати задачу згідно варіанту:

№ вар.	Задача	Параметри
1.	Визначити, чи лежить точка А в області, обмеженею параболою $y = 2 - x^2$ і віссю абсцис. Відповідь вивести у вигляді повідомлення.	1. *x=3.5; y=7.2* 2. *x=-0.5; y=1.2* 3. *x=0.72; y=-3.12*
2.	Дано три числа. Вивести на екран їх середнє значення.	1. *a=3; b=3.5; c=-2.1* 2. *a=2,1; b=-6.55; c=0.1* 3. *a=-9; b=-3.7; c=-0.1*
3.	Дано координати точки x і y і радіус кола R. Центром кола є початок координат. Визначити, чи належить дана точка колу.	1. *x=2; y=1.5* 2. *x=-0.5; y=2.3* 3. *x=1.5; y=-0.8*
4.	Дано значення трьох сторін трикутника *a, b* і *c*. Визначити, чи є він прямокутним.	1. *a=3; b=3.5; c=-2.1* 2. *a=21; b=-6.55; c=0.1* 3. *a=-9; b=-3.7; c=-0.1*
5.	Ввести з клавіатури номер місяця. Визначити сезон в залежності від номера місяця і вивести повідомлення (весна (3,4,5), літо (6,7,8), осінь (9,10,11) зима (12,1,2)).	1. *4* 2. *6* 3. *11* 4. *2*
6.	Дано три числа *A, B, C*. Збільшити числа в два рази, якщо $A + B + C > 0$, в іншому випадку замінити на нулі.	1. *A=-3; B=3.5; C=0.1* 2. *A=58; B=27; C=-87* 3. *A=-8; B=-35;*

Національний університет
водного господарства
та природокористування

		C=42
7.	Дано координати точок *A (x0, y0)* і *B (x1, y1)*. Визначити, яка з точок *A* чи *B* найменш віддалена від початку координат *(O (0,0))*.	1. $x_0=2;\ y_0=2;\ x_1=-4;\ y_1=0;$ 2. $x_0=8;\ y_0=9;\ x_1=12;\ y_1=1;$ 3. $x_0=-3.5;\ y_0=0.9;\ x_1=2;\ y_1=3;$
8.	Дано значення трьох сторін трикутника *a, b* і *c* і визначити, чи є він рівнобедреним. Відповідь вивести у вигляді повідомлення.	1. *a=3; b=3.5; c=1,1* 2. *a=3; b=6.55; c=6.55* 3. *a=0.9; b=0.9; c=0.9*
9.	Дано вартість хвилини розмови у трьох операторів мобільного зв'язку, а також процентна знижка на дзвінки у вихідні дні. Ввести з клавіатури кількість часу розмов в будні і у вихідні дні, визначити вартість розмов у різних операторів.	1. *50 коп, 10%* 2. *45 коп, 8%* 3. *55 коп, 12%*
10.	Поштовий автомат пропонує вітальні листівки на три теми в трьох варіантах. Ввести з клавіатури номер теми (від 1 до 3) і номер варіанта (a, b, c), видати потрібну листівку (повідомлення, наприклад: «Новорічні, варіант 3"). Передбачити обробку неправильного номера або варіанту.	1. *новорічні.* 2. *з днем нарождения.* 3. *З 8 березня.*
11.	Дано два числа *a* і *b*. Розробити і описати алгоритм, в результаті якого числа змінюють свій знак на протилежний, якщо вони мають різний знак і будуть замінені на нулі, якщо це не так.	1. *a=-3; b=3.5;* 2. *a=5.8; b=2.7;* 3. *a=-8; b=-3.5;*
12.	Дано три числа. Вивести найменше із них.	1. *a=45, в=13, c=-23* 2. *a=-31, в=65, c=12* 3. *a=52,в=-1, c=-33*
13.	Дано три числа *x, y, z*. Вивести на екран числа по модулю більші середнього арифметичного цих чисел.	1. *x=3.2; y=-7; z=0.5* 2. *x=2.3; y=3; z=2.5* 3. *x=23; y=-34; z=89.5*
14.	Дано координати точки B (x, y). Визначити, чи лежить дана точка на	1. *x=2.2; y=-6* 2. *x=0.3; y=3*

Національний університет
водного господарства
та природокористування

	кривій $f(x)=\begin{cases}\sin^2 x^3, & якщо \ \ |x|>1,\\ \sqrt{6\arcsin x^2+4{,}5x^6+4x^2+2}, & якщо \ \ |x|\le 1.\end{cases}$ Похибка складає $eps=10^{-3}$, тобто $\lvert f(x)-y\rvert \le eps$.	3. *x=-0.8; y=-4*
15.	Дано координати точки *A (x, y)*. Визначити, чи лежить дана точка в четвертій чверті.	1. *x=-2.2; y=-6* 2. *x=5.3; y=3* 3. *x=-0.8; y=-4*
16.	Дано значення трьох сторін трикутника *a, b*, і *c*. Найменша зі сторін трикутника є стороною квадрата. Визначити, площа якої з фігур більша.	*1. a=3; b=5; c=4* *2. a=3; b=8; c=11* *3. a=9; b=9; c=9*

1.4. Зміст звіту:

1. Прізвище та ім'я студента.
2. Номер і назва лабораторної роботи.
3. Мета роботи.
4. Номер індивідуального завдання. Текст завдання (постановка завдання).
5. Лістинг програми.
6. Скриншот робочої програми.
7. Висновок про засвоєнні знання та вміння.

1.5. Питання для самоперевірки

1. Назвіть базові поняття платформи .Net.
2. Що таке «збірка»?
3. Яку дію виконує JIT компілятор?
4. Що таке «керований код»?
5. Опишіть систему типів платформи .Net.
6. Опишіть процес створення консольного додатку в середовищі Visual Studio.NET
7. З чого складається алфавіт мови C#?

Національний університет
водного господарства
та природокористування

8. Наведіть приклад лексем у мові C#.
9. Для чого призначені директиви препроцесора?
10. З чого складаються арифметичні вирази?
11. З чого складаються логічні вирази?
12. Назвіть основні математичні функції, які доступні у мові C#.
13. Який пріоритет операцій у мові C#.
14. Що таке буквальні літерали?
15. На які категорії поділяються типи даних в мові C#?
16. Які типи відносяться до типів-значень (value types)?
17. Які типи відносяться до типів-посилань (reference types)?
18. Що таке приведення типів? Як воно здійснюється в мові C#?
19. У яких випадках використовується неявне перетворення типів?
20. Який загальний синтаксис явного перетворення типів?
21. Що таке унарна операція? Назвіть унарні операції в мові C#.
22. Що таке бінарна операція? Наведіть приклади бінарних операції в мові C#.
23. Як можна здійснити розгалуження в мові C#?
24. Як працює умовний оператор?
25. Яка різниця між повним на неповним умовними операторами?
26. Яка конструкція використовується для більш зручної та компактної реалізації багатоальтернативних галужень в мові *C#*?

Національний університет
водного господарства
та природокористування

Лабораторна робота № 2. Організація циклів в програмах мовою C#.

Мета роботи: набути навички програмування консольних додатків з використанням операторів циклу.

Послідовність виконання роботи:

1. Ознайомитись із теоретичними відомостями. (*Актуалізація опорних знань*).
2. Виконати програмування програм за поданими прикладами. Результат представити викладачу (Застосування набутих *знань*).
3. Виконати варіант самостійної роботи. (*Закріплення набутих знань*).
4. Оформити звіт на виконану роботу. (*Узагальнення та систематизація набутих знань*).
5. Захист звітів, відповіді на запитання.

2.1. Опорні теоретичні відомості

Для реалізації повторень деякої команди або групи команд використовують оператори циклів. Оператор циклу складається з заголовку, в якому вказується умова завершення циклу, та тіла циклу. Тіло циклу – це оператор (складений оператор), який потрібно виконати декілька разів. Кожне виконання оператора тіла циклу називається його *ітерацією*. В мові програмування C# передбачено 4 види операторів циклів:

1. цикл for;
2. цикл while;
3. цикл do-while;
4. цикл foreach.

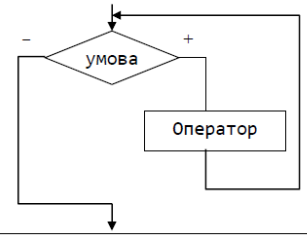
Розглянемо перші три види операторів циклів.

Оператор циклу while. Оператор *while* циклічно

Національний університет
водного господарства
та природокористування

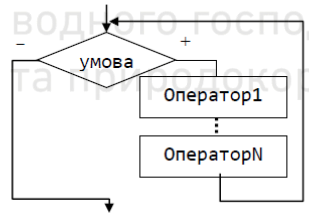
виконує своє тіло до тих пір, поки умова виконується (логічний вираз приймає значення true):

Програмна структура	Аналог на мові блок-схем	**Приклад.** Знайти суму перших n натуральних чисел.

`while (<умова>)`
`  <оператор>;`
		`int sum=0;`
		`int i=1;`
		`while(i<=n)`
		`  sum=sum+i++;`

Якщо тіло циклу складається з більше ніж одного оператора, то необхідно використати складений оператор:

Програмна структура	Аналог на мові блок-схем	**Приклад.** Знайти суму і добуток перших n натуральних чисел.

`while (<умова>)`
`{`
`  <оператор1>;`
`  . . . . . .`
`  <операторN>;`
`}`
		`int sum=0;`
		`int mult=1;`
		`int i=1;`
		`while(i<=n)`
		`{`
		`  sum=sum+(i++);`
		`  mult=mult*(i++);`
		`}`

Порядок виконання циклу наступний:

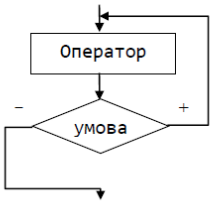
1. Обчислюється <умова>, значення якої має бути типу bool. Якщо значенням є false, то дія циклу закінчується. Інакше виконується <оператор циклу>.
2. Після цього повторюється перший крок.

Зрозуміло, що в операторі циклу (або в блоці операторів) слід передбачити якийсь вплив на умову циклу, інакше він, розпочавшись, не зможе завершитись.

Оператор циклу do-while. Цей цикл на відміну від попереднього називають циклом з післяумовою, оскільки умова виходу з циклу аналізується вже після його

Національний університет
водного господарства
та природокористування
виконання.

Програмна структура	Аналог на мові блок-схем	Приклад. Знайти суму перших n натуральних чисел.

do
{
<оператор>;
}
while (<умова>)
		int sum=0;
		int i=1;
		do
		{
		sum=sum+i++;
		}
		while(i<=n)

Порядок виконання циклу наступний:

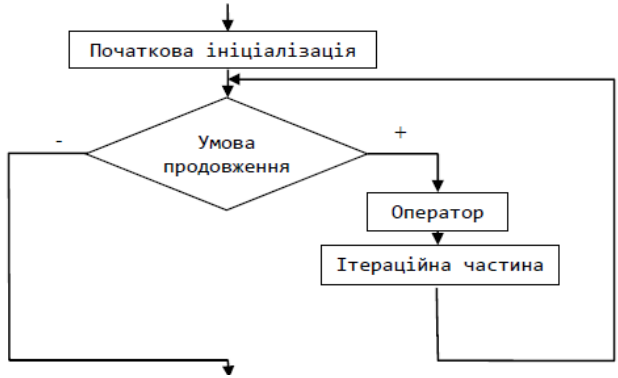
1. Виконуються всі інструкції в тілі циклу.
2. Обчислюється <умова>, значення якої має бути типу bool. Якщо його значенням є false, то дія циклу закінчується. Інакше повторюється перший крок.

Цикл *do-while* відрізняється від циклів *while* та *for* тим, що його інструкції виконуються ***завжди*** при наймні один раз.

Оператор циклу for. Загальний вигляд оператора такий:

`for` (<вираз ініціалізації>`;`<умова продовження>`;`<ітераційна частина>) <оператор>`;`

Аналог мовою блок-схем:



Оператор for працює у відповідності до такого алгоритму:

Національний університет
водного господарства
та природокористування

1. Обчислюється вираз ініціалізації. У цій частині допустима ініціалізація декількох лічильників циклу.
2. Перевіряється умова продовження. Якщо умова невірна то робота циклу завершується і передається управління наступному оператору.
3. Якщо умова істинна, виконується тіло даного оператора.
4. Виконується приріст одного або декількох лічильників циклу (або виконується довільна інша операція).
5. Здійснюється перехід до кроку 2.

Керування виходом із циклів C#. Для дострокового виходу з циклу використовуються оператори break та continue.

Оператор continue може бути використаний у будь-якому із циклів у випадку, коли немає потреби виконувати до кінця усі оператори тіла циклу поточної ітерації, а необхідно одразу перейти до наступної ітерації. Наприклад, для обчислення суми парних натуральних чисел, що менше за N, можна описати такий цикл:

```
for (int i = 1;i<=N; i++)
{
if ((i % 2) != 0)
continue;
suma += i;
}
```

Оператор break у циклах використовують для негайного виходу з того циклу, в тілі якого він знаходиться.

2.2. Приклади виконання завдань

Приклад 2.2.1. Використовуючи оператор циклу з передумовою (цикл *while*) та оператор циклу з

Національний університет
водного господарства
та природокористування

постумовою (цикл do-while) скласти програму для обчислення добутку квадратів непарних чисел в інтервалі, який задано значеннями змінних *m* і *n*.

Опишемо послідовність дій, які потрібно здійснити для обчислення, використовуючи цикл з передумовою .

1. Ввести значення змінних *m* і *n*.
2. Надати добутку *P* значення 1.
3. Перевірити чи $m \le n$. Якщо так, то перейти до пункту 4, інакше перейти до пункту 7.
4. Якщо *m* – непарне, то домножити *P* на m^2.
5. Збільшити *m* на 1.
6. Перейти до пункту 3.
7. Вивести результат обчислення.

Розв'язок:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Program1
{
 class Program
 {
  static double x, y;
  static double a, b, c;
  static void Main()
  {
   string buf;
   Console.WriteLine("Введіть значення m=");
   buf = Console.ReadLine();
   int m = Convert.ToInt32(buf);
   int m1 = m; // запам'ятовуємо початкове значення m
   Console.WriteLine("Введіть значення n=");
   buf = Console.ReadLine();
   int n = Convert.ToInt32(buf);
   int P = 1;
   while (m<n)
```

Національний університет
водного господарства
та природокористування

```
    {
     if (m % 2 != 0) P *= m * m;
     m++;
    }
    Console.WriteLine("m= {0} n={1} P={2}", m1, n, P);
    Console.Read();
   }
  }
}
```

Розв'яжемо цю ж задачу з використанням циклу з постумовою (цикл *do-while*).

Опишемо послідовність дій, які потрібно здійснити для обчислення:

1. Ввести значення змінних m і n.
2. Надати добутку P значення 1.
3. Якщо m – непарне, то домножити P на m^2.
4. Збільшити m на 1.
5. Перевірити чи $m \le n$. Якщо так, то перейти до пункту 3, інакше перейти до пункту 6.
6. Вивести результат обчислення.

Розв'язок:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Program1
{
 class Program
 {
  static double x, y;
  static double a, b, c;
  static void Main()
  {
   string buf;
```

Національний університет
водного господарства
та природокористування

```
    Console.WriteLine("Введіть значення m=");
    buf = Console.ReadLine();
    int m = Convert.ToInt32(buf);
    int m1 = m; // запам'ятовуємо початкове значення m
    Console.WriteLine("Введіть значення n=");
    buf = Console.ReadLine();
    int n = Convert.ToInt32(buf);
    int P = 1;
    do
    {
     if (m % 2 != 0) P *= m * m;
     m++;
    }
    while (m < n);
    Console.WriteLine("m= {0} n={1} P={2}", m1, n, P);
    Console.Read();
   }
  }
}
```

Національний університет
водного господарства
та природокористування

Як бачимо, результати обчислення співпадають.

Зауваження. В наведених прикладах не розглядалися таки випадки:

1. Неправильне введення інтервалу, тобто *m* більше *n;*
2. Наявність в інтервалі нульового значення. Наприклад інтервал [-5; 4] містить нуль.

Приклад 2.2.2. Дано дійсне число A і натуральні числа від 1 до N>0. Знайти різницю квадратів A і натуральних чисел N.

Розв’язок:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Program2
```

Національний університет
водного господарства
та природокористування

```
{
 class Program
 {

  static void Main()
  {
   string buf;
   Console.WriteLine("Введіть значення A=");
   buf = Console.ReadLine();
   double A = Convert.ToDouble(buf);

   Console.WriteLine("Введіть значення N=");
   buf = Console.ReadLine();
   int N = Convert.ToInt32(buf);
   for (int i = 1; i <= N; i++)
   {
    Console.WriteLine("N={0} {1}", i, A * A - i * i);
   }
   Console.Read();
  }
 }
}
```

Національний університет
водного господарства
та природокористування

Приклад 2.2.3. Обчислити суму перших N членів: $sin(x)+sin(3x^3)+sin(5x^5)+...$

Розв'язок:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Program2
{
 class Program
 {

  static void Main()
  {
   string buf;
   Console.WriteLine("Введіть значення N=");
```

```
    buf = Console.ReadLine();
    int N = Convert.ToInt32(buf);
    Console.WriteLine("Введіть значення x=");
    buf = Console.ReadLine();
    double x = Convert.ToDouble(buf);
    double x1 = x*x;
    double S = 0;
    for (int i = 1; i <= N; i++)
    {
     S += Math.Sin((2 * i - 1) * x);
     x *= x1;

    }
    Console.WriteLine("N={0} {1}", N, S);
    Console.Read();
  }
 }
}
```

Зауваження. Розглянемо як змінюється аргумент: перший множник – це непарне число, починаючи з 1. Другий доданок кожного разу множиться на x^2. Тому перший доданок задаємо як *(2 * i – 1)*, де i – це номер ітерації, а для того, щоб не підносити кожного разу значення x до непарного степеня, множимо це значення на *x1 = x*x*. (Користуємося загальновідомою формулою: $x^a \cdot x^b = x^{a+b}$).

Приклад 2.2.4. Обчислити суму $sin(x)+sin(3x^3)+sin(5x^5)+...$ з похибкою $eps=10^{-3}$.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Program2
{
 class Program
 {
```

Національний університет
водного господарства
та природокористування

```
  static void Main()
  {
   string buf;

   Console.WriteLine("Введіть значення x=");
   buf = Console.ReadLine();
   double x = Convert.ToDouble(buf);
   double x1 = x * x;
   Console.WriteLine("Введіть значення eps=");
   buf = Console.ReadLine();
   double eps = Convert.ToDouble(buf);
   double S = 0;
   int i=1;
   double Curr; // поточний елемент суми

   do
   {
    Curr = Math.Sin((2 * i - 1) * x);
    S += Curr;    //додавання в суму
    i++;
    x *= x1;

   } while (Math.Abs(Curr)>eps); //перевірка похибки
   Console.WriteLine("i={0} S={1:f7}", i, S);
   Console.Read();
  }
 }
}
```

Національний університет
водного господарства
та природокористування

2.3. Індивідуальні завдання

Варіант кожного завдання обирається за номером студента в журналі.

Завдання 1. Скласти програму для очислення з використанням операторів циклу з перед- та післяумовою.

1. Піднести число A в цілу степінь N.

Національний університет
водного господарства
та природокористування

2. Обчислити факторіал заданого цілого числа. Факторіал числа N обчислюється за такою формулою: $n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots \cdot n$.
3. Обчислити суму S квадратів чисел від 1 до N.
4. Обчислити суму S квадратів парних і кубів непарних чисел від 1 до N.
5. Знайти всі числа некратні п'яти і кратні 3 з інтервалу *[m,n].*
6. Знайти всі числа кратні п'яти для чисел від 1 до N.
7. Чи є задане натуральне число ступенем двійки?
8. Знайти суму квадратів чисел від m до n.
9. Знайти суму квадратів непарних чисел в інтервалі, заданому значеннями змінних m і n.
10. Знайти добуток цілих непарних чисел, кратних 7 від -80 до 80.
11. Знайти суму цілих додатних чисел, кратних 9 від -10 до 10.
12. Знайти 100 перших простих чисел (просте число це натуральне число, яке має рівно два різних натуральних дільники: одиницю і самого себе).
13. Знайти добуток квадратів парних чисел в інтервалі, заданому значеннями змінних m і n.
14. Знайти суму цілих чисел, кратних 5 і від 20 до 20.
15. Знайти суму цілих додатних чисел, кратних 4 і менших 100.
16. Напишіть програму виведення всіх непарних чисел від 100 до 200 включно.

Завдання 2. Скласти програму для очислення з використанням оператору циклу *for.*

1. Дано два цілих числа *A* і *B (A*≺*B).* Вивести на екран всі цілі числа, розташовані між цими числами (включаючи самі ці числа), в порядку їх зростання, а також обчислити кількість цих чисел.

Національний університет
водного господарства
та природокористування

2. Дано два цілих числа *A* і *B (A<B).* Вивести на екран всі цілі числа, розташовані між даними числами (не включаючи самі ці числа), в порядку їх спадання, а також обчислити кількість цих чисел.
3. Дано дійсне число *A* і ціле число *n>* 0. Знайти *A* в степені *N*: (числа *A* перемножуються *n* раз).
4. Дано дійсне число *a* і ціле число *n>0.* Знайти всі цілі ступені числа *A* від *1* до *n.*
5. Дано дійсне число *a* і ціле число *n>0.* Вивести значення суми $1 + a + a^2 + a^3 + \ldots + a^n$.
6. Дано дійсне число *a* і ціле число *n>0*. Знайти
   6.1.1. $1 - a + a^2 - a^3 + \ldots + (-1)^n a^n$.
7. Дано дійсне число *a>1*. Вивести найменше із цілих чисел *n*, для яких сума $1 + 1/2 + \cdots + 1/n$ буде більше *A*, і саму цю суму.
8. Дано ціле число *n>0*. Знайти добуток *1·2 ·…· n*. Щоб уникнути цілочисельного переповнення, обчислювати цей добуток за допомогою дійсної змінної і виводити його як дійсне число.
9. Дано ціле число *n>0*.Знайти добуток *2·1/(2)· 1/(3)·…·1/(n)*.
10. Дано дійсне число *x* і ціле число *n>0*. Знайти значення
    $1 + x + x^2 / 2 + \ldots + x^n/n$.
11. Дано дійсне число x і ціле число n>0.Знайти значення
    $x - x^3/3 + x^5/5 - \ldots + (-1)^{n-1} x^{2n-1}/(2n - 1)$.
12. Дано дійсне число *x* і ціле число *n>0*. Знайти значення
    $1 - x^2/2 + x^4/4 - + (-1)^{n+1} x^{2n}/(2n)$.
13. Дано дійсне число x ($|x| < 1$) і ціле число n (> 0).
14. Знайти значення
    $x - x^2/2 + x^3/3 - \ldots + (-1)^{n-1} x^n/n$.
    Отримане число є наближених значенням функції *ln* в точці $1 + x$.
15. Вивести квадрати чисел від 11 до 99.
16. Для заданого *n* в одному циклі обчислити *n*! и 2^n.

Національний університет
водного господарства
та природокористування

17. Задано натуральні числа від 10 до *n*. Вивести непарні кратні трьом числа.

Завдання 3.

Обчислити суму ряду:

а) перших *n* членів ряду;

б) з похибкою *eps.*

№ вар.	Завдання
1.	$1+\frac{x}{1!}+\frac{x^2}{2!}+\ldots$
2.	$\cos(x)+\frac{\cos(2x)}{2}+\frac{\cos(3x)}{3}+\ldots$
3.	$x-\frac{x^3}{3!}+\frac{x^5}{5!}-\ldots$
4.	$\sin(x)-\frac{\sin(2x)}{2}+\frac{\sin(3x)}{3}-\ldots$
5.	$\cos(x)+\frac{\cos(3x)}{9}+\frac{\cos(5x)}{25}+\ldots$
6.	$1+\frac{\cos\left(\frac{\pi}{4}\right)}{1!}x+\frac{\cos\left(2\frac{\pi}{4}\right)}{2!}x^2+\ldots$
7.	$x-\frac{x^2}{2!}+\frac{x^4}{4!}-\ldots$
8.	$x\sin\left(\frac{\pi}{4}\right)+x^2\sin 2\left(\frac{\pi}{4}\right)+x^3\sin 3\left(\frac{\pi}{4}\right)+\ldots$

Національний університет
водного господарства
та природокористування

9.	$x+\frac{x^5}{5!}+\frac{x^9}{9!}+\frac{x^{13}}{13!}+...$
10.	$1+\frac{\cos(x)}{1!}+\frac{\cos(2x)}{2!}+...$
11.	$1+\frac{3x^2}{1!}+\frac{5x^4}{2!}+\frac{7x^2}{3!}...$
12.	$\frac{x\cos\left(\frac{\pi}{3}\right)}{1}+\frac{x^2\cos\left(2\frac{\pi}{3}\right)}{2}+\frac{x^3\cos\left(3\frac{\pi}{3}\right)}{3}+...$
13.	$\frac{x-1}{x+1}+\frac{1}{3}\left(\frac{x-1}{x+1}\right)^2+\frac{1}{5}\left(\frac{x-1}{x+1}\right)^5+...$
14.	$-\cos(x)+\frac{\cos(2x)}{4}-\frac{\cos(3x)}{9}+...$
15.	$\frac{x^3}{3}-\frac{x^5}{15}+\frac{x^7}{35}-\frac{x^9}{63}+...$
16.	$\sin(x)+\frac{\sin(3x)}{3}+\frac{\sin(5x)}{5}+...$

2.4. Зміст звіту:

1. Прізвище та ім’я студента.
2. Номер і назва лабораторної роботи.
3. Мета роботи.
4. Номер індивідуального завдання. Текст завдання (постановка завдання).
5. Блок-схема програми.
6. Лістинг програми.
7. Скриншот робочої програми.
8. Висновок про засвоєнні знання та вміння.

Національний університет
водного господарства
та природокористування

2.5. Питання для самоперевірки

1. Який алгоритм називають циклічним?
2. Який блок позначає в блок-схемі команду перевірки умови? Наведіть приклади його використання.
3. Які види циклів ви знаєте? Зобразіть їх графічно.
4. У яких випадках застосовують оператори циклу?
5. Що таке тіло циклу?
6. Що називають ітерацією циклу?
7. Який загальний вигляд оператора циклу while?
8. Який загальний вигляд оператора циклу do-while?
9. Які відмінності у роботі операторів циклу while та do-while?
10. Який загальний вигляд оператора for?
11. За яким алгоритмом виконується оператор for?
12. Яке призначення оператора break?
13. Яке призначення оператора continue?

Національний університет
водного господарства
та природокористування

Література

1. Сборник задач по программированию / за ред. Глазунова Л. В. Одесса : ОНАС им. А. С. Попова, 2011. 212 с.
2. Соловей Л. В., Мирошниченко Н. Н., Пономарёв Н. Г. Программирование на языке C#: учеб. пособие. Х. : НТУ «ХПИ», 2016. 356 с.
3. Брила А. Ю., Антосяк П. П., Глебена М. І., Чупов С. В., Семйон І. В. Основи програмування у C#. Методичні вказівки до лабораторних робіт для студентів І-го курсу математичного факультету спеціальності "Прикладна математика". Ужгород, 2014. 60 с.
4. Парфьонов Ю. Е., Федорченко В. М., Лосєв М. Ю., Щербаков О. В. Об’єктно-орієнтоване програмування : конспект лекцій для студентів напряму підготовки

Національний університет
водного господарства
та природокористування
"Комп'ютерні науки" всіх форм навчання. Харків : Вид. ХНЕУ, 2010. 312 с.
5. Настенко Д. В., Нестерко А. Б. Об'єктно-орієнтоване програмування. Частина 1. Основи об'єктно-орієнтованого програмування на мові C# : навчальний посібник. К.: НТУУ «КПІ», 2016. 76 с.
6. Конспект лекцій, завдання та методичні вказівки до виконання лабораторних робіт з дисципліни "Програмування" розташовані на електронному ресурсі https://sites.google.com/site/emonitorlab/.
7. 04-04-203 Ольшанський, П. В. (2017) Методичні вказівки для виконання лабораторних та самостійних робіт з дисципліни "Об'єктно-орієнтоване програмування" студентами напряму підготовки 6.050102 "Комп'ютерна інженерія". Частина І. [Методичне забезпечення] URL: http://ep3.nuwm.edu.ua/5988/.
8. 04-04-201 Ольшанський П. В. (2017) Методичні вказівки для виконання лабораторних робіт з дисципліни "Програмування під платформу .NET" студентами напряму підготовки 6.050102 "Комп'ютерна інженерія". Частина ІІ [Методичне забезпечення] URL: http://ep3.nuwm.edu.ua/6059/1/04-04-201.pdf
9. 04-04-200 Ольшанський П. В. (2017) Методичні вказівки для виконання лабораторних робіт з дисципліни "Програмування під платформу .NET" студентами напряму підготовки 6.050102 "Комп'ютерна інженерія". Частина І. [Методичне забезпечення] URL: http://ep3.nuwm.edu.ua/5986/1/04-04-200.pdf
10. 04-03-154 Сафоник, А. П. (2017) Методичні вказівки для виконання лабораторних та самостійних робіт з дисципліни "Інформатика та комп'ютерна техніка"

Національний університет
водного господарства
та природокористування

студентами спеціальності 151 "Автоматизація і комп’ютерно-інтегровані технології". [Методичне забезпечення] URL: http://ep3.nuwm.edu.ua/6081/1/04-03-153.pdf.

Національний університет
водного господарства
та природокористування