

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ ЛАБОРАТОРНІ РОБОТИ ЧАСТИНА 1

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за спеціальністю 141 «Електроенергетика, електротехніка та електромеханіка»*

Київ
КПІ ім. Ігоря Сікорського
2020

Обчислювальна техніка та програмування: Лабораторні роботи (Частина 1) [Електронний ресурс]: навч. посіб. для студ. спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / КПІ ім. Ігоря Сікорського; уклад.: А.Б.Нестерко, Д.В.Настенко, Г.О.Труніна – Електронні текстові данні (1 файл: 1,514 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2020. – 83с.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 3 від 05.11.2020р.)
за поданням Вченої ради Факультету електроенерготехніки та автоматики
(протокол № 2 від 28.09.2020 р.)*

Електронне мережне навчальне видання

ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ ЛАБОРАТОРНІ РОБОТИ ЧАСТИНА 1

Укладачі: *Нестерко Артем Борисович, канд. техн. наук, ст.викл.,
Настенко Дмитро Васильович, ст.викл.,
Труніна Ганна Олексіївна, канд. техн. наук, ст.викл.*

Відповідальний редактор *Яндульський О.С., д.т.н., проф.*

Рецензент: *Кацадзе Т.Л., к.т.н., доц.
КПІ ім.Ігоря Сікорського, ФЕА, кафедра електричних мереж*

Посібник містить матеріали 12-ти занять, які включають в себе теоретичні відомості та приклади розв'язання різних задач за допомогою елементів алгоритмічної мови С#: операторів розгалуження, циклів, масивів і т.д. В наведених матеріалах міститься роз'яснення принципів роботи з базовими типами даних, програмування арифметичних виразів та роботи з текстовими рядками. Для кожного заняття приведено індивідуальні завдання для студентів, виконання яких дозволить закріпити знання основних принципів роботи з використанням сучасної алгоритмічної мови програмування С#.

© КПІ ім. Ігоря Сікорського, 2020

ЗМІСТ

Вступ.....	4
Вимоги безпеки під час роботи в лабораторії обчислювальної техніки	5
Загальні вимоги до виконання лабораторних робіт	8
Лабораторна робота №1. Знайомство з середовищем Visual Studio. Розробка елементарної програми мовою програмування С#.....	9
Лабораторна робота №2. Базові типи даних. Форматування консольного виведення	18
Лабораторна робота №3. Програмування арифметичних виразів	30
Лабораторна робота №4. Робота з текстовими рядками. Частина 1. Базові операції з рядками	38
Лабораторна робота №5. Робота з текстовими рядками. Частина 2. Розширені можливості роботи з рядками	43
Лабораторна робота №6. Оператор розгалуження if/else	50
Лабораторна робота №7. Оператор множинного вибору switch/case.....	54
Лабораторна робота №8. Оператори циклу. Частина 1. Цикл for.....	59
Лабораторна робота №9. Оператори циклу. Частина 2. Цикли while і do/while	63
Лабораторна робота №10. Одновимірні масиви. Частина 1. Базові операції з масивами	70
Лабораторна робота №11. Одновимірні масиви. Частина 2. Принципи обробки даних в одномірних масивах.....	74
Лабораторна робота №12. Двовірні масиви. Основи роботи з матрицями	78
ЛІТЕРАТУРА.....	83

ВСТУП

Розвиток економіки, промисловості, науки і техніки, сфери освіти сьогодні значною мірою залежить від масового запровадження та використання обчислювальної техніки. Це вимагає підготовки і перепідготовки фахівців з програмування і використання персональних комп'ютерів.

Вибір мови програмування C# пояснюється такими чинниками:

- простотою і природністю основних конструкцій мови, що дозволяє швидко її освоїти і створювати алгоритмічно складні програми;
- можливістю використання розвинених засобів подання структур даних, що забезпечує зручність роботи як з числовою, так і з символічною інформацією;
- відповідністю принципам об'єктно-орієнтованого програмування, що робить програми наочними;
- наявністю бібліотеки процедур і функцій для роботи як з текстовою, так і з графічною інформацією, що дозволяє створювати досить складні програми.

ВИМОГИ БЕЗПЕКИ ПІД ЧАС РОБОТИ В ЛАБОРАТОРІЇ ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

1. Ці правила є обов'язковими для всіх студентів та інших осіб, які працюють в лабораторіях постійно чи тимчасово.
2. Перевірка знань цих правил проводиться:
3. обслуговуючого персоналу – завідуючим лабораторією;
4. студентів – викладачами, які є керівниками лабораторних робіт.
5. Після перевірки знань цих правил, кожен з тих, хто працює в лабораторії ставить свій підпис в спеціальному журналі. Без цього підпису ніхто до роботи в лабораторії не повинен бути допущеним.
6. Дотримання правил з техніки безпеки повинно гуртуватися на високій свідомості всіх, хто працює в лабораторіях. Кожен, хто помітить порушення правил, а також несправність, яка являє собою небезпеку для людей і обладнання, повинен сповістити про це керівника.
7. Робота в лабораторії по виконанню конкретного завдання може проводитися тільки після ретельного ознайомлення студентів з обладнанням і роботою, та чіткого уявлення про те, які елементи установки будуть під напругою та дотик до яких є небезпечними у стані роботи.
8. Забороняється виконання ремонтних робіт на обладнанні, яке знаходиться під напругою.
9. Забороняється наближатися або торкатися до струмоведучих частин, що обертаються, або усувати несправності без відключення установок.
10. Забороняється проводити переключення в схемі, яка знаходиться під напругою.
11. Перевірку наявності напруги дозволяється проводити тільки за допомогою вольтметра.

12. Апарати управління та вимірювальні прилади слід розташовувати так, щоб було зручно вести спостереження за приладами, не перегинаючись через проводи та апарати.

13. У випадку виникнення будь-яких несправностей необхідно негайно вимкнути живлення установки та сповістити керівника занять про це.

14. Кнопки управління, рубильники встановлювати в легкодоступних місцях для швидкого виключення схеми.

15. У випадку припинення подачі електроенергії в лабораторію всі установки в лабораторії обов'язково вимикаються на робочих місцях.

16. В лабораторіях категорично забороняється:

- Палити в усіх приміщеннях, крім спеціально відведених для цього місць;
- Прокладати без дозволу постійні та тимчасові лінії;
- Користуватися побутовими електронагрівальними приладами;
- Користуватися зіпсованим електрообладнанням, саморобними запобіжниками, провідниками із зіпсованою ізоляцією та саморобними електросвітільниками;
- Проводити в непристосованих приміщеннях обмивку та фарбування деталей горючими рідинами та фарбниками;
- Зберігати паливно-мастильні матеріали, хімікати та інші горючі речовини;
- Загромаджувати проходи в лабораторіях;

17. Всі, хто працює в лабораторії повинні знати, де знаходиться аптечка з медикаментами для надання першої допомоги.

18. При ураженні людини електричним струмом треба негайно вимкнути напругу, надати першу допомогу та покликати лікаря.

19. Порятунком осіб, які постраждали, залежить від того, як швидко вони будуть звільнені від електричного струму та як швидко їм буде надано першу допомогу.
20. Першу допомогу необхідно надати негайно на місці події.
21. Переносити людину, яка постраждала в інше місце необхідно тільки в тих випадках, коли небезпека продовжує загрожувати або надання допомоги на місці неможливе.
22. При відсутності у постраждалого дихання, серцебиття, пульсу ніколи не треба ставити під сумнів необхідність першої допомоги, тому що при ураженні електричним струмом смерть часто буває несправжньою. Тільки лікар може дати висновок про смерть постраждалого.
23. До приїзду лікаря постраждалому необхідно надати допомогу і провести штучне дихання з дотриманням всіх правил надання першої допомоги.
24. Про випадок негайно треба сповістити керівництво кафедри, деканату та інституту.
25. Недотримання цих вимог не дозволяється. Якщо розпорядження суперечить діючим правилам, необхідно надати роз'яснення з приводу неухильною виконання цих правил і довести це до відома керівництва.

ЗАГАЛЬНІ ВИМОГИ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Підготовка до кожної роботи проводиться в поза аудиторний час. Студенти знайомляться із загальними відомостями, пишуть необхідні програми відповідно до отриманого варіанта індивідуального завдання.

Усі роботи виконуються на мові програмування C#. Під час занять студенти проводять тестування написаних програм, тобто запускають їх в інтегрованому середовищі розробки на персональних комп'ютерах, займаються налагодженням і виконують необхідні розрахунки.

Після виконання роботи студент оформляє звіт, який складається з таких розділів:

1. Назва, тема і мета роботи
2. Індивідуальне завдання
3. Текст програми
4. Результати виконання програми

Шаблон звіту про виконання лабораторної роботи наведено в додатку А. Робота оформлюється в друкованому вигляді на папері стандартного формату А4.

Під час захисту роботи необхідно відповісти на контрольні питання і вміти пояснити роботу програми.

ЛАБОРАТОРНА РОБОТА №1. ЗНАЙОМСТВО З СЕРЕДОВИЩЕМ VISUAL STUDIO. РОЗРОБКА ЕЛЕМЕНТАРНОЇ ПРОГРАМИ МОВОЮ ПРОГРАМУВАННЯ C#

Мета

Ознайомитися з інтегрованим середовищем розробки Visual Studio. Засвоїти структуру простої програми написаної мовою C#.

Стислі теоретичні відомості

Microsoft .NET — програмна технологія компанії Microsoft, яка є платформою для створення як звичайних додатків, так і веб- додатків. .NET-додатки можуть розроблятися і виконуватися як в середовищі операційних систем Microsoft, так і в інших включаючи Mac OS X, різні дистрибутиви Linux, Solaris, а також на мобільних пристроях iOS і Android (через API-інтерфейс MonoTouch).

C# – це об'єктно-орієнтована мова програмування, розроблена в 1998-2001 роках групою інженерів під керівництвом Андерса Хейлсберга в компанії Microsoft як мова розробки додатків для платформи Microsoft .NET. Синтаксис C# близький до C++ і Java. На сьогодні C# є флагманською мовою корпорації Microsoft для програмування в середовищі Windows.

Основним інтегрованим середовищем розробки, тобто комп'ютерною програмою, що допомагає програмістові розробляти нове програмне забезпечення, для C# є Visual Studio. Альтернативними середовищами програмування є SharpDevelop та MonoDevelop.

C # – не єдина мова, яка може використовуватися для побудови .NET-додатків. Середовище Visual Studio дає змогу використовувати п'ять мов, а саме – C #, Visual Basic, C ++/CLI, JavaScript і F#. Існують .NET-компілятори які призначені для мов Ruby, Python, Pascal тощо.

Visual Studio працює на платформі Windows і орієнтована на створення Windows- і веб-додатків, однак є можливість роботи і з консольними

додатками. При запуску консольного додатка операційна система створює так зване консольне вікно, через яке відбувається взаємодія з програмою. Консольні додатки найкраще підходять для вивчення мови, тому що в них використовується мінімум стандартних об'єктів. У першій частині курсу ми будемо створювати тільки консольні додатки, щоб зосередити увагу на базових властивостях мови C#.

Робота з вихідним кодом та іншими об'єктами програми у Visual Studio відбувається в контексті рішення. Рішення – це віртуальний контейнер найвищого рівня для всіх елементів розробки. Рішення може містити один або кілька проектів, а також файли різноманітних типів (текстові документи, зображення тощо), але не може містити всередині себе інші рішення.

Проект – це контейнер більш низького рівня, який завжди знаходиться всередині якогось рішення і використовується в якості організаційної одиниці для розміщення і групування файлів програми.

Для створення проекту слід після запуску Visual Studio в головному меню вибрати команду File > New > Project... (Рисунок 1)

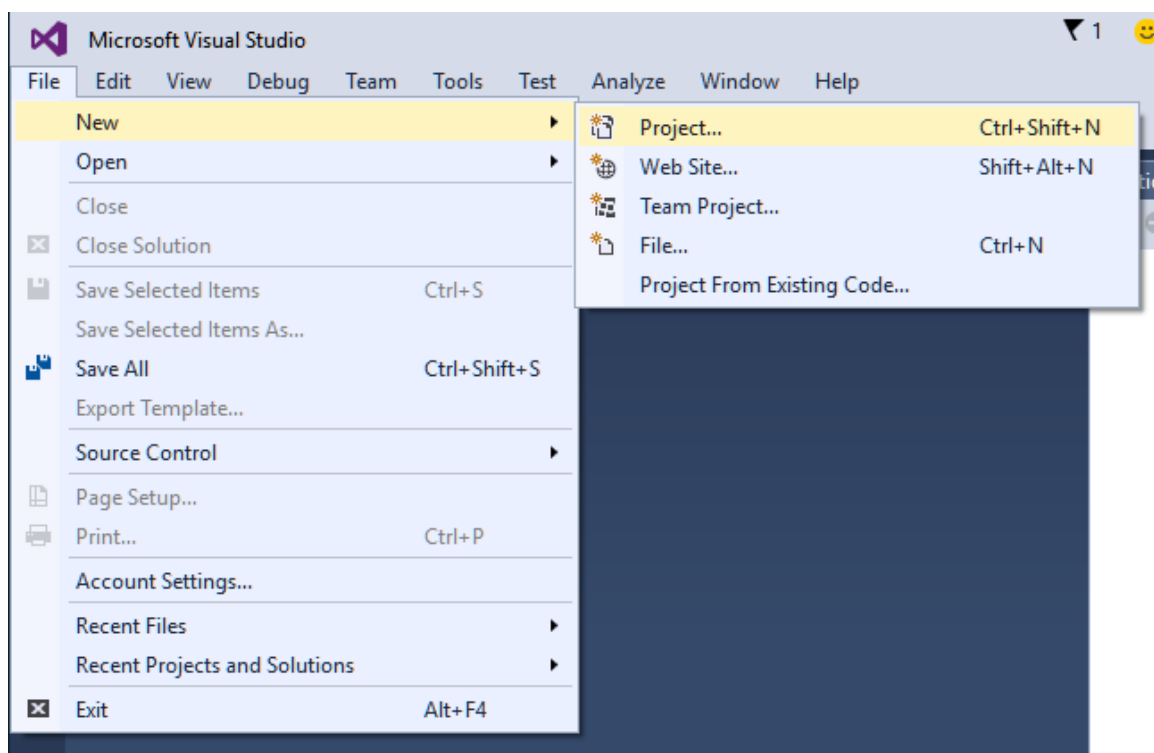


Рисунок 1 – Створення проекту за допомогою меню File

У лівій частині діалогового вікна потрібно вибрати пункт Visual C#, в правій – пункт Console Application (Рисунок 2). У полі Name можна ввести ім'я проекту, а в полі Location – місце його збереження на диску.

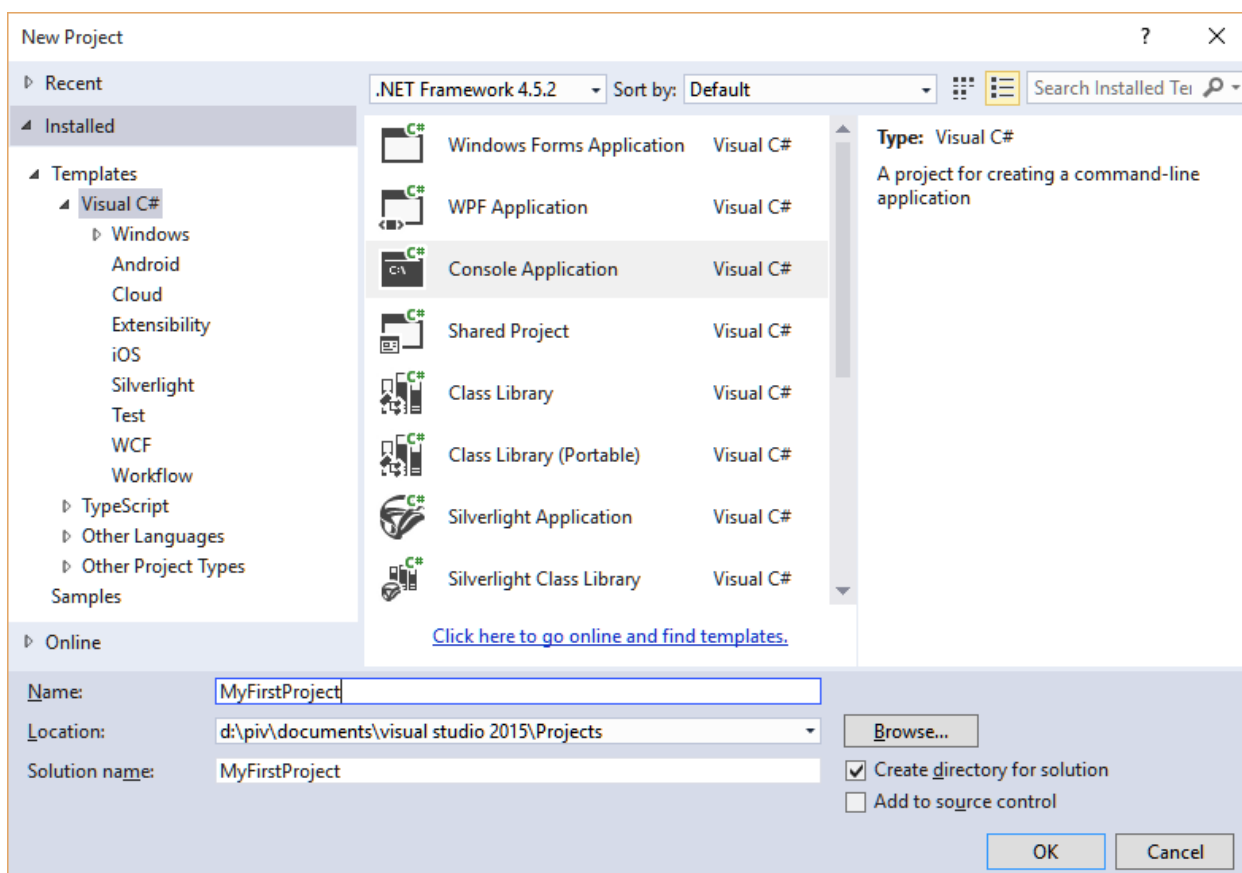


Рисунок 2 – Вибір типу проекту

У верхній частині екрана розташовується головне меню та панелі інструментів.

У верхній лівій частині екрана розташовується вікно управління проектом Solution Explorer (якщо воно не відображається, слід скористатися командою View > Solution Explorer головного меню). У вікні перераховані всі ресурси, що входять в проект. У нижній лівій частині екрана розташоване вікно властивостей Properties (якщо вікна не видно, скористайтеся командою View > Properties головного меню). У вікні властивостей відображаються найважливіші характеристики виділеного елемента.

Основний простір екрана займає вікно редактора, в якому розташовується текст програми, створений середовищем автоматично. Текст являє собою каркас, в який програміст додає код.

Для початку створимо новий проект консольного додатку з іменем MyFirstProject. Автоматично згенерований код програми має такий вигляд:

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

namespace MyFirstProject
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

Перші рядки містять директиву using, яка повідомляє компілятору, де він повинен шукати класи (типи), не визначені в даному просторі імен. За замовчуванням вказано стандартний простір імен System, де визначена основна частина типів середовища .NET.

Наступний рядок

```
namespace MyFirstProject
```

визначає простір імен додатку. За замовчаннями як ім'я простору імен вибирається ім'я проекту. Область дії простору імен визначається блоком коду між фігурними дужками. Простір імен забезпечує спосіб відокремлення одного набору імен від іншого. Імена, оголошені в одному просторі імен, не конфліктують з іменами, оголошеними в іншому просторі імен.

Слово class, розташоване в першому рядку тексту першої програми, відноситься до об'єктно орієнтованої частини мови, і буде детально розглянуто

пізніше. Слово `class` повинне бути присутнім у будь-якій програмі на C# хоча б один раз.

Фраза `static void Main ()` є заголовком методу `Main`. Наступний блок фігурних дужок обмежує тіло методу `Main`. Ім'я методу `Main` не може бути змінено, оскільки система саме з цієї підпрограми починає виконання додатку (так звана точка входу).

Додамо в тіло методу `Main` два рядки:

```
static void Main(string[] args)
{
    Console.WriteLine("Hello world!");
    Console.ReadLine();
}
```

Рядок

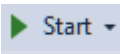
```
Console.WriteLine("Hello world!");
```

виводить повідомлення на екран. Оператор закінчується символом «;», який є обов'язковим для завершення більшості конструкцій C#.

Рядок

```
Console.ReadLine();
```

вказує програмі очікувати введення даних з клавіатури та натиснення клавіші `Enter`.

Скопілюємо і запустимо додаток на виконання. Для цього необхідно вибрати пункт меню `Debug > Start Debugging`, натиснути кнопку  на панелі інструментів або натиснути клавішу `F5`. В результаті виконання програми з'явиться вікно:

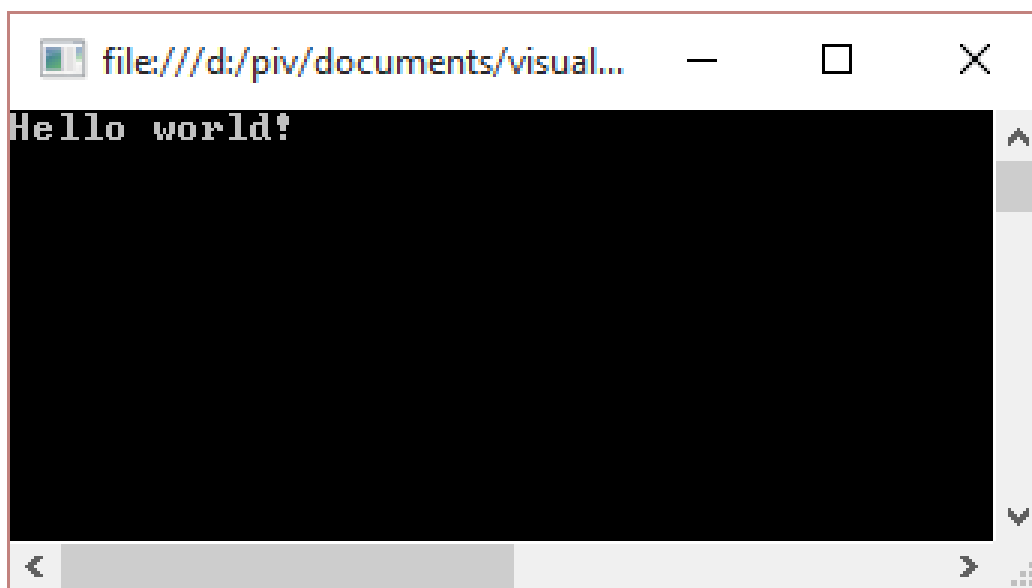


Рисунок 3 – Результат виконання програми

Часто в програмах містяться помилки. Деякі з цих помилок автоматично виявляються засобами Visual Studio. Після виявлення помилки процес компіляції і запуску програми переривається і видається відповідне діагностичне повідомлення для користувача.

Розглянемо автоматичне виявлення помилок на окремому прикладі. Для цього створимо консольний додаток раніше описаним способом. У вікні редактора коду наберіть такий текст програми

```
static void Main(string[] args)
{
    Console.WriteLine("Hello world!");
    Console.ReadLine();
}
```

Спробуйте скомпілювати і запустити додаток виконання. Після цього у вікні Error List повинні з'явитися повідомлення про виявлені помилки (Рисунок 4).

У відкритому вікні Error List двічі клацніть на знайденій помилці і переконайтеся, що при цьому в редакторі коду курсор переміститься на рядок з помилкою. Повідомлення про помилку містить номер помилки і короткий її опис.

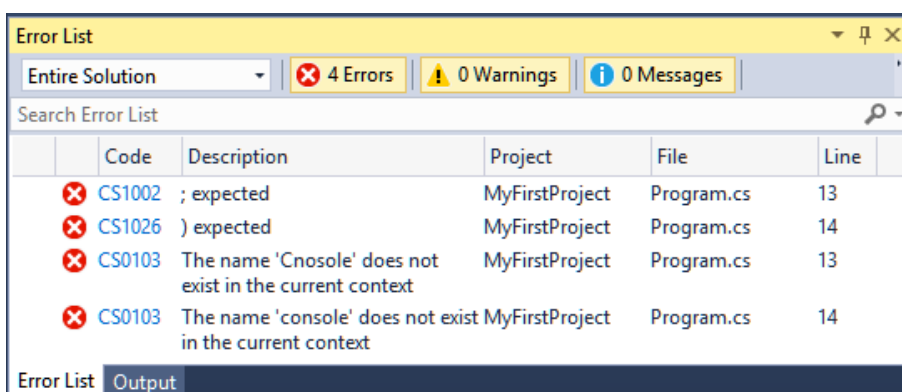


Рисунок 4 – Вікно зі списком помилок

Перегляньте весь список виявлених компілятором помилок і ознайомтеся з їх описом. Використовуючи отриману інформацію, виправте усі помилки. Якщо всі синтаксичні помилки були виправлені, то компіляція пройде успішно і програма запуститься.

Важливою частиною будь-якої мови програмування є коментарі, які дозволяють зручно описувати та пояснювати різні ділянки коду. У С# використовуються традиційні коментарі в стилі С – однорядкові (`// ...`) та багаторядкові (`/*... */`). Все, що знаходиться в коментарі (для однорядкового – від `//` до кінця рядка; для багаторядкового – весь текст розташований між `/*` та `*/`), ігнорується компілятором:

```
static void Main(string[] args)
{
    //Вивід повідомлення на екран
    Console.WriteLine("Hello world!");
    //Очікування вводу
    Console.ReadLine();
}
```

Розглянемо інші функції головного меню середовища Visual Studio.

Меню File

Команда **Open** відкриває діалогове вікно в якому вибирається файл для відкриття.

Команда **Save** записує проект на диск.

Команда **Save as** – зберігання з перейменуванням.

Команда **Save all** – зберігає всі файли в відкритих вікнах.

Команда **Print** – роздруковує активне вікно редагування.

Меню Edit.

Команда **Cut** – видаляє виділений текст та розміщує його в Clipboard.

Команда **Copy** – не видаляючи з тексту виділеного фрагменту, копіює його в Clipboard.

Команда **Paste** – вставляє текст з Clipboard до програми в позиції курсору.

Робоче завдання

Вміти створювати, зберігати, відкривати, запускати програми. Вивчити структуру програми на мові C#. Знати розділи з яких складається програма.

Хід роботи

1. Запустити інтегроване середовище розробки Visual Studio.
2. Ознайомитись з інтерфейсом програми.
3. Ознайомитись з меню File. Навчитися створювати, зберігати та відкривати проекти.
4. Ознайомитись з меню Edit. Навчитися копіювати, видаляти, вирізати текст.
5. Ознайомитись з меню View.
6. Ознайомитись з меню Debug. Навчитися виконувати програму та переглядати результати роботи програми.
7. Ознайомитись з меню Window. Навчитися переходити між вікнами, закривати вікна, переміщати вікна.

8. Набрати програму задану викладачем.
9. Розібрати набрану програму.
10. Виконати програму.
11. Набрати програму з помилками.
12. Ознайомитися з описом помилок та виправити їх.

Індивідуальне завдання

Написати програму яка виведе на екран ваше прізвище та ім'я, а також назву навчальної групи.

Контрольні питання

1. Що таке алгоритм?
2. Що таке програма?
3. Яка відмінність між програмою та алгоритмом?
4. На які етапи можна поділити вирішення програмістської задачі?
5. Що таке коментар до коду програми. Для чого він призначений. Як у мові програмування C# позначається коментар до програмного коду?
6. Як створити новий проект у середовищі Visual Studio?
7. Як зберігати проект у середовищі Visual Studio?
8. Як відкрити існуючий проект у середовищі Visual Studio?
9. Як виконати пошук тексту у файлі в середовищі Visual Studio?
10. Як запустити програму на виконання в середовищі Visual Studio?
11. Що таке компілятор та інтерпретатор? Яка між ними різниця?
12. Хто із якою метою розробив мову програмування C#?

ЛАБОРАТОРНА РОБОТА №2. БАЗОВІ ТИПИ ДАНИХ. ФОРМАТУВАННЯ КОНСОЛЬНОГО ВИВЕДЕННЯ

Мета

Ознайомитися з основними типами даних мови програмування C#. Засвоїти роботу операторів вводу, виводу.

Стислі теоретичні відомості

Типи даних

Дані, з якими працює програма, зберігаються в оперативній пам'яті. Компілятору необхідно точно знати, скільки місця вони займають, як саме закодовані і які дії з ними можна виконувати. Все це задається при описі даних за допомогою типу.

Тип даних однозначно визначає:

- внутрішнє подання даних, а отже, і діапазон їх можливих значень;
- допустимі дії над даними (операції і функції).

Наприклад, цілі і дійсні числа, навіть якщо вони займають однаковий обсяг пам'яті, мають зовсім різні діапазони можливих значень; цілі числа можна множити один на одного, а, наприклад, символи – не можна. Компілятор використовує інформацію про тип при перевірці допустимості описаних в програмі дій.

Основні дані, що обробляються програмою, – це змінні і константи.

Змінні – це дані, які можуть змінювати свої значення в процесі виконання програми.

Всі змінні в мові C# повинні бути описані явно. Це означає, що, по-перше, на початку кожної програми або функції Ви повинні привести список імен (ідентифікаторів) всіх використовуваних змінних, а по-друге, вказати тип кожної з них.

Оператор опису складається з назви типу і списку імен змінних, розділених комою. Наприкінці обов'язково повинна стояти крапка з комою. При описі можливе завдання початкового значення змінної.

Ім'я змінної – є ідентифікатором – тобто будь-яка послідовність великих і малих літер англійського алфавіту, цифр і символу підкреслення '_'.

В мові C# визначені ключові слова для фундаментальних типів даних, які застосовуються для опису змінних. Їх перелік представлений в Таблиця 2.1

Внутрішні типи даних C #

Таблиця 2.1 Внутрішні типи даних C #

Позначення	Системний тип	Діапазон	Опис
bool	System.Boolean	true або false	Ознака істинності чи хибності
sbyte	System.SByte	від -128 до 127	8-бітне число зі знаком
byte	System.Byte	від 0 до 255	8-бітне число без знака
short	System.Int16	від -32768 до 32767	16-бітне число зі знаком
ushort	System.UInt16	від 0 до 65535	16-бітне число без знака
int	System.Int32	від -2147483648 до 2147483647	32-бітне число зі знаком
uint	System.UInt32	від 0 до 4294967295	32-бітне число без знака
long	System.Int64	від -9223372036854775808 до 9223372036854775807	64-бітне число зі знаком
ulong	System.UInt64	від 0 до 18446744073709551615	64-бітне число без знака
char	System.Char	від U +0000 до U +ffff	Одиночний 16-бітний символ Unicode
float	System.Single	від $-3.4 \cdot 10^{38}$ до $+3.4 \cdot 10^{38}$	32-бітне число з плаваючою точкою
double	System.Double	від $\pm 5.0 \cdot 10^{-324}$ до $\pm 1.7 \cdot 10^{308}$	64-бітне число з плаваючою точкою
decimal	System.Decimal	(від $-7.9 \cdot 10^{28}$ до $7.9 \cdot 10^{28}$) / $(10^{\text{от}0\text{во}28})$	128-бітне число зі знаком
string	System.String	Обмежений обсягом системної пам'яті	Набір символів Unicode

Числові типи .NET підтримують властивості MaxValue та MinValue, які надають інформацію про діапазоні значень, що зберігаються в конкретному типі. Крім MinValue і MaxValue, кожен числовий тип може визначати додаткові корисні властивості. Наприклад, тип System.Double дозволяє

отримувати значення для епсилон (точності обчислень) та нескінченності. З метою ілюстрації розглянемо такий код:

```
static void Main(string[] args)
{
    Console.WriteLine("Data type Properties:");
    Console.WriteLine("int range: {0} – {1}", int.MinValue, int.MaxValue);
    Console.WriteLine("double range: {0} – {1}", double.MinValue, double.MaxValue);
    Console.WriteLine("double.Epsilon: {0}", double.Epsilon);
    Console.WriteLine("Infinities: {0} – {1}", double.NegativeInfinity, double.PositiveInfinity);
    Console.ReadLine();
}
```

При введенні вихідних даних відбувається перетворення з зовнішньої форми представлення у внутрішню, обумовленим типом змінних. Змінні, що утворюють список введення, можуть належати або до цілого, або до дійсного, або до символьного типу. Значення вихідних даних можуть відокремлюватися один від одного пробілами і натисканням клавіш табуляції і <Enter>.

Тип Console визначає методи, які дозволяють виконувати консольне введення і виведення:

- Метод WriteLine() дозволяє вивести рядок тексту (включаючи символ повернення каретки – тобто перевод курсора на новий рядок).
- Метод Write() виводить текст без символу повернення каретки – тобто без переводу курсору на новий рядок.
- Метод ReadLine() дозволяє отримати інформацію з потоку введення аж до натискання клавіші <Enter>.
- Метод Read() використовується для введення одиночного символу.

Для демонстрації базового введення-виведення із застосуванням класу Console розглянемо програму, яка запитує у користувача його прізвище та рік народження (який для простоти будемо вважати рядком, а не числом):

```

static void Main(string[] args)
{
    Console.Write("Введіть ваше прізвище: ");
    string userName = Console.ReadLine();
    Console.Write("Введіть рік вашого народження: ");
    string userBirthYear = Console.ReadLine();
    Console.WriteLine("Привіт {0}! Ви народилися в {1} році.",
        userName, userBirthYear);
    Console.ReadLine();
}

```

При виведенні рядку з сегментами даних, значення яких залишаються невідомими до етапу виконання, всередині нього допускається вказувати заповнювач, використовуючи синтаксис у вигляді фігурних дужок. Під час виконання програми на місце кожного такого заповнювача підставляється передане в `Console.WriteLine()` значення.

Перший параметр методу `WriteLine()` є строковим літералом, який містить заповнювачі виду `{0}`, `{1}`, `{2}` і т.д. Порядкові номери заповнювачів у фігурних дужках завжди починаються з 0. Інші параметри `WriteLine()` – це значення, які повинні підставлятися на місці заповнювачів.

Один і той же заповнювач може повторюватися в межах заданої рядки. Кожен заповнювач допускається розміщувати в будь-якому місці всередині строкового літерала, і не обов'язково, щоб заповнювачі вказувалися в зростаючій послідовності своїх номерів.

Якщо для числових даних потрібно більш складне форматування, кожен заповнювач може додатково містити різноманітні символи форматування. Найбільш корисні з них представлено в Таблиця 2.2.

Таблиця 2.2 Символи для форматування даних

Символ форматування	Опис
---------------------	------

C або c	Використовується для форматування грошових значень. За замовчуванням значення доповнюється символом локальної валюти (наприклад, знаком долара (\$) для культури US English)
D або d	Використовується для форматування десяткових чисел. З цим символом можна також вказувати мінімальну кількість цифр для представлення значення
E або e	Використовується для експоненціального представлення. Регістр цього символу вказує, в якому регістрі повинна бути представлена експоненціальна константа – у верхньому (E) або в нижньому (e)
F або f	Використовується для форматування чисел з фіксованою точкою. З цим символом можна також вказувати мінімальну кількість цифр для представлення значення
G або g	Означає general (загальний). Цей символ може використовуватися для подання чисел у форматі з фіксованою точкою або експоненційному форматі
N або n	Використовується для базового числового форматування (з комами)
X або x	Використовується для шістнадцятирічного форматування. У разі символу X у верхньому регістрі шістнадцяткове подання буде містити символи верхнього регістру

Ці символи форматування додаються у вигляді суфіксів до заповнювачів після двокрапки (наприклад, {0:3}, {1:d}, {2:X}).

Робоче завдання

Мати поняття змінна, константа, оператор, операнд. Засвоїти роботу операторів вводу, виводу. Вивчити ієрархію типів даних мови C#. Вивчити властивості простих типів даних.

Хід роботи

Для кожного підпункту завдання необхідно написати окрему програму. Необхідно описати вказані змінні, ввести їх з клавіатури та вивести на екран результати обрахунків.

Індивідуальне завдання

Варіант 1

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	

	b	int	$a+b-c$	
	c	int	$a-b-c$	
2	a	int	$a*b$	
	b	double	a/b	
3	a	long	$a*x*t$	3 знаки після коми
	x	double	$a*(a+x)/t$	1 знак після коми
	t	double	$a/t*(a+x/t)$	6 знаків після коми

Варіант 2

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	$a-b$	
	b	long	$a+b+c$	
	c	byte	$a-(b-c)$	
2	a	double	$a*b*b$	
	b	double	$a/(b-a)$	
3	a	long	$a*x*y$	1 знак після коми
	x	int	$(a+x)/y$	2 знаки після коми
	y	double	$a/(y*(a+x/y))$	10 знаків після коми

Варіант 3

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	long	$a+b$	
	b	int	$a+b-c$	
	c	byte	$a+(b-c)+(a-b)$	
2	a	int	$a*b$	
	b	double	a/b	5 знаків після коми
3	x	long	$z*x/y$	2 знаки після коми
	z	double	$z*(z+x)/y$	
	y	double	$(y+x/z)/z$	3 знаки після коми

Варіант 4

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	double	a-b-c	
2	a	long	a*b-b	
	b	double	a/b	
3	a	long	a*x*t+a	0 знаків після коми
	x	long	a+x/t	1 знак після коми
	t	double	a*a+x/t	7 знаків після коми

Варіант 5

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b-(a+b+a)	
	b	byte	-a+b-c	
	c	int	a-b+c	
2	a	byte	a*b+b	3 знаки після коми
	b	double	a/b-a	
3	a	double	-a*x*t	3 знаки після коми
	x	double	a*(a-x)/t	1 знак після коми
	t	double	a/t*(a+x/t)/(a*t)	6 знаків після коми

Варіант 6

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+c-3*f-1	
	f	short	f+(b-c+2*f)	
	c	int	c-f+a	
2	a	byte	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	a/t*(a+x/t)	6 знаків після коми

Варіант 7

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	byte	a+b-c	
	c	int	a-b-c	
2	a	double	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	2 знак після коми
	t	byte	a/t*(a+x/t)	5 знаків після коми

Варіант 8

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	-a-b	
	b	byte	-a+b	
	c	byte	a-b+c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	int	a*(a+x)/t	2 знаки після коми
	t	double	a/t*(a+x/t)	6 знаків після коми

Варіант 9

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	short	a+b	
	b	int	a-b	
	c	long	a+b-c	
2	a	int	a*b/5	6 знаків після коми
	b	double	a/b	
3	a	long	a*x*t/3	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	a/t*(a+x/t)	

Варіант 10

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	short	a+b-c	
	c	long	a-b-c	
2	a	byte	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	long	a*(a+x)/t	1 знак після коми
	t	double	a/t*(a+x/t)	7 знаків після коми

Варіант 11

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	long	a+b	
	b	int	a+b-c	
	c	short	a-b+c+1	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	2 знаки після коми
	t	double	a/t*(a+x/t)	10 знаків після коми

Варіант 12

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	short	a+b-c	
	c	int	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	2/a*x*t	4 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	a/t*(a+x/t)	5 знаків після коми

Варіант 13

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	short	a+b-c	
	c	int	a-b-c	
2	a	int	1/(a*b)	
	b	double	a/b	3 знаки після коми
3	a	byte	a*x*t	
	x	long	a*(a+x)/t	3 знак після коми
	t	double	a/t*(a+x/t)	1 знаків після коми

Варіант 14

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	int	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	a/t*(a+x/t)/3	6 знаків після коми

Варіант 15

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	short	a-b-c	
2	a	int	a*b	1 знак після коми
	b	double	a/b	
3	a	short	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	
	t	double	5/t*(a+x/t)	6 знаків після коми

Варіант 16

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	short	a+b-c	
	c	int	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	a/t*(a+x/t)	6 знаків після коми

Варіант 17

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	int	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	1/t*(a+x/t)	6 знаків після коми

Варіант 18

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	int	a-b-c	
2	a	int	a*b-1	
	b	double	a/b+5	2 знаки після коми
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	7 знаків після коми
	t	double	a/t*(a+x/t)	

Варіант 19

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	long	a+b	
	b	int	a+b-c	
	c	byte	a-b-c	
2	a	int	a*b	2 знаки після коми
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a-x)/t	1 знак після коми
	t	byte	a/t*(a-x*x)	

Варіант 20

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	int	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	a/t*(a+x/t)	6 знаків після коми

Варіант 21

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	int	a-b-c	
2	a	int	a*b-a	
	b	double	a/b+b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	a/x*(a+x/t)	6 знаків після коми

Варіант 22

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	long	a+b	
	b	int	a+b-c	
	c	shortint	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	double	a/t*(a+x/t)	6 знаків після коми

Варіант 23

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	shortint	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	short	a/t*(a+x/t)	6 знаків після коми

Варіант 24

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	long	a+b-c	
	c	int	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	byte	a*x*t	3 знаки після коми
	x	double	a*(a+x)/t	1 знак після коми
	t	short	a/t*(a+x/t)	6 знаків після коми

Варіант 25

п.п.	Змінна	Тип	Вивести на екран	Примітка
1	a	int	a+b	
	b	int	a+b-c	
	c	int	a-b-c	
2	a	int	a*b	
	b	double	a/b	
3	a	long	a*x*t	4 знаки після коми
	x	double	a*(a+x)/t	2 знак після коми
	t	double	a/t*(a+x/t)	4 знаків після коми

Контрольні питання

1. Що таке змінна?
2. Що таке ідентифікатор?
3. Яка різниця між змінною та ідентифікатором?
4. Що таке оператор?
5. Як задається тип змінної?
6. Що таке тип даних?
7. Назвіть цілочисельні типи даних.
8. Назвіть дійсні типи даних.
9. Які значення може приймати значення логічного типу даних?
10. Яка різниця між типом int та uint?
11. Які значення може приймати змінна типу byte?
12. Яка різниця між типом long та char?
13. Як виконується виведення даних? Наведіть приклад.
14. Як виконується введення даних? Наведіть приклад.

ЛАБОРАТОРНА РОБОТА №3. ПРОГРАМУВАННЯ АРИФМЕТИЧНИХ ВИРАЗІВ

Мета

Отримати досвід програмування арифметичних виразів. Ознайомитися з основними арифметичними операторами та вбудованими математичними функціями.

Стислі теоретичні відомості

Вираз – це правило обчислення значення.

Вираз складають операнди, об'єднані знаками операцій. Операндами найпростішого вирази можуть бути константи, змінні і виклики функцій.

За кількістю операндів що беруть участь в одній операції, операції діляться на унарні, бінарні і тернарний.

Оператори мови C# представлено в Таблиця 3.1 Оператори C#.

Таблиця 3.1 Оператори C#

Категорія	Знак операції	Назва
Первинні	x()	Виклик методу або делегата
	x[]	Доступ до елемента
	x++	Постфіксний інкремент
	x--	Постфіксний декремент
	new	Виділення пам'яті
	typeof	Отримання типу
Унарні	+	Унарний плюс
	-	Унарний мінус (арифметичне заперечення)
	!	Логічне заперечення
	~	Порозрядне заперечення
	++x	Префіксний інкремент
	--x	Префіксний декремент
	(тип)x	Перетворення типу
Мультиплікативні	*	Множення
	/	Ділення
	%	Залишок від ділення
Адитивні	+	Додавання
	-	Віднімання

Категорія	Знак операції	Назва
Зсуву	<<	Зсув вліво
	>>	Зсув вправо
Відношення та перевірки типу	<	Менше
	>	Більше
	<=	Менше або дорівнює
	>=	Більше або дорівнює
	is	Перевірка приналежності типом
	as	Приведення типу
Перевірки на рівність	==	дорівнює
	!=	Не дорівнює
Порозрядні логічні	&	Поразрядна кон'юнкція (І)
	^	Порозрядне виключаюче АБО
		Поразрядна диз'юнкція (АБО)
Умовні логічні	&&	Логічне І
		Логічне АБО
Умовний	?	Умовна операція
Присвоєння	=	Присвоєння
	*=	Множення з присвоєнням
	/=	Ділення з присвоєнням
	%=	Залишок від ділення з присвоєнням
	+=	Додавання з присвоєнням
	-=	Віднімання з присвоєнням
	<<=	Зсув вліво з присвоєнням
	>>=	Зсув вправо з присвоєнням
	&=	Порозрядне І з присвоєнням
	^=	Порозрядне виключає АБО з присвоєнням
	=	Порозрядне АБО з присвоєнням

Операції у виразі виконуються в певному порядку відповідно до пріоритетів, як і в математиці. У Таблиця 3.1 Оператори C# операції розташовані за спаданням пріоритетів, рівні пріоритетів розділені в таблиці горизонтальними лініями.

Для зміни порядку виконання операцій використовуються круглі дужки, рівень їх вкладеності практично не обмежений.

При обчисленні виразів може виникнути необхідність у перетворенні типів. Якщо операнди, що входять у вираз, одного типу і операція для цього типу визначена, то результат виразу буде мати той же тип.

Якщо операнди різного типу або операція для цього типу не визначена, перед обчисленнями автоматично виконується перетворення типу за правилами, що забезпечує приведення коротших типів до більш довгих для збереження значущості і точності.

Якщо неявного перетворення з одного типу в інший не існує, програміст може задати явне перетворення типу за допомогою операції (тип) x.

Операції інкремента (++) і декремента, що називають також операціями збільшення та зменшення на одиницю, мають дві форми запису – префіксну, коли знак операції записується перед операндом, і постфіксну.

У префіксній формі спочатку змінюється операнд, а потім його значення стає результуючим значенням виразу, а в постфіксній формі значенням виразу є початкове значення операнда, після чого він змінюється.

Арифметичне заперечення (унарний мінус -) змінює знак операнда на протилежний. Стандартна операція заперечення визначена для типів int, long, float, double і decimal.

Логічне заперечення (!) Визначено для типу bool. Результат операції – значення false, якщо операнд дорівнює true, і значення true, якщо операнд дорівнює false.

Порозрядне заперечення (~) інвертує кожен розряд в двійковому представленні операнда типу int, uint, long або ulong.

Операції зсуву (<< та >>) застосовуються до цілочисельних операндів. Вони зсувають двійкове представлення першого операнда вліво або вправо на кількість двійкових розрядів, задане другим операндом. При зсуві вліво (<<) звільнені розряди обнуляються. При зсуві вправо (>>) звільнені біти заповнюються нулями, якщо перший операнд беззнакового типу (тобто виконується логічний зсув), і знаковим розрядом – в іншому випадку (виконується арифметичний зсув).

Операції відношення (<, <=, >, >=, !=) порівнюють перший операнд з другим. Операнди повинні бути арифметичного типу. Результат операції – логічного типу.

Порозрядні логічні операції (&, |, ^) застосовуються до цілочисельних операндів і працюють з їх двійковими представленням. При виконанні операцій операнди зіставляються побітно (перший біт першого операнда з першим бітом другого, другий біт першого операнда з другим бітом другого і т. Д.). Стандартні операції визначені для типів int, uint, long і ulong.

При поразрядній кон'юнкції, або порозрядному І (операція позначається &), біт результату дорівнює 1 тільки тоді, коли відповідні біти обох операндів рівні 1. При поразрядній диз'юнкції, або порозрядному АБО (операція позначається |), біт результату дорівнює 1 тоді, коли відповідний біт хоча б одного з операндів дорівнює 1.

При порозрядному виключає АБО (операція позначається ~) біт результату дорівнює 1 тільки тоді коли відповідний біт тільки одного з операндів дорівнює 1.

Умовні логічні операції І (&&) і АБО (||) найчастіше використовуються з операндами логічного типу. Результатом логічної операції є true або false.

Результат операції логічне І має значення true, тільки якщо обидва операнда мають значення true. Результат операції логічне АБО має значення true, якщо хоча б один з операндів має значення true.

Операції присвоювання (=, +=, -=, *= і т. д.) задають нове значення змінної.

Механізм виконання операції присвоювання такий: обчислюється вираз і його результат заноситься в пам'ять за адресою, що визначається ім'ям змінної, що знаходиться зліва від знака операції. Значення, що раніше зберігалось в цій області пам'яті, втрачається.

У складних операціях присвоєння (+=, -=, *= і т. д.) при обчисленні виразу, що стоїть в правій частині, використовується значення з лівої

частини. Наприклад, при додаванні з присвоюванням до другого операнд додається перший, і результат записується в перший операнд, тобто вираз

```
a + = 5;
```

є більш компактною формою запису виразу

```
a += a + 5;
```

У виразах часто використовуються математичні функції, наприклад синус або піднесення до степені. Вони реалізовані в класі `Math`, визначеному в просторі імен `System`. Опис методів і полів наведено в Таблиця 3.2.

Таблиця 3.2 Основні поля і методи класу `Math`

Ім'я	Опис	Пояснення
Abs	Модуль	$ x $ записується як Abs (x)
Acos	Арккосинус	Acos (double x)
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
Ceiling	Округлення до більшого цілого	Ceiling (double x)
Cos	Косинус	Cos (double x)
Cosh	Гіперболічний косинус	Cosh (double x)
E	База натурального логарифма (число e)	2,71828182845905
Exp	Експонента	e^x записується як Exp (x)
Floor	Округлення до меншого цілого	Floor (double x)
IEEE Remainder	Залишок від ділення	IEEERemainder (double x, double y)
Log	Натуральний логарифм	$\text{Log}_e x$ записується як Log (x)
Log10	Десятковий логарифм	$\text{log}_{10} x$ записується як Log10 (x)
Max	Максимум з двох чисел	Max (x, y)
Min	Мінімум з двох чисел	Min (x, y)
PI	Значення числа π	3,14159265358979
Pow	Степінь	x^y записується як Pow (x, y)
Round	Округлення	Round (3.1) дасть в результаті 3, Round (3.8) дасть в результаті 4
Sign	Знак числа	-

Ім'я	Опис	Пояснення
Sin	Синус	Sin (double x)
Sinh	Гіперболічний синус	Sinh (double x)
Sqrt	Квадратний корінь	записується як Sqrt (x)
Tan	Тангенс	Tan (double x)
Tanh	Гіперболічний тангенс	Tanh (double x)

Робоче завдання

Навчитися програмувати арифметичні вирази на мові програмування C#. Засвоїти роботу вбудованих математичних функцій класу Math.

Хід роботи

Написати програму на мові C#, яка розраховує значення виразів по заданих арифметичних формулах. Дані для розрахунку вводяться з клавіатури. Результати розрахунку вивести на екран з 10-ма знаками після коми.

Індивідуальне завдання

№ вар.	Вираз 1	Вираз 2	Вираз 3
1	$y = \cos 2x$	$y = \sin^2 x + \cos^2 x$	$y = \sqrt[3]{x} + e^{tgx^a} + x^{4\pi}$
2	$y = x^3 + 1$	$y = \arctg x $	$y = \arctg^x(\sqrt[n+1]{x - \log_5 x})$
3	$y = \sin 4x$	$y = \cos^2 e^x$	$y = \left \frac{ctg e^{\sin x^2}}{\ln x } \right $
4	$y = \sqrt{\cos x}$	$y = \cos x + 1 $	$y = \arcsin(x + a^{100}) \lg^{n+1} \sqrt{x+a}$
5	$y = \ln(x + 1)$	$y = \sin 2x$	$y = \arccos \frac{tg a}{\log_a(x + \sqrt[n-1]{ a-1 })}$
6	$y = \cos x + \sin x $	$y = \cos x^3$	$y = \frac{tg x}{e^{n+1}} + \left \frac{a+1}{x-1} \right $
7	$y = \sin 3x + x$	$y = \sin e^x$	$y = \frac{ a-x }{a+x^{n+1}} + \arcsin^4 \frac{x-a}{x+a}$

№ вар.	Выраз 1	Выраз 2	Выраз 3
8	$y = \cos \ln x$	$y = \arctg x $	$y = \log_{x^3-x^2+x} (x^4 + x^3 + x^2 + x) + \sin^3 x^3 $
9	$y = \cos^2 x$	$y = \cos 2x - \sin x$	$y = \frac{\arctg \frac{\arcsin x^2}{ a+x-1 }}{e^{\ln x + a^{\frac{x}{n+1}}}}$
10	$y = x^2 + \cos x $	$y = \sin e^x$	$y = \sqrt[n+1]{\frac{\lg \left x + \frac{e^{x+1}}{a-5} \right }{\operatorname{tg}^x (xa) + x^a}}$
11	$y = \sin 5x$	$y = \cos x + \sin x $	$y = \arccos x^{\arcsin \frac{x+1}{\operatorname{tg} x}}$
12	$y = \arctg x + 1 $	$y = \arctg x + rr$	$y = \frac{\log_{x+1} (x + a + 1)}{\sqrt[n]{\sin x + \cos a^e}}$
13	$y = \cos x^3$	$y = \cos x + 1 $	$y = x + x^{\arcsin x^{\ln a}} - \frac{ a - \operatorname{ctg} x }{a + \cos x^{100}}$
14	$y = \sin e^x$	$y = \cos^2 e^x$	$y = \frac{\log_{x^2} (a + x)}{\sqrt{x - \frac{\arcsin x}{\arccos^5 (x + a^2)}}}$
15	$y = \arctg x + rr$	$y = \cos x^3$	$y = e^{\frac{x + \arcsin e^x}{\operatorname{tg} \left x + \frac{a-1}{a+1} \right }}$
16	$y = \sqrt{x + \ln x}$	$y = \cos 2x$	$y = \frac{\ln \frac{x - e}{e^{\pi+x-\sqrt{x}}}}{\arccos (x + a \operatorname{tg} x)^{\sqrt{x}}}$
17	$y = \cos 2x - \sin x$	$y = x^3 + 1$	$y = \left e^{ x^e-10 } + \frac{\operatorname{tg} \frac{x+5}{a-\pi}}{\sqrt[e]{\pi^x}} \right $
18	$y = \ln x + e^{ x }$	$y = \cos \ln x$	$y = x^{ax} + \frac{\log_3 (x - \sqrt{a})}{\arccos e^{x+a-\sqrt{x-a}}}$
19	$y = \sin^2 x + \cos^2 x$	$y = \cos \ln x$	$y = \frac{\operatorname{ctg} (x - \sqrt{a-x+5})^e}{ a + \operatorname{tg}^{100} (x+a) } + \lg (x - \sqrt[3]{a})$
20	$y = -x^3 + 3x^2 + 3$	$y = \cos^2 x$	$y = \sqrt[n+1]{\arcsin (x + x^2 + \sqrt[n-1]{x})} + \frac{e^{\pi+x}}{\operatorname{ctg} \log_5 (x+a)}$

№ вар.	Вираз 1	Вираз 2	Вираз 3
21	$y = \cos 2x - \sin^3 x$	$y = x^2 + \cos x $	$y = \ln^2 \arccos^3 \frac{e^x}{a + \frac{ x + \sqrt{x} }{x}}$
22	$y = \operatorname{arctg} x $	$y = \sqrt{\cos x}$	$y = \frac{x + 1}{\operatorname{arctg} \ln \arccos x^{\frac{e + \frac{x}{a + \sqrt{x}}}} + \log_{a+x} a - x}$
23	$y = \cos^2 e^x$	$y = \cos x^3$	$y = \left x - \frac{x + \sqrt[n]{ a + \pi^{x + \cos a} }}{e + e^e + \operatorname{arctg} x^{e+x-1}} \right $
24	$y = \cos x + 1 $	$y = \sin 3x + x$	$y = \operatorname{arcsin}^{\arccos x^{x+1}} \left(\frac{\sqrt[n+1]{x}}{a + x} - 5x^{100} \right)$
25	$y = \sin 2x$	$y = \sin 5x$	$y = \log_{ a+x } \frac{(a^2 - x^2)(\cos a + \sin^2 \sqrt{x-a})}{\sqrt{\pi} + \ln x + \cos e^{\pi+x+a}}$

Контрольні питання

1. Назвіть арифметичні операції які є в C#.
2. Розташуйте оператори мови C# за пріоритетом.
3. За допомогою яких операторів можна збільшити або зменшити змінної на 1?
4. Назвіть вбудовані тригонометричні функції які є в C#?
5. За допомогою якої функції можна отримати квадратний корінь числа?
6. Виконайте перетворення арифметичного виразу у вираз на мові C#.

ЛАБОРАТОРНА РОБОТА №4. РОБОТА З ТЕКСТОВИМИ РЯДКАМИ. ЧАСТИНА 1. БАЗОВІ ОПЕРАЦІЇ З РЯДКАМИ

Мета

Засвоїти поняття текстових рядків. Вивчити основні алгоритми для роботи з рядками.

Стислі теоретичні відомості

Текстові дані в C# представляються за допомогою ключових слів **string** і **char**, які є скороченими позначеннями для типів **System.String** і **System.Char**. **string** представляє набір символів (наприклад, "Text"), а **char** – одиночний символ (наприклад, 'T').

Символьний тип **char** призначений для зберігання символів в кодуванні Unicode. У цьому класі визначені статичні методи, що дозволяють задати вид і категорію символу, а також перетворити символ в верхній або нижній регістр і в число. Основні методи типу **char** наведено в Таблиця 4.1.

Таблиця 4.1 Основні методи типу **char**

Метод	Пояснення
GetNumericValue	Повертає числове значення символу, якщо він є цифрою, або -1 в іншому випадку
GetUnicodeCategory	Повертає категорію символу
IsControl	Повертає true, якщо символ є керуючим
IsDigit	Повертає true, якщо символ є десятковою цифрою
IsLetter	Повертає true, якщо символ є буквою
IsLetterOrDigit	Повертає true, якщо символ є буквою або цифрою
IsLower	Повертає true, якщо символ заданий у нижньому регістрі
IsNumber	Повертає true, якщо символ є числом (десятковим або шістнадцятковим)
IsPunctuation	Повертає true, якщо символ є знаком пунктуації
IsSeparator	Повертає true, якщо символ є роздільником
IsUpper	Повертає true, якщо символ записаний у верхньому регістрі
IsWhiteSpace	Повертає true, якщо символ є пробільним (пробіл, символ нового рядка та символ повернення каретки)
Parse	Перетворити рядок в символ (рядок повинен складатися з одного символу)

Метод	Пояснення
<code>ToLower</code>	Перетворює символ в нижній регістр
<code>ToUpper</code>	Перетворює символ у верхній регістр

Тип `string`, призначений для роботи з рядками символів в кодуванні Unicode. Тип `string` містить методи для визначення довжини символічних даних, пошуку підрядка в поточному рядку і перетворення символів між верхнім і нижнім регістрами.

Для рядків визначені такі оператори операції:

- присвоювання (`=`);
- перевірка на рівність (`==`);
- перевірка на нерівність (`!=`);
- звернення за індексом (`[]`);
- зчеплення (конкатенація) рядків (`+`).

Рядки рівні, якщо мають однакову кількість символів і збігаються посимвольно.

Основні елементи типу `string` наведено в Таблиця 4.2.

Таблиця 4.2 Основні елементи типу `string`

Елемент	Пояснення
<code>Compare</code>	Порівняння двох рядків у лексикографічному (алфавітному) порядку
<code>CompareOrdinal</code>	Порівняння двох рядків за кодами символів
<code>CompareTo</code>	Порівняння поточного екземпляра рядка з іншим рядком
<code>Concat</code>	Конкатенація рядків. Метод допускає зчеплення довільного числа рядків
<code>Copy</code>	Створення копії рядка
<code>Empty</code>	Порожній рядок
<code>Format</code>	Форматування відповідно до заданих специфікаторами формату
<code>IndexOf</code> , <code>IndexOfAny</code> , <code>LastIndexOf</code> , <code>LastIndexOfAny</code>	Визначення індексів першого і останнього входження заданого підрядка або будь-якого символу із заданого набору
<code>Insert</code>	Вставка підрядка в задану позицію
<code>Join</code>	Злиття масиву рядків в єдиний рядок. Між елементами масиву вставляються роздільники

Елемент	П'яснення
Length	Довжина рядка (кількість символів)
PadLeft, PadRight	Вирівнювання рядків по лівому або правому краю шляхом вставки потрібного числа пробілів на початку або в кінці рядка
Remove	Видалення підрядка із заданої позиції
Replace	Заміна всіх входжень заданого підрядка або символу новими підрядком або символом
Split	Розділяє рядок на елементи, використовуючи задані роздільники. Результати поміщаються в масив рядків
StartsWith, EndsWith	Повертає true або false залежно від того, починається або закінчується рядок заданим підрядком
Substring	Виділення підрядка, починаючи з заданої позиції
ToCharArray	Перетворення рядка в масив символів
ToLower, ToUpper	Перетворення символів рядка до нижнього або верхнього регістру
Trim, TrimStart, TrimEnd	Видалення пробілів на початку і наприкінці рядка або тільки з одного його кінця

Робоче завдання

Навчитися виконувати прості операції над символами та рядками.

Хід роботи

Для кожного з двох індивідуальних завдань написати програму на мові C#, яка розв'язує поставлену задачу.

Індивідуальне завдання

Завдання 1.

1. Ввести три окремих символи. Перевірити чи всі символи різні.
2. Ввести три окремих символи. Перевірити чи хоча б один з них є буквою або цифрою.
3. Ввести три окремих символи. Вивести на екран всі символи у верхньому регістрі.
4. Ввести три окремих символи. Перевірити чи хоча б один з них є цифрою.
5. Ввести три окремих символи. Перевірити чи хоча б один з них є символом у верхньому регістрі.
6. Ввести три окремих символи. Вивести на екран числовий код кожного символу.

7. Ввести три окремих символи. Перевірити чи хоча б один з них є знаком пунктуації.
8. Ввести три окремих символи. Вивести на екран всі символи у нижньому регістрі.
9. Ввести три окремих символи. Перевірити чи всі вони є роздільниками.
10. Ввести три окремих символи. Перевірити чи хоча б один з них є пробілом.
11. Ввести три окремих символи. Перевірити чи хоча б один з них є роздільником.
12. Ввести три окремих символи. Перевірити чи всі вони є цифрами.
13. Ввести три окремих символи. Перевірити чи всі вони є символами у верхньому регістрі.
14. Ввести три окремих символи. Перевірити чи всі вони є знаками пунктуації.
15. Ввести три окремих символи. Перевірити чи хоча б один з них є буквою.
16. Ввести три окремих символи. Перевірити чи всі вони є пробілом.
17. Ввести три окремих символи. Визначити категорію кожного символу.
18. Ввести три окремих символи. Перевірити чи всі вони є буквами.
19. Ввести три окремих символи. Перевірити чи всі вони є буквами або цифрами.
20. Ввести три окремих символи. Перевірити чи всі символи однакові.

Завдання 2

1. Ввести три окремих рядки тексту. Перевірити чи хоча б один з рядків закінчується символом «.».
2. Ввести три окремих рядки тексту. Перевірити чи кожен рядок починається буквою.
3. Ввести три окремих рядки тексту. Створити новий рядок, який міститиме останні символи кожного з рядків записані через кому.
4. Ввести три окремих рядки тексту. Перевірити чи всі рядки однакові.
5. Ввести три окремих рядки тексту. Перевірити чи кожен рядок закінчується буквою.
6. Ввести три окремих рядки тексту. Вивести на екран всі рядки у верхньому регістрі.
7. Ввести три окремих рядки тексту. Перевірити чи кожен рядок починається пробілом.
8. Ввести три окремих рядки тексту. Перевірити чи довжина хоча б одного з рядків більша 5 символів .

9. Ввести три окремих рядки тексту. Створити новий рядок, який міститиме довжини рядків записані через пробіл.
10. Ввести три окремих рядки тексту. Перевірити чи кожен рядок закінчується пробілом.
11. Ввести три окремих рядки тексту. Перевірити чи хоча б один з рядків починається на букву «А».
12. Ввести три окремих рядки тексту. Перевірити чи кожен рядок починається знаком пунктуації.
13. Ввести три окремих рядки тексту. Перевірити чи всі рядки різні.
14. Ввести три окремих рядки тексту. Вивести на екран всі рядки у нижньому регістрі.
15. Ввести три окремих рядки тексту. Створити нові рядки, які є записом початкових рядків великими та малими літерами.
16. Ввести три окремих рядки тексту. Перевірити чи кожен рядок закінчується знаком пунктуації.
17. Ввести три окремих рядки тексту. Створити нові рядки, які є попарними об'єднанням різних рядків і вивести їх на екран.
18. Ввести три окремих рядки тексту. Перевірити чи всі рядки одної довжини.
19. Ввести три окремих рядки тексту. Перевірити чи хоча б один з рядків є порожнім.
20. Ввести три окремих рядки тексту. Створити новий рядок, який міститиме перші символи кожного з рядків записані через кому.

Контрольні питання

1. Що таке символний тип даних?
2. Що таке рядковий тип даних?
3. Як перевірити чи символ належить до певної категорії?
4. Як перетворити символ у верхній чи нижній регістр?
5. Чи можна зберігати у змінній типу **char** символи української абетки?
6. Що таке операція конкатенації?
7. Як виконується порівняння рядків?
8. Як звернутися до окремого символу з рядку?
9. Як знайти довжину рядка?

ЛАБОРАТОРНА РОБОТА №5. РОБОТА З ТЕКСТОВИМИ РЯДКАМИ. ЧАСТИНА 2. РОЗШИРЕНІ МОЖЛИВОСТІ РОБОТИ З РЯДКАМИ

Мета

Вивчити розширені можливості роботи з рядками.

Стислі теоретичні відомості

Рядкові літерали C# можуть містити керуючі послідовності, які дозволяють уточнювати те, як символні дані виводяться на екран. Кожна керуюча послідовність починається з символу зворотної косої риски, за яким слідує знак, що інтерпретується. У Таблиця 5.1 перераховані основні керуючі послідовності.

Таблиця 5.1 Керуючі послідовності в рядкових літералах

Керуюча послідовність	Опис
\'	Вставляє в строковий літерал символ одинарної лапки
\"	Вставляє в строковий літерал символ подвійної лапки
\\	Вставляє в строковий літерал символ зворотної косої риски. Особливо корисна при визначенні шляхів до файлів і мережевих ресурсів
\a	Змушує систему видавати звуковий сигнал, який в консольних додатках може служити аудіо-підказкою користувачеві
\n	Вставляє символ нового рядка
\r	Вставляє символ повернення каретки
\t	Вставляє в строковий літерал символ горизонтальної табуляції

Метод `Format` замінює всі входження заповнювачів у фігурних дужках значеннями відповідних змінних зі списку виводу. Після номера заповнювача можна задати мінімальну ширину поля виводу, а також вказати специфікатор формату, який визначає форму подання виведеного значення.

У загальному вигляді заповнювач задається таким чином:

{n [, m [: специфікатор_формата [число]]}

де `n` – номер заповнювача. Заповнювачі нумеруються з нуля, нульовий заповнювач замінюється значенням першої змінної зі списку виведення,

перший заповнювач – значенням другої змінної і т. д. Параметр *m* визначає мінімальну ширину поля, яке відводиться під виведене значення. Якщо виведеному числу досить меншої кількості позицій, невикористовувані позиції заповнюються пробілами. Якщо числу потрібно більше позицій, параметр ігнорується.

Специфікатор формату визначає формат виведення значення. Наприклад, специфікатор *C* (Currency) означає, що параметр повинен форматуватися як валюта з урахуванням національних особливостей подання, а специфікатор *X* (Hexadecimal) задає шестнадцатеричну форму подання виведеного значення. Після деяких специфікаторів можна задати кількість позицій, що відводяться під дробову частину виведеного значення.

Можливо використовувати власні шаблони форматування. Після двокрапки задається вид виведеного значення посимвольно, причому на місці кожного символу може стояти або *#*, або *0*. Якщо вказаний знак *#*, на цьому місці буде виведена цифра числа, якщо вона не дорівнює нулю. Якщо вказаний *0*, буде виведена будь-яка цифра, в тому числі і *0*.

Робоче завдання

Навчитися виконувати розширені операції над рядками, такі як, пошук, копіювання та видалення підрядка, форматування рядка, використання керуючих послідовностей символів.

Хід роботи

Написати програму на мові *C#*, яка розв'язує поставлену задачу з обов'язковим використанням текстових рядків.

Індивідуальне завдання

1. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Об'єднати два рядки в один і вивести на екран кожне слово з нового рядку

- Замінити у першому рядку всі слова, які співпадають з першим словом другого рядку на слово «hello»
 - Додати на початок першого рядку всі символи другого рядку між першим і останнім словом
 - Вивести отримані рядки на екран
2. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:
- Поміняти місцями останні слова у двох рядках
 - Видалити з першої половини першого рядку всі слова які співпадають з останнім словом другого рядку
 - Доповнити другий рядок до 40 символів пробілами в кінці рядку
 - Вивести отримані рядки на екран
3. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:
- Вставити в перший рядок після першого слова довжину другого рядку
 - Додати на початок першого рядку всі символи другого рядку між першим і останнім словом
 - Доповнити перший рядок до 40 символів пробілами на початку рядку
 - Вивести отримані рядки на екран
4. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:
- Замінити у першому рядку всі слова, які співпадають з першим словом другого рядку на слово «hello»
 - Доповнити перший рядок до 40 символів пробілами на початку рядку
 - Поміняти місцями останні слова у двох рядках
 - Вивести отримані рядки на екран
5. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:
- Видалити з другої половини першого рядку всі слова які співпадають з першим словом другого рядку
 - Вставити в перший рядок після першого слова довжину другого рядку
 - Замінити у першому рядку всі пробіли на знак табуляції
 - Вивести отримані рядки на екран
6. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:

- Вставити в перший рядок перед останнім словом довжину другого рядку
 - Видалити з першої половини першого рядку всі слова які співпадають з останнім словом другого рядку
 - Доповнити другий рядок до 40 символів пробілами в кінці рядку
 - Вивести отримані рядки на екран
7. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:
- Доповнити другий рядок до 40 символів пробілами в кінці рядку
 - Замінити у першому рядку всі слова, які співпадають з першим словом другого рядку на слово «hello»
 - Вставити в перший рядок перед останнім словом довжину другого рядку
 - Вивести отримані рядки на екран
8. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:
- Визначити чи співпадають останні слова у двох рядків
 - Замінити у першому рядку всі пробіли на знак табуляції
 - Поміняти місцями перші слова у рядках
 - Вивести отримані рядки на екран
9. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:
- Видалити з першої половини першого рядку всі слова які співпадають з останнім словом другого рядку
 - Видалити з другого рядку перші 5 символів і вставити їх на початок першого рядку
 - Визначити чи співпадають останні слова у двох рядків
 - Вивести отримані рядки на екран
10. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:
- Поміняти місцями перші слова у рядках
 - Об'єднати два рядки в один і вивести на екран кожне слово з нового рядку
 - Замінити у першому рядку всі слова, які співпадають з першим словом другого рядку на слово «hello»
 - Вивести отримані рядки на екран
11. Ввести два окремих рядки, які містять щонайменше по 4 слова.
Виконати такі дії:

- Додати на початок першого рядку всі символи другого рядку між першим і останнім словом
- Вставити в перший рядок перед останнім словом довжину другого рядку
- Поміняти місцями перші слова у рядках
- Вивести отримані рядки на екран

12. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Видалити з другого рядку перші 5 символів і вставити їх на початок першого рядку
- Визначити чи співпадають останні слова у двох рядків
- Видалити з першої половини першого рядку всі слова які співпадають з останнім словом другого рядку
- Вивести отримані рядки на екран

13. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Замінити у першому рядку всі пробіли на знак табуляції
- Видалити з другої половини першого рядку всі слова які співпадають з першим словом другого рядку
- Об'єднати два рядки в один і вивести на екран кожне слово з нового рядку
- Вивести отримані рядки на екран

14. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Доповнити перший рядок до 40 символів пробілами на початку рядку
- Поміняти місцями останні слова у двох рядках
- Визначити чи співпадають останні слова у двох рядків
- Вивести отримані рядки на екран

15. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Додати на початок першого рядку всі символи другого рядку між першим і останнім словом
- Вставити в перший рядок перед останнім словом довжину другого рядку
- Об'єднати два рядки в один і вивести на екран кожне слово з нового рядку
- Вивести отримані рядки на екран

16. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Доповнити другий рядок до 40 символів пробілами в кінці рядку
- Замінити у першому рядку всі пробіли на знак табуляції
- Визначити чи співпадають останні слова у двох рядків
- Вивести отримані рядки на екран

17. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Доповнити перший рядок до 40 символів пробілами на початку рядку
- Видалити з другого рядку перші 5 символів і вставити їх на початок першого рядку
- Поміняти місцями останні слова у двох рядках
- Вивести отримані рядки на екран

18. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Поміняти місцями останні слова у двох рядках
- Замінити у першому рядку всі слова, які співпадають з першим словом другого рядку на слово «hello»
- Видалити з першої половини першого рядку всі слова які співпадають з останнім словом другого рядку
- Вивести отримані рядки на екран

19. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Вставити в перший рядок після першого слова довжину другого рядку
- Поміняти місцями перші слова у рядках
- Додати на початок першого рядку всі символи другого рядку між першим і останнім словом
- Вивести отримані рядки на екран

20. Ввести два окремих рядки, які містять щонайменше по 4 слова.

Виконати такі дії:

- Видалити з другої половини першого рядку всі слова які співпадають з першим словом другого рядку
- Доповнити перший рядок до 40 символів пробілами на початку рядку
- Об'єднати два рядки в один і вивести на екран кожне слово з нового рядку
- Вивести отримані рядки на екран

Контрольні питання

1. Як видалити підрядок з рядка?
2. Як знайти підрядок в рядку?
3. Як вставити підрядок у рядок?
4. Як виконати вирівнювання за довжиною рядків по лівому або правому краю?
5. Як замінити всі входження заданого підрядка у рядок?
6. Як виконати перетворення символів рядка до нижнього або верхнього регістру?
7. Як видалити пробіли на початку і наприкінці рядка?
8. Як виконати форматування рядка?
9. Які керуючі послідовності символів можна використовувати в рядкових літералах?
10. Який загальний вигляд заповнювача у рядкових літералах?

ЛАБОРАТОРНА РОБОТА №6. ОПЕРАТОР РОЗГАЛУЖЕННЯ IF/ELSE

Мета

Засвоїти роботу оператора розгалуження **if/else**.

Стислі теоретичні відомості

Рядкові літерали C# можуть містити керуючі послідовності, які Умовний оператор **if/else** використовується для розгалуження процесу обчислень на два напрямки.

Формат оператора:

```
if (логічний_вираз) оператор_1; [else оператор_2; ]
```

Спочатку обчислюється логічний вираз. Якщо він має значення true, виконується перший оператор, інакше – другий. Після цього управління передається на оператор, наступний за умовним. Гілка else може бути відсутня.

Якщо в гілці потрібно виконати декілька операторів, їх необхідно об'єднати в блок за допомогою фігурних дужок { }, інакше компілятор не зможе зрозуміти, де закінчується розгалуження. Блок може містити будь-які оператори, у тому числі опису та інші умовні оператори (але не може складатися з одних описів). Необхідно враховувати, що змінна, описана в блоці, поза блоком не існує.

Робоче завдання

Вивчити роботу оператора **if/else**. Засвоїти роботу логічних операцій. Навчитись розраховувати логічні вирази.

Хід роботи

Написати програму на мові C#, яка розраховує значення функції з використанням оператора **if/else** та логічних операцій **&&** та **||** (за потреби).

Перевіряти вираз на ОДЗ не обов'язково. У разі, якщо вираз має декілька варіантів рішення, то вибирається той вираз, що йде першим в умові.

Індивідуальне завдання

$$1. z = \begin{cases} ax + b & , x < 0, b < 0 \\ ax^2 & , x > -1, x < 1, a > 0 \\ ax - b & , x > 0, b < 0 \end{cases}$$

$$2. z = \begin{cases} \cos x + a & , -\pi < x < \pi \\ \sin x + b & , x > 2\pi, b \geq 0 \\ \operatorname{ctg} x + ab & , \text{в інших випадках} \end{cases}$$

$$3. z = \begin{cases} ax & , a > 0, x > 0 \\ bx & , b > 0, x < 0 \\ \ln x & , x > 0 \\ ab + \ln x & , x > 0, ab < 0 \end{cases}$$

$$4. z = \begin{cases} 5 + \sqrt[3]{|x|} + a & , x > 0, a < -5 \\ 10.956 - x^5 & , x < 0, a > 8 \end{cases}$$

$$5. z = \begin{cases} 95 - e^x & , x > 0, x < 5 \\ \frac{1-x}{1+x} & , x > 5 \\ x^2 - x + 1 & , x < 0, x \geq -20 \end{cases}$$

$$6. z = \begin{cases} x + y & , x > 5, y < 10 \\ \frac{x-y}{x+y} & , x \neq y, x < 0 \\ x - y & , x > 20, y > 0 \end{cases}$$

$$7. z = \begin{cases} x + a + b & , a + x > b, x > 10 \\ x - y + c & , a + x < b, y - 1 < 10 \end{cases}$$

$$8. z = \begin{cases} a + \frac{x-y}{a+y} & , a + y \neq 0, x > 20 \\ 50 - y + a & , y - a \geq 0 \\ \frac{c}{\sin x^2} & , -\frac{\pi}{2} \leq x \leq \frac{\pi}{2} \end{cases}$$

$$9. z = \begin{cases} a & , x > 0, a > 0 \\ b & , x < 0, b > 0 \\ \frac{a+b}{\cos x} & , a < 0 \\ \frac{\pi}{4} & , \text{в інших випадках} \end{cases}$$

$$10. z = \begin{cases} e^{x-\cos x} & , 0 < x < 6 \\ x^2 + y^2 & , |x + y| \leq 1 \\ x^2 - y^2 + \cos^2 y & , -\frac{\pi}{2} \leq y \leq \frac{\pi}{2} \end{cases}$$

$$11. z = \begin{cases} 4 - 5.76x \cos y & , |x + y| \leq 1 \\ e - \frac{1}{\pi}(x + y) & , |x - y| > 1 \\ x^e & , x > 0 \end{cases}$$

$$12. z = \begin{cases} \arcsin x^2 & , -1 \leq x \leq 1 \\ e^{x-1} & , 4 \leq |x| < 6 \\ \sqrt[x]{|y - a|} & , x > 0, y - a \neq 0 \end{cases}$$

$$13. z = \begin{cases} \operatorname{arctg} \frac{x-1}{x+1} & , -1 < x \leq 3 \\ e^{\sin x+1} - x^8 & , y < 0, 4 \leq x < 10 \end{cases}$$

$$14. z = \begin{cases} \log_x |x - 4| & , x > 0, a < 0, b < 0 \\ ax + b & , x < 0, a \in \{1; 4; 6\} \\ \operatorname{tg} x + \operatorname{ctg}^e x & , -\pi \leq x \leq -\frac{\pi}{4}, a > 0 \\ 5 & , \text{в інших випадках} \end{cases}$$

$$15. z = \begin{cases} \lg \left| \frac{x-y}{6} \right| & , x < y \\ \sqrt[3]{x} & , x \neq 0, y < 0 \\ 10 - xy^e & , |y| \leq 5 \end{cases}$$

$$16. z = \begin{cases} \frac{1}{x^2-1} & , -3 < x \leq 6 \\ y^2 - 2xya + x^2 & , y \neq x, y > -4 \\ \sqrt[5]{|x - y + 1|} & , y \leq -1, x \geq -0.5 \end{cases}$$

$$17. z = \begin{cases} \operatorname{arctg} e^x & , 0 \leq x \leq 6 \\ |x - y| + x & , x \geq 10, y < 0 \\ -5x + 6y - c & , x + y > 0 \end{cases}$$

$$18. z = \begin{cases} \operatorname{sh} x + y & , x < 0, y > 10 \\ \operatorname{ch} x - y & , x > 0, y \leq 1 \end{cases}$$

$$19. z = \begin{cases} \cos x & , x \geq \frac{3\pi}{2} \\ \cos^2 x & , 0 \leq x \leq \pi \\ \sin x - \frac{y}{c} & , \text{в інших випадках} \end{cases}$$

$$20.z = \begin{cases} \ln a + \cos x & , a > 0, x \leq 0 \\ y - 5 \ln x & , x < 4 \\ e^{|a-b|} & , -1 \leq a < 4, -2 \leq b < 2 \end{cases}$$

$$21.z = \begin{cases} \log_4^2 |x + y| & , x < 0, y \geq 4 \\ x^4 + 3x^3 + 5x + 2 & , 0 \leq x \leq 3 \\ x + \operatorname{tg} e^{xy} & , x > 0, y > 0 \end{cases}$$

$$22.z = \begin{cases} \sqrt{x - y + 2xy} & , -3 \leq y < 3 \\ \frac{\cos x}{\cos x + 1} & , y > 3 \\ \sin x - \cos y & , y < 3 \end{cases}$$

$$23.z = \begin{cases} x - y + 2^x & , -2 \leq x < 5, y < 3 \\ 5 - 6y & , y \geq 3 \\ 33.3 - 2(x - y)^2 & , \text{в інших випадках} \end{cases}$$

$$24.z = \begin{cases} x^2 - xy - 2xy^2 & , x > 4, y < 5 \\ y + 2x - ye^x & , x < 4, y < 5 \\ |\ln x| - \ln |y| & , \text{в інших випадках} \end{cases}$$

$$25.z = \begin{cases} f + xy - \log_3 f & , f > 3 \\ \cos f - \ln x & , -3 \leq f < 3 \\ 45x - e & , f < -3 \end{cases}$$

Контрольні питання

1. Які оператори розгалуження ви знаєте?
2. Дайте опис оператора **if/else**.
3. Намалюйте узагальнену блок-схему оператора **if/else**.
4. Чи можна не використовувати конструкцію «в інших випадках» в операторі **if/else**?
5. Що таке операторні дужки?
6. Написати програму. Умову і вимоги до задачі надає викладач.

ЛАБОРАТОРНА РОБОТА №7. ОПЕРАТОР МНОЖИННОГО ВИБОРУ SWITCH/CASE

Мета

Засвоїти роботу оператора множинного вибору **switch/case**.

Стислі теоретичні відомості

Оператор **switch/case** призначений для розгалуження процесу обчислень на кілька напрямів.

Формат оператора:

```
switch (вираз) {  
  case константний_вираз_1: [список_операторів_1]  
  case константний_вираз_2: [список_операторів_2]  
  case константний_вираз_n: [список_операторів_3]  
  [default: оператори]
```

Виконання оператора починається з обчислення виразу. Тип виразу найчастіше цілочисельний або рядковий. У загальному випадку вираз може бути будь-якого типу, для якого існує неявне перетворення до зазначених, а також типу переліку. Потім управління передається першому оператору зі списку, позначеного константним виразом, значення якого співпало з обчисленим.

Всі константні вирази повинні мати можливість неявного приведення до типу виразу в дужках. Якщо збігів не сталося, виконуються оператори, розташовані після слова **default** (а при його відсутності управління передається наступному за **switch/case** оператору).

Кожна гілка повинна закінчуватися явним оператором переходу, а саме оператором **break**, **goto** або **return**:

- оператор **break** виконує вихід з самого внутрішнього з операторів **switch/case**;

- оператор **goto** виконує перехід на вказану після нього мітку, зазвичай це мітка **case** однієї з нижчих гілок оператора **switch/case**;
- оператор **return** виконує вихід з функції, в тілі якої він записаний.

Робоче завдання

Вивчити роботу оператора **switch/case**.

Хід роботи

Написати програму на мові C#, яка вирішує поставлене завдання з використанням оператора **switch/case**.

Індивідуальне завдання

1. Елементи рівностороннього трикутника пронумеровані таким чином: 1 – сторона a , 2 – радіус R_1 вписаного кола ($R_1 = A_3/6$), 3 – радіус R_2 описаного кола ($R_2 = 2R_1$), 4 – площа $S = a^2 \sqrt{3}/4$. Дано номер одного з цих елементів і його значення. Вивести значення інших елементів даного трикутника.
2. Дано ціле число в діапазоні 1-7. Вивести повідомлення – назву дня тижня, що відповідає даному числу.
3. Дано ціле число K . Вивести рядок-опис оцінки, що відповідає числу K (1 – «погано», 2 – «незадовільно», 3 – «задовільно», 4 – «добре», 5 – «відмінно»). Якщо K не лежить в діапазоні 1-5, то вивести текст «помилка».
4. Робот може переміщатися в чотирьох напрямках («П» – північ, «З» – захід, «Д» – південь, «С» – схід) і приймати три цифрові команди: 0 – продовжувати рух, 1 – поворот наліво, -1 – поворот направо. Дано символ S – вихідний напрям руху робота і ціле число N – надіслана цифрова команда. Вивести напрям руху робота після виконання отриманої команди.

5. У східному календарі прийнятий 60-річний цикл, що складається з 12-ти річних підциклів, що позначаються назвами кольорів: зелений, червоний, жовтий, білий і чорний. У кожному підциклі роки носять назви тварин: пацюка, корови, тигра, зайця, дракона, змії, коня, вівці, мавпи, курки, собаки і свині. За номером року визначити його назву, якщо 1984 року – початок циклу: «рік зеленого пацюка».
6. Дано номер місяця – ціле число в діапазоні 1-12 (1 – січень, 2 – лютий і т. д.). Вивести назву відповідної пори року («зима», «весна», «літо», «осінь»).
7. Локатор орієнтований на одну зі сторін світу («П» – північ, «З» – захід, «Д» – південь, «С» – схід) і може приймати три цифрові команди повороту: 1 – поворот наліво, -1 – поворот направо, 2 – поворот на 180° . Дано символ S – вихідна орієнтація локатора й цілі числа N_1 і N_2 – дві послані команди. Вивести орієнтацію локатора після виконання цих команд.
8. Дано номер місяця – ціле число в діапазоні 1-12 (1 – січень, 2 – лютий і т.д.). Визначити кількість днів у цьому місяці для невисокосного року.
9. Арифметичні дії над числами пронумеровані таким чином: 1 – додавання, 2 – віднімання, 3 – множення, 4 – ділення. Дано номер дії N (ціле число в діапазоні 1-4) і дійсні числа A і B (B не дорівнює 0). Виконати над числами вказану дію і вивести результат.
10. Одиниці довжини пронумеровані таким чином: 1 – дециметр, 2 – кілометр, 3 – метр, 4 – міліметр, 5 – сантиметр. Дано номер одиниці довжини (ціле число в діапазоні 1-5) і довжина відрізка в цих одиницях (дійсне число). Знайти довжину відрізка в метрах.
11. Одиниці маси пронумеровані таким чином: 1 – кілограм, 2 – міліграм, 3 – грам, 4 – тонна, 5 – центнер. Дано номер одиниці маси (ціле число в діапазоні 1-5) і маса тіла в цих одиницях (дійсне число). Знайти масу тіла в кілограмах.

12. Дано два цілих числа: D (день) і M (місяць), що визначають правильну дату невисокосного року. Вивести значення D і M для дати, що передуює вказаній.
13. Дано два цілих числа: D (день) і M (місяць), що визначають правильну дату невисокосного року. Вивести значення D і M для дати, що відповідає наступній за вказаною.
14. Елементи кола пронумеровані таким чином: 1 – радіус кола R , 2 – діаметр $D = 2R$, 3 – довжина $L = 2\pi R$, 4 – площа кола $S = \pi R^2$. Дано номер одного з цих елементів і його значення. Вивести значення решти елементів даного кола.
15. Елементи рівнобедреного прямокутного трикутника пронумеровані наступним чином: 1 – катет a , 2 – гіпотенуза $c = a\sqrt{2}$, 3 – висота h , опущена на гіпотенузу ($h=c/2$), 4 – площа $S=ch/2$. Дано номер одного з цих елементів і його значення. Вивести значення інших елементів даного трикутника.
16. Мастям гральних карт присвоєно порядкові номери: 1 – піки, 2 – трефи, 3 – бубни, 4 – черви. Номіналу карт, старших десятки, привласнені номери: 11 – валет, 12 – дама, 13 – король, 14 – туз. Дано два цілих числа: N – номіналом ($6 \leq N \leq 14$) і M – масть карти ($1 \leq M \leq 4$). Вивести назву відповідної карти виду «шістка бубен», «дама пік», «туз трэф» і т. д.
17. Дано ціле число в діапазоні 20-69, яке визначає вік (в роках). Вивести рядок-опис зазначеного віку, забезпечивши правильне погодження числа зі словом «рік», наприклад: 20 – «двадцять років», 32 – «тридцять два роки», 41 – «сорок один рік».
18. Дано ціле число в діапазоні 10-40, яке визначає кількість навчальних завдань з деякої теми. Вивести рядок-опис зазначеної кількості завдань, забезпечивши правильне узгодження числа зі словами «Навчальне завдання», наприклад: 18 – «вісімнадцять навчальних завдань», 23 – «двадцять три навчальних завдання».

19. Дано ціле число в діапазоні 100-999. Вивести рядок-опис даного числа, наприклад: 256 – «двісті п'ятьдесят шість», 814 – «вісімсот чотирнадцять».
20. Дано два цілих числа: D (день) і M (місяць), що визначають правильну дату. Вивести знак Зодіаку, який відповідає цій даті: «Водолій» (20.01 - 18.02), «Риби» (19.02-20.03), «Овен» (21.03-19.04), «Телець» (20.04-20.05), «Близнюки» (21.05-21.06), «Рак» (22.06-22.07), «Лев» (23.07-22.08), «Діва» (23.08-22.09), «Терези» (23.09-22.10), «Скорпіон» (23.10-22.11), «Стрілець» (23.11-21.12), «Козеріг» (22.12-19.01).

Контрольні питання

1. Які оператори розгалуження ви знаєте?
2. Дайте опис оператора **switch/case**.
3. Яка різниця між оператором **if/else** та **switch/case**?
4. Які типи даних може приймати вираз-селектор в операторі **switch/case**?
5. Чи має оператор **switch/case** конструкцію «в інших випадках»?
6. Чи можна не використовувати конструкцію «в інших випадках» в операторі **switch/case**?
7. Що таке блок операторів?
8. Написати програму. Умову і вимоги до задачі надає викладач.

ЛАБОРАТОРНА РОБОТА №8. ОПЕРАТОРИ ЦИКЛУ. ЧАСТИНА 1. ЦИКЛ FOR

Мета

Засвоїти поняття циклу. Отримати досвід використання циклічної конструкції **for** для вирішення задач.

Стислі теоретичні відомості

Цикл – різновид керуючої конструкції у високорівневих мовах програмування, призначена для організації багаторазового виконання набору інструкцій. Цикл багаторазово повторює набір інструкцій, доки не буде виконана умова виходу з циклу. Одне повторення набору інструкцій (тіло циклу) також ще називають ітерацією.

Параметром циклу називається змінна, яка використовується при перевірці умови продовження циклу і примусово змінюється на кожній ітерації, причому, як правило, на одну і ту ж величину. Якщо параметр циклу цілочисельний, він називається лічильником циклу. Кількість повторень такого циклу можна визначити заздалегідь.

Цикл завершується, якщо умова його продовження не виконується. Можливо примусове завершення як поточної ітерації, так і циклу в цілому. Для цього служать оператори **break**, **continue**, **return** і **goto**.

Цикл з параметром має наступний формат:

for (ініціалізація; вираз; модифікації) оператор;
--

Ініціалізація служить для оголошення величин, що використовуються в циклі, і присвоєння їм початкових значень. У цій частині можна записати декілька операторів, розділених комою. Областю дії змінних, оголошених в частині ініціалізації циклу, є весь цикл. Ініціалізація виконується один раз на початку виконання циклу.

Вираз типу **bool** визначає умову виконання циклу: якщо його результат дорівнює **true**, цикл виконується.

Модифікації виконуються після кожної ітерації циклу і служать зазвичай для зміни параметрів циклу. У частині модифікацій можна записати декілька операторів через кому.

Робоче завдання

Навчитись використовувати цикли з параметром **for** мови C# для вирішення задач. Дослідити роботу таких циклів. Вивчити роботу операторів **break** та **continue**.

Хід роботи

Написати програму на мові C#, яка розраховує значення виразу з використанням оператора **for**. Забороняється використовувати функції класу **Math**. Для варіантів 1-10 порівняти розраховані значення виразу зі значеннями отриманими за допомогою перевірконої функції.

Індивідуальне завдання

№	Завдання	Обмеження	Перевірочна функція
Ряди			
1	$S = \sum_{i=0}^n \frac{(-1)^i x^{2i+1}}{(2i+1)!}$	$x \in R, n \in N$	$\sin x$
2	$S = 1 + \sum_{i=1}^n \frac{x^i \prod_{j=n-i+1}^n j}{i!}$	$x \in R, n \in N$ $-1 < x < 1$	$(1+x)^n$
3	$S = 1 + \sum_{i=1}^n \frac{(-1)^i x^{2i}}{(2i)!}$	$x \in R, n \in N$	$\cos x$
4	$S = 1 + \sum_{i=1}^n \frac{x^i}{i!}$	$x \in R, n \in N$	e^x
5	$S = \sum_{i=1}^n \frac{-x^i}{i}$	$x \in R, n \in N$ $-1 \leq x < 1$	$\ln(1-x)$
6	$S = \sum_{i=0}^n \frac{(-1)^i x^{2i+1}}{2i+1}$	$x \in R, n \in N$ $-1 < x < 1$	$\arctg x$
7	$S = \sum_{i=0}^n \frac{(2i)! x^{2i+1}}{4^i (i!)^2 (2i+1)}$	$x \in R, n \in N$ $-1 < x < 1$	$\arcsin x$
8	$S = \sum_{i=1}^n \frac{(-1)^{i+1} x^i}{i}$	$x \in R, n \in N$ $-1 < x \leq 1$	$\ln(1+x)$

№	Завдання	Обмеження	Перевірочна функція
9	$S = \frac{1}{a} + \sum_{i=1}^n \frac{(-1)^i x^i}{a^{i+1}}$	$x, a \in R, n \in N$ $-1 \leq x \leq 1$	$\frac{1}{a+1}$
Вирази			
10	$S = \sum_{i=1}^n (-1)^i \frac{i!}{2^i}$	$n \in N$	
11	$S = \sum_{i=1}^n \prod_{j=i}^{2i} j$	$n \in N$	
12	$S = \prod_{i=1}^n \frac{2i-1}{i}$	$n \in N$	
13	$S = \prod_{i=1}^n \sum_{j=1}^i j!$	$n \in N$	
14	$S = \sum_{i=1}^n \frac{(-1)^{i+1} x^{2i+1} 2^i}{(2i+1)!}$	$x \in R, n \in N$	
15	$S = \sum_{i=1}^n (-1)^i \frac{\prod_{j=i}^{i^2} x}{(2i)!}$	$x \in R, n \in N$	
16	$S = \sum_{i=1}^n \frac{(2i-1)!}{(2i+1)!^i}$	$n \in N$	
17	$S = \prod_{i=1}^n \frac{x^i}{i!}$	$x \in R, n \in N$	
18	$S = \sum_{i=1}^n \frac{(2i+1)^{2i+1}}{(-1)^{i-1}} (x+1)^i$	$x \in R, n \in N$	
19	$S = \sum_{i=1}^n \left((-1)^i \prod_{j=2i}^{3i} x^j j! \right)$	$x \in R, n \in N$	
20	$S = \sum_{i=1}^{2n} \frac{i^i}{(2i+1)!}$	$n \in N$	
21	$S = \sum_{i=1}^n \left(i + \sum_{j=1}^i \frac{2^{2i-1}}{(4i+1)!} \right)$	$n \in N$	
22	$S = \sum_{i=1}^n (-1)^{i-1} x^{i!} (2i-1)!$	$x \in R, n \in N$	
23	$S = \sum_{i=0}^n (-1)^i \frac{x^{i+1}}{i+1} (i+1)!$	$x \in R, n \in N$	
24	$S = \sum_{i=1}^n \frac{x^{i+1}}{(2i+1)!} - \sum_{j=1}^k \frac{2i-1}{x^{(-1)^i i}}$	$x \in R, n \in N, k \in N$	

№	Завдання	Обмеження	Перевірочна функція
25	$S = \sum_{i=1}^n (7i + 1)x^{\frac{i!}{(i+1)^2}}$	$x \in R, n \in N$	

Контрольні питання

1. Що таке цикл?
2. Дайте опис циклу **for**.
3. Намалюйте узагальнену блок-схему циклу **for**.
4. Що таке цикл з параметром?
5. Як потрібно записати цикл **for**, щоб змінна циклу збільшувалась на 1.
6. Як потрібно записати цикл **for**, щоб змінна циклу зменшувалася на 1.
7. Що таке ітерація циклу?
8. Назвіть оператор дострокового виходу з циклу. Опишіть його роботу.
9. Назвіть оператор дострокового завершення ітерації циклу. Опишіть його роботу.
10. Написати програму. Умову і вимоги до задачі надає викладач.

ЛАБОРАТОРНА РОБОТА №9. ОПЕРАТОРИ ЦИКЛУ. ЧАСТИНА 2. ЦИКЛИ WHILE I DO/WHILE

Мета

Отримати досвід використання циклічних конструкцій **while** і **do/while** для вирішення задач.

Стислі теоретичні відомості

Оператор **while** має такий формат:

```
while (вираз) оператор;
```

Вираз повинен бути логічного типу. Наприклад, це може бути операція відношення або просто логічна змінна. Якщо результат обчислення виразу дорівнює **true**, виконується простий або складений оператор (блок операторів). Ці дії повторюються до того моменту, поки результатом виразу не стане значення **false**. Після закінчення циклу управління передається на наступний за ним оператор.

Вираз обчислюється перед кожною ітерацією циклу. Якщо при першій перевірці вираз дорівнює **false**, цикл **while** не виконається жодного разу.

Поширеним прийомом програмування є організація нескінченного циклу із заголовком

```
while (true)
```

і примусовим виходом з тіла циклу за допомогою оператора передачі управління **break**.

Цикл з післяумовою **do/while** має такий формат:

```
do  
оператор  
while вираз;
```

Спочатку виконується простий або складений оператор, який утворює тіло циклу, а потім обчислюється вираз (він повинен мати тип **bool**). Якщо вираз істинний, тіло циклу виконується ще раз і перевірка повторюється. Цикл

завершується, коли вираз стане рівним **false** або в тілі циклу буде виконаний який-небудь оператор передачі управління.

Цей вид циклу застосовується в тих випадках, коли тіло циклу необхідно обов'язково виконати хоча б один раз, наприклад, якщо в циклі вводяться дані і виконується їх перевірка. Якщо ж такої необхідності немає, переважно користуватися циклом з передумовою.

Робоче завдання

Навчитись використовувати цикли з передумовою **while** та цикли з післяумовою **do/while** мови C# для вирішення задач. Дослідити роботу таких циклів. Вивчити роботу операторів **break** та **continue**.

Хід роботи

Написати програми на мові C#, які вирішують поставлені завдання:

- для першого завдання – з використанням циклу **while**;
- для другого завдання – з використанням циклу **do/while**.

Індивідуальне завдання

Завдання 1.

1. Дано ціле число N . Вивести найменше з цілих чисел K , для яких сума $1 + 2 + \dots + K$ буде більше або дорівнює N , і саму цю суму.
2. Скласти програму для пошуку суми цифр цілого числа, що вводиться з клавіатури.
3. Знайти усі двозначні числа, такі що, якщо до суми цифр цього числа додати квадрат цієї суми, то отримаємо це ж число.
4. Дано число. Визначити, чи є це число степенем двійки.
5. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення, знайти число, отримане при прочитанні числа N справа на ліво .

6. Вивести на друк усі тризначні числа, які діляться націло на власні цифри, а також перевертиш цього числа ділиться націло на свої цифри.
7. Дано ціле число N . Знайти найменше ціле додатне число K , при якому виконується нерівність $3^K > N$.
8. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення, вивести всі його цифри, починаючи з самої правої (розряду одиниць).
9. Дано ціле число N . Перевірити чи є число N числом Фібоначчі.
10. Скласти програму, яка знаходить найбільший спільний дільник двох цілих чисел, що вводяться з клавіатури.
11. Скласти програму-гру «Більше-Менше». Комп'ютер генерує випадкове ціле число, а користувач намагається вгадати згенероване число. Комп'ютер надає користувачу підказку, чи є введене число більшим або меншим від загаданого. Також з клавіатури вводиться максимальна кількість спроб.
12. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення визначити, чи присутня у записі числа N цифра «2».
13. Початковий вклад у банку дорівнює 5000 грн. Через місяць розмір вкладу збільшується на P процентів від наявний суми. За даним P визначити, через скільки місяців розмір вкладу перевищить 7000 грн і вивести знайдену кількість місяців K (ціле число) і остаточний розмір вкладу S (дійсне число).
14. Написати програму, яка обчислює суму і середнє арифметичне послідовності чисел (довжина послідовності не обмежена), які вводяться з клавіатури.
15. Знайдіть перше двозначне число, рівне квадрату числа його одиниць, складеному з кубом числа його десятків.

16. Спортсмен-лижник почав тренування, пробігши в перший день 5 км. Кожен наступний день він збільшував довжину пробігу на P відсотків від пробігу попереднього дня. За даним P визначити, після якого дня сумарний пробіг лижника за всі дні перевищить 100 км, і вивести знайдену кількість днів K (ціле) і сумарний пробіг S (дійсне число).
17. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення визначити, чи є в запису числа N непарні цифри.
18. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення, знайти кількість і суму його цифр.
19. Знайти число Фібоначчі, порядковий номер якого вводиться з клавіатури.
20. Написати програму, яка визначає максимальне число з введеної з клавіатури послідовності позитивних чисел (довжина послідовності не обмежена).
21. Дано ціле число N . Знайти найменше ціле додатне число K , квадрат якого перевершує N : $K^2 > N$. Функцію отримання квадратного кореня не використовувати.
22. Знайти кількість чотиризначних чисел, що діляться на 23 і на останню свою цифру.
23. Дано цілі додатні числа N і K . Використовуючи тільки операції додавання та віднімання, знайти цілу частину від ділення N на K , а також залишок від ділення.
24. Дані додатні числа A , B , C . На прямокутнику розміру $A * B$ розміщено максимально можлива кількість квадратів, зі стороною C (без накладання). Знайти кількість квадратів, розміщених на прямокутнику. Операції множення і ділення не використовувати.
25. Знайти найближче число Фібоначчі до заданого числа.

Завдання 2.

1. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення, знайти число, отримане при прочитанні числа N справа на ліво .
2. Написати програму, яка визначає максимальне число з введеної з клавіатури послідовності позитивних чисел (довжина послідовності не обмежена).
3. Написати програму, яка обчислює суму і середнє арифметичне послідовності чисел (довжина послідовності не обмежена), які вводяться з клавіатури.
4. Скласти програму-гру «Більше-Менше». Комп'ютер генерує випадкове ціле число, а користувач намагається вгадати згенероване число. Комп'ютер надає користувачу підказку, чи є введене число більшим або меншим від загаданого. Також з клавіатури вводиться максимальна кількість спроб.
5. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення визначити, чи є в запису числа N непарні цифри.
6. Скласти програму для пошуку суми цифр цілого числа, що вводиться з клавіатури.
7. Знайдіть перше двозначне число, рівне квадрату числа його одиниць, складеному з кубом числа його десятків.
8. Скласти програму, яка знаходить найбільший спільний дільник двох цілих чисел, що вводяться з клавіатури.
9. Дано ціле число N . Знайти найменше ціле додатне число K , при якому виконується нерівність $3^K > N$.
10. Знайти число Фібоначчі, порядковий номер якого вводиться з клавіатури.

11. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення, знайти кількість і суму його цифр.
12. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення, вивести всі його цифри, починаючи з самої правої (розряду одиниць).
13. Дано ціле число N . Використовуючи операції ділення без остачі та взяття залишку від ділення визначити, чи присутня у записі числа N цифра «2».
14. Спортсмен-лижник почав тренування, пробігши в перший день 5 км. Кожен наступний день він збільшував довжину пробігу на P відсотків від пробігу попереднього дня. За даним P визначити, після якого дня сумарний пробіг лижника за всі дні перевищить 100 км, і вивести знайдену кількість днів K (ціле) і сумарний пробіг S (дійсне число).
15. Знайти усі двозначні числа, такі що, якщо до суми цифр цього числа додати квадрат цієї суми, то отримаємо це ж число.
16. Дані додатні числа A, B, C . На прямокутнику розміру $A * B$ розміщено максимально можлива кількість квадратів, зі стороною C (без накладання). Знайти кількість квадратів, розміщених на прямокутнику. Операції множення і ділення не використовувати.
17. Знайти кількість чотиризначних чисел, що діляться на 23 і на останню свою цифру.
18. Дано ціле число N . Вивести найменше з цілих чисел K , для яких сума $1 + 2 + \dots + K$ буде більше або дорівнює N , і саму цю суму.
19. Дано цілі додатні числа N і K . Використовуючи тільки операції додавання та віднімання, знайти цілу частину від ділення N на K , а також залишок від ділення.
20. Вивести на друк усі тризначні числа, які діляться націло на власні цифри, а також перевертиш цього числа ділиться націло на свої цифри.

21. Дано ціле число N . Знайти найменше ціле додатне число K , квадрат якого перевершує N : $K^2 > N$. Функцію отримання квадратного кореня не використовувати.
22. Знайти найближче число Фібоначчі до заданого числа.
23. Дано ціле число N . Перевірити чи є число N числом Фібоначчі.
24. Дано число. Визначити, чи є це число степенем двійки.
25. Початковий вклад у банку дорівнює 5000 грн. Через місяць розмір вкладу збільшується на P процентів від наявний суми. За даним P визначити, через скільки місяців розмір вкладу перевищить 7000 грн і вивести знайдену кількість місяців K (ціле число) і остаточний розмір вкладу S (дійсне число).

Контрольні питання

1. Дайте опис циклу **while**.
2. Дайте опис циклу **do/while**.
3. Намалюйте узагальнену блок-схему циклу **while**.
4. Намалюйте узагальнену блок-схему циклу **do/while**.
5. Що таке цикл з передумовою?
6. Що таке цикл з післяумовою?
7. В яких випадках доцільно використовувати цикл **do/while**, а в яких **while**?
8. Написати програму. Умову і вимоги до задачі надає викладач.

ЛАБОРАТОРНА РОБОТА №10. ОДНОВИМІРНІ МАСИВИ. ЧАСТИНА 1. БАЗОВІ ОПЕРАЦІЇ З МАСИВАМИ

Мета

Засвоїти поняття масиву даних. Розуміти як масиви зберігаються в пам'яті ЕОМ та способи адресації елементів масиву Отримати досвід використання масивів для вирішення простих задач.

Стислі теоретичні відомості

Масив – це сукупність даних одного типу, об'єднаних однією назвою. Приклади масивів з навколишнього світу: вулиця з будинками, де будинки є елементами масиву, колона автомобілів, шафа з книжками тощо.

Масив (структура даних) – це сукупність даних одного типу, об'єднаних однією назвою, що зберігаються в одному місці пам'яті комп'ютера та впорядковані за номерами комірок. Номера комірок називають індексами.

Масив в мові C# відноситься до змінних-посилань, тобто вони розташовуються в динамічній області пам'яті, тому створення масиву починається з виділення пам'яті під його елементи. Елементами масиву можуть бути величини як значущих, так і посилальних типів (у тому числі масиви). Масив значущих типів зберігає певні конкретні значення, а масив з даними посилальних типів – посилання на елементи. Всім елементам при створенні масиву присвоюються значення за замовчуванням: нулі для значущих типів і **null** – для посилальних.

Елементи масиву нумеруються з нуля, тому максимальний індекс елемента завжди на одиницю менше розмірності. Для звернення до елемента масиву після імені масиву вказується номер елемента в квадратних дужках, наприклад:

m[0]	b[3]
------	------

З елементом масиву можна виконувати всі дії, що припустимі для змінних такого ж типу. При роботі з масивом автоматично виконується

контроль виходу за його межі: якщо значення індексу виходить за межі масиву, генерується виключення під час роботи програми.

Масиви одного типу можна присвоювати один одному . При цьому відбувається присвоювання посилань, а не елементів, як і для будь-якого іншого об'єкта посилального типу.

У мові C # можна виділити три різновиди масивів: одновимірні, багатовимірні і ступінчасті (невирівняні).

В одновимірних масивах положення елемента задається тільки одним числом. Варіанти опису одновимірних масивів:

```
тип[] ім'я_масиву;  
тип[] ім'я_масиву = new тип[кількість елементів];  
тип[] ім'я_масиву = {значення елементів};  
тип[] ім'я_масиву = new тип[] {значення елементів};  
тип[] ім'я_масиву = new тип[кількість елементів]  
{значення елементів};
```

де **тип** визначає тип даних кожного елемента масиву.

Робоче завдання

Навчитись працювати з одновимірними масивами. Вміти використовувати масиви для вирішення поставлених задач. Досліди роботу масивів та способів індексації елементів масиву для одновимірних масивів.

Хід роботи

Написати програму для вирішення поставленого завдання. Всі вхідні масиви заповнюються випадковими цілими числами з діапазону [- 50; 50], окрім випадків зазначених в завданні.

Індивідуальне завдання

1. Дано масив розміру N. Вивести його елементи в зворотному порядку.

2. Дано масив розміру N . Вивести спочатку його елементи з парними індексами, а потім – з непарними.
3. Дано масив розміру N . Знайти найменший елемент в масиві.
4. Дано масив розміру N . Знайти кількість додатних елементів масиву.
5. Дано масив розміру N . Знайти кількість від'ємних елементів масиву.
6. Дано масив розміру N . Визначити, скільки разів зустрічається число 7 серед елементів масиву.
7. Дано масив розміру N . Знайти суму всіх невід'ємних елементів масиву.
8. Дано масив розміру N . Знайти найменший за модулем елемент масиву.
9. Дано масив розміру N . Знайти добуток від'ємних елементів масиву.
10. Дано масив розміру N . Знайти суму всіх елементів масиву, які мають парні індекси.
11. Дано масив розміру N . Знайти найбільший з елементів масиву, які мають непарні індекси.
12. Дано масив розміру N . Знайти суму елементів масиву, що перевищують число 5.
13. Дано дійсне число R і масив розміру N . Знайти елемент масиву, який найбільш близький до даного числа.
14. Дано число R і масив розміру N . Перетворити масив, замінивши всі його додатні елементи на число R .
15. Дано число R і масив розміру N . Перетворити масив, замінивши всі його елементи, котрі кратні числу 5, на число R .
16. Дано число R і масив розміру N . Перетворити масив, замінивши всі його від'ємні елементи на число R .
17. Дано число R і масив розміру N . Перетворити масив, замінивши всі його елементи, з парними індексами, на число R .
18. Дано число R і масив розміру N . Перетворити масив, замінивши всі його додатні елементи, котрі кратні 3, на число R .

19. Ввести одновимірний масив розміром N з клавіатури. Видалити з масиву всі від'ємні елементи, що не кратні семи, але кратні трьом.
20. Ввести одновимірний масив розміром N з клавіатури. Знайти найбільший з елементів масиву, які мають непарні індекси.
21. Ввести одновимірний масив розміром N з клавіатури. Знайти різницю між найбільшим і найменшим елементами масиву.
22. Ввести одновимірний масив розміром N з клавіатури. Поміняти місцями мінімальний і максимальний елементи масиву.
23. Ввести одновимірний масив розміром N з клавіатури. Підрахувати суму квадратів елементів масиву.
24. Ввести одновимірний масив розміром N з клавіатури. Знайти суму всіх елементів масиву, які мають парні індекси.
25. Ввести одновимірний масив розміром N з клавіатури. Знайти найбільший з від'ємних елементів масиву.

Контрольні питання

1. Що таке масив?
2. Наведіть приклад масиву.
3. Що таке індекс масиву?
4. Яким чином задається розмірність масиву?
5. Як масив представляється у пам'яті комп'ютера?
6. Наведіть приклад опису одновимірного масиву.
7. Як можна геометрично представити одновимірні масиви?
8. Чи можна змінну типу масив присвоїти іншій змінній того ж типу?
9. Чи може бути елементом масиву інший масив?
10. Що буде, якщо програма випадково вийде за межі масиву?
11. Написати програму. Умову і вимоги до задачі надає викладач.

ЛАБОРАТОРНА РОБОТА №11. ОДНОВИМІРНІ МАСИВИ. ЧАСТИНА 2. ПРИНЦИПИ ОБРОБКИ ДАНИХ В ОДНОМІРНИХ МАСИВАХ

Мета

Засвоїти принципи обробки даних в одновимірних масивах. Отримати практичний досвід використання масивів для вирішення задач.

Стислі теоретичні відомості

У мові C # можна виділити три різновиди масивів: одновимірні, багатовимірні і ступінчасті (невирівняні).

В одновимірних масивах положення елемента задається тільки одним числом. Варіанти опису одновимірних масивів:

```
тип[] ім'я_масиву;  
тип[] ім'я_масиву = new тип[кількість елементів];  
тип[] ім'я_масиву = {значення елементів};  
тип[] ім'я_масиву = new тип[] {значення елементів};  
тип[] ім'я_масиву = new тип[кількість елементів]  
{значення елементів};
```

де **тип** визначає тип даних кожного елемента масиву.

Робоче завдання

Навчитись працювати з одновимірними масивами. Вміти використовувати масиви для вирішення поставлених задач.

Хід роботи

Написати програму для вирішення поставленого завдання. Всі вхідні масиви заповнюються випадковими цілими числами з діапазону [- 50; 50], окрім випадків зазначених в завданні.

Індивідуальне завдання

1. Дано масив чисел. Знайти значення максимального елемента. Якщо таких елементів декілька, то визначити, скільки їх.
2. Дано масив чисел. Знайти, скільки в ньому пар однакових сусідніх елементів.
3. Масив цілих чисел з 10 елементів заповнити випадковим чином. Упорядкувати масив по зростанню. Знайти суму квадратних коренів непарних елементів масиву, вивести отримані результати.
4. Масив цілих чисел з 14 елементів заповнити випадковим чином. Упорядкувати масив за зростанням елементів. Вивести на екран максимальний і мінімальний елементи, знайти їх різницю.
5. Масив цілих чисел з 10 елементів заповнити випадковим чином. Упорядкувати масив по зростанню. Підрахувати кількість і добуток елементів масиву, більших 20 і менших 50, вивести отриманий результат.
6. Масив цілих чисел з 13 елементів заповнити випадковим чином. Упорядкувати масив за спаданням. Підрахувати кількість і суму елементів масиву, більших 10, вивести отриманий результат.
7. Масив цілих чисел з 10 елементів заповнити випадковим чином. Упорядкувати масив по зростанню. Підрахувати кількість і суму елементів масиву, які діляться на 5 без остачі.
8. Масив цілих чисел з 15 елементів заповнити випадковим чином. Упорядкувати масив за спаданням. Підрахувати суму квадратів елементів масиву, що кратні 3 і вивести отриманий результат.
9. Масив цілих чисел з 12 елементів заповнити випадковим чином. Упорядкувати масив за зростанням. Знайти кількість і суму парних елементів масиву, вивести отриманий результат.
10. Дано масив чисел. Знайти значення мінімального елемента. Якщо таких елементів декілька, то визначити, скільки їх.

11. Дано масив чисел. Відсортувати його за спаданням. Знайти середнє арифметичне елементів масиву, що більше 30.
12. Масив цілих чисел з 10 елементів заповнити випадковим чином. Упорядкувати масив по спаданню. Знайти суму квадратів непарних елементів масиву, вивести отримані результати.
13. Масив цілих чисел з N елементів заповнити випадковим чином. Зробити дзеркальне відображення елементів відносно середини масиву. Заборонено використовувати додаткові масиви.
14. Масив цілих чисел з N елементів заповнити випадковим чином. Відшукати найбільший та найменший елемент масиву та поміняти їх місцями. Якщо максимальних або мінімальних елементів декілька, то міняти місцями максимальний та мінімальний елемент масиву не потрібно.
15. Масив цілих чисел з 13 елементів заповнити випадковим чином. Упорядкувати масив за спаданням. Підрахувати кількість і суму елементів масиву, великих 10, вивести отриманий результат.
16. Масив цілих чисел з 10 елементів заповнити випадковим чином. Упорядкувати масив за зростанням. Підрахувати кількість і суму елементів масиву, які діляться на 5 без залишку.
17. Масив цілих чисел з N елементів заповнити випадковим чином. Упорядкувати масив за спаданням. Підрахувати суму квадратів елементів масиву і вивести отриманий результат.
18. Масив цілих чисел з 12 елементів заповнити випадковим чином. Упорядкувати масив за зростанням. Знайти кількість і суму парних елементів масиву, вивести отриманий результат.
19. Дано масив чисел. Знайти значення мінімального елемента. Якщо таких елементів декілька, то визначити, скільки їх.
20. Дано масив чисел. Знайти, скільки в ньому пар однакових сусідніх елементів.

21. Дано масив чисел. Відсортувати його за спаданням абсолютних значень елементів масиву.
22. Масив цілих чисел з 10 елементів заповнити випадковим чином. Упорядкувати масив по спаданню. Знайти суму квадратів непарних елементів масиву, вивести отримані результати.
23. Дано масив розміру N . Переставити в зворотному порядку елементи масиву, розташовані між його мінімальним і максимальним елементами .
24. Дано масив чисел. Відсортувати його за зростанням. Знайти середнє арифметичне елементів масиву, що менше 10.
25. Ввести з клавіатури масив цілих чисел з 10 елементів. Упорядкувати масив за зростанням. Знайти суму квадратів парних елементів масиву, вивести отримані результати.

Контрольні питання

1. Що таке масив?
2. Що таке індекс масиву?
3. Яким чином задається розмірність масиву?
4. Наведіть приклад опису одновимірного масиву.
5. Як вивести масив на екран?
6. Як виконати сортування масиву?
7. Як знайти суму елементів масиву?
8. Як знайти найбільший та найменший елементи масиву?
9. Написати програму. Умову і вимоги до задачі надає викладач.

ЛАБОРАТОРНА РОБОТА №12. ДВОМІРНІ МАСИВИ. ОСНОВИ РОБОТИ З МАТРИЦЯМИ

Мета

Отримати досвід використання двовимірних масивів. Засвоїти основні прийоми роботи з матрицями.

Стислі теоретичні відомості

Багатовимірним називається такий масив, який характеризується двома або більше вимірами, причому доступ до кожного елементу такого масиву здійснюється за допомогою певної комбінації двох або більше індексів.

Найпростішою формою багатовимірного масиву є двовимірний масив. Положення будь-якого елемента в двовимірному масиві визначається двома індексами. Такий масив можна уявити в вигляді таблиці, на рядки якої вказує один індекс, а на стовпці – інший.

Варіанти опису двовимірного масиву:

```
тип[, ] ім'я_масиву;
тип[, ] ім'я_масиву = new тип[розмір_1, розмір_2];
тип[, ] ім'я_масиву = {значення елементів};
тип[, ] ім'я_масиву = new тип[, ] {значення елементів};
тип[, ] ім'я_масиву = new тип[розмір_1, розмір_2]
{значення елементів};
```

Якщо список ініціалізації не заданий, розмірності можуть бути не тільки константами, але і виразами типу, що приводиться до цілого. До елементу двовимірного масиву звертаються, вказуючи номери рядка і стовпця, на перетині яких він розташований, наприклад:

$m[0,0]$	$a[6,4]$
----------	----------

Робоче завдання

Навчитись працювати з двовимірними масивами. Вміти використовувати масиви для вирішення поставлених задач. Досліди прийоми роботи та способи індексації елементів двовимірних масивів.

Хід роботи

Написати програму для вирішення поставленого завдання. Всі вхідні масиви заповнюються випадковими цілими числами з діапазону $[- 50; 50]$, окрім випадків зазначених в завданні. Схеми масивів наведені після текстових завдань.

Індивідуальне завдання

1. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.
2. Задана матриця цілих чисел розміром $(N;N)$. Знайти середнє арифметичне елементів в зафарбованій області. Заборонено використання додаткових масивів.
3. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.
4. Задана матриця цілих чисел розміром $(N;N)$. Знайти максимальний елемент в зафарбованій області. Заборонено використання додаткових масивів.
5. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.
6. Задана матриця цілих чисел розміром $(N;N)$. Знайти мінімальний елемент в зафарбованій області. Заборонено використання додаткових масивів.
7. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.
8. Задана матриця цілих чисел розміром $(N;N)$. Знайти суму елементів в зафарбованій області. Заборонено використання додаткових масивів.
9. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.

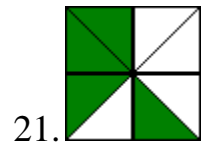
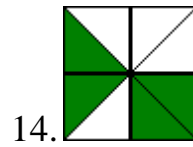
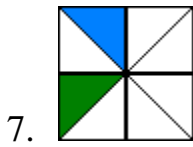
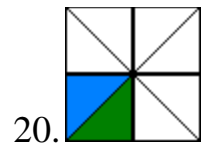
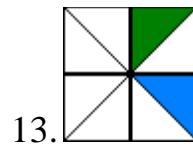
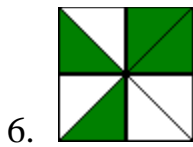
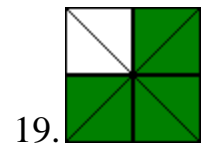
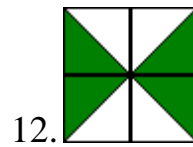
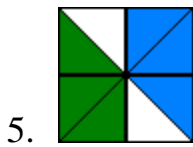
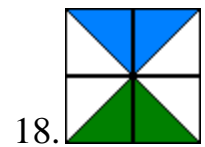
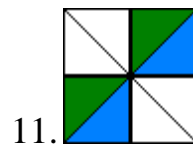
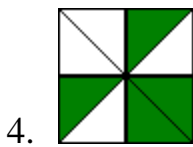
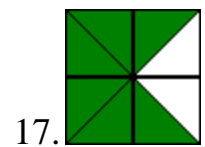
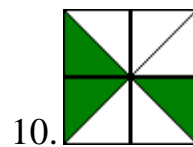
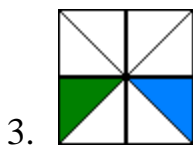
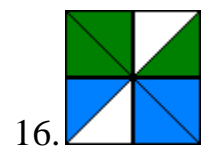
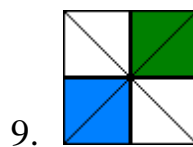
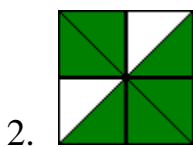
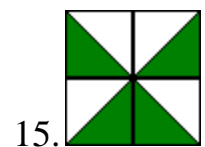
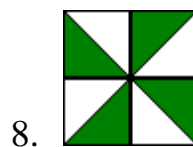
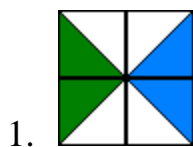
10. Задана матриця цілих чисел розміром $(N;N)$. Знайти максимальний елемент в зафарбованій області. Заборонено використання додаткових масивів.
11. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.
12. Задана матриця цілих чисел розміром $(N;N)$. Знайти середнє арифметичне елементів в зафарбованій області. Заборонено використання додаткових масивів.
13. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.
14. Задана матриця цілих чисел розміром $(N;N)$. Знайти мінімальний елемент в зафарбованій області. Заборонено використання додаткових масивів.
15. Задана матриця цілих чисел розміром $(N;N)$. Підрахувати кількість чисел кратних 5 в зафарбованій області. Заборонено використання додаткових масивів.
16. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.
17. Задана матриця цілих чисел розміром $(N;N)$. Знайти максимальний елемент в зафарбованій області. Заборонено використання додаткових масивів.
18. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.
19. Задана матриця цілих чисел розміром $(N;N)$. Підрахувати кількість чисел кратних 3 в зафарбованій області. Заборонено використання додаткових масивів.
20. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.

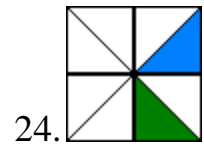
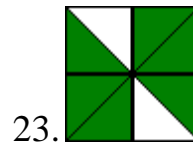
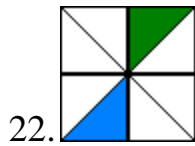
21. Задана матриця цілих чисел розміром $(N;N)$. Знайти максимальний елемент в зафарбованій області. Заборонено використання додаткових масивів.

22. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.

23. Задана матриця цілих чисел розміром $(N;N)$. Знайти мінімальний елемент в зафарбованій області. Заборонено використання додаткових масивів.

24. Задана матриця цілих чисел розміром $(N;N)$. Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.





Контрольні питання

1. Що таке масив?
2. Що таке індекс масиву?
3. Яким чином задається розмірність масиву?
4. Наведіть приклад опису одно-, дво-, трьох- вимірного масивів.
5. Якими способами можливо виконати індексацію трьох-вимірного масиву?
6. Як можна геометрично представити одно-, дво-, трьох- вимірні масиви?
7. Як вивести двовимірний масив на екран?
8. Яка різниця між масивом-вектором і масивом-матрицею?
9. Написати програму. Умову і вимоги до задачі надає викладач.

ЛІТЕРАТУРА

1. Д.В. Настенко, А. Б. Нестерко. Об'єктно-орієнтоване програмування. Частина 1. Основи об'єктно-орієнтованого програмування на мові С#: Навчальний посібник /– К.: НТУУ «КПІ», 2016. - 76с.
2. Обчислювальна техніка та програмування: Конспект лекцій (Частина 1) [Електронний ресурс] : навч. посіб. для студ. спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / КПІ ім. Ігоря Сікорського ; уклад.: Г.О.Труніна, Д.В.Настенко, А.Б.Нестерко. – Електронні текстові данні (1 файл: 1,52 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 117 с.
3. Голуб Б.М. С#. Концепція та синтаксис. Навч. посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2006. – 136 с.
4. Троелсен Э. Язык программирования С# 5.0 и платформа .NET 4.5, 6-е изд.: Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2013. – 1312 с.
5. Шилдт Г. С# 3.0. Полное руководство / Пер. с англ. – М.: Диалектика-Вильямс, 2009. – 992 с.
6. Котов, О.М. Язык С#: краткое описание и введение в технологии программирования: учебное пособие / О.М. Котов. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 208 с.
7. Уотсон К. Microsoft Visual С# 2008. Базовый курс / К. Уотсон, К. Нейгел, Я.Х. Педерсен, Дж. Д. Рид, М. Скиннер, Э. Уайт. / Пер. с англ. – М.: Диалектика-Вильямс, 2009. – 1216 с.