### Кам’янець-Подільський національний університет імені Івана Огієнка

### КАФЕДРА КОМП’ЮТЕРНИХ НАУК

### Навчальна дисципліна «Об’єктно-орієнтоване програмування»

### ЛАБОРАТОРНА РОБОТА #0111

## Тема:

Обробка рядків і тексту. Робота з датою і часом

Варіант №1

**Виконав**:  
студент 1-го курсу  
групи KNms1-B24  
Білик Я. Ю.

**Прийняв**:  
старший викладач,  
Слободянюк О. В.

### Кам’янець-Подільський – 2025

1. **Короткі теоретичні відомості.**

Рядок є об'єктом типу String, значенням якого є текст. По суті, текст зберігається у вигляді послідовної доступною тільки для читання колекції об'єктів Char. В кінці рядка в мові C# відсутній символ, який закінчується на NULL; тому рядок C# може містити будь-яке число символів NULL ( "\0"). Властивість рядка Length представляє число об'єктів Char, що містяться в цьому рядку, а не число символів Юнікоду. Для доступу до окремих точок в рядку коду Юникода використовується об'єкт StringInfo object.

Рядкові об'єкти є незмінними: після створення їх не можна змінити. Всі методи [String](https://translate.google.com/translate?hl=uk&prev=_t&sl=ru&tl=uk&u=https://msdn.microsoft.com/uk-ua/library/system.string.aspx) і оператори C#, які, як можна було б уявити, змінюють рядок, в дійсності повертають результати в новий рядковий об'єкт. У наступному прикладі, коли вміст рядків s1 і s2 об'єднується в один рядок, два вихідні рядки не змінюються. Оператор + = створює новий рядок з об'єднаним вмістом. Цей новий об'єкт присвоюється змінній s1, а вихідний об'єкт, який був присвоєний рядку s1, звільняється, оскільки жодна змінна не містить посилання на нього.

string s1 = "A string is more ";

string s2 = "than the sum of its chars.";

s1 += s2;

System.Console.WriteLine(s1);

Оскільки "зміною" рядка в дійсності є створенням нового рядка, необхідна обережність при створенні посилань на рядки. Якщо створити посилання на рядок, а потім "змінити" вихідний, то посилання буде як і раніше вказувати на вихідний об'єкт, а не на новий об'єкт, який був створений при зміні рядка. Це демонструється в наступному коді.

string s1 = "Hello ";

string s2 = s1;

s1 += "World";

System.Console.WriteLine(s2);

Підрядки

Підрядком є послідовність символів, що містяться в рядку. Метод Substring використовується для створення нового рядка на підставі частини початкового рядка. Одне або кілька входжень підрядка можна знайти з використанням методу IndexOf. Метод Replace використовується для заміни всіх входжень заданого підрядка новим рядком. Так само, як і метод Substring, метод Replace фактично повертає новий рядок і не змінює вихідний.

string s3 = "Visual C# Express";

System.Console.WriteLine(s3.Substring(7, 2));

// виведеться: "C#"

System.Console.WriteLine(s3.Replace("C#", "Basic"));

// виведеться: "Visual Basic Express"

// Index values are zero-based

int index = s3.IndexOf("C");

// index = 7

Доступ до окремих знаків

Нотацію масиву можна використовувати зі значенням індексу, щоб отримати доступ до окремих символів, як в наступному прикладі:

string s5 = "Printing backwards";

for (int i = 0; i<s5.Length; i++)

{

System.Console.Write(s5[s5.Length - i - 1]);

}

// Output: "sdrawkcab gnitnirP"

Структура DateTime

Для роботи з датами і часом в.NET призначена структура DateTime. Вона представляє дату і час від 00:00:00 1 січня 0001 року до 23:59:59 31 грудня 9999 року.

Для створення нового об’єкту DateTime можна використовувати конструктор. Пустий конструктор створює початкову дату:

DateTime date1 = new DateTime();

Console.WriteLine(date1);

Щоб задати конкретну дату, потрібно використати один з конструкторів, що приймають параметри:

DateTime date1 = new DateTime(2017, 7, 20);

Console.WriteLine(date1); // 20.07.2017 0:00:00

Якщо необхідно отримати поточну дату чи час, то можна використати ряд властивостей DateTime:

Console.WriteLine(DateTime.Now);

Console.WriteLine(DateTime.UtcNow);

Console.WriteLine(DateTime.Today);

1. **Повні умови завдань.**

**Задача 1**

Написати програму, яка замінює в початковому рядку символів всі одиниці нулями і всі нулі одиницями. Заміна має виконуватися, починаючи із заданої позиції рядка.

Написати програму, яка підраховує кількість днів, до вказаної дати, якщо цей день вже минув, видати відповідне повідомлення.

Дано рядок, який містить три різні дати, розділені комою. Кожна дата – це число, місяць і рік. Знайти:

1. рік з найменшим номером;
2. всі весняні дати;
3. саму пізнішу дату.
4. **Лістинги програм.**

Задача 1

using System;

using System.Globalization;

using System.Linq;

class Program

{

static void Main()

{

Console.OutputEncoding = System.Text.Encoding.UTF8;

// === Завдання 1 ===

Console.WriteLine("Завдання 1. Замінити 0 на 1 і 1 на 0 з позиції:");

Console.Write("Введіть рядок з 0 і 1: ");

string binaryString = Console.ReadLine();

Console.Write("З якої позиції почати заміну (0-based index): ");

int position = int.Parse(Console.ReadLine());

string result = ReplaceBits(binaryString, position);

Console.WriteLine("Результат: " + result);

// === Завдання 2 ===

Console.WriteLine("\nЗавдання 2. Підрахунок днів до дати:");

Console.Write("Введіть цільову дату (у форматі рррр-мм-дд): ");

DateTime targetDate;

if (DateTime.TryParse(Console.ReadLine(), out targetDate))

{

int days = (targetDate - DateTime.Today).Days;

if (days > 0)

Console.WriteLine($"До цієї дати залишилось {days} днів.");

else if (days == 0)

Console.WriteLine("Це сьогодні!");

else

Console.WriteLine("Ця дата вже минула.");

}

else

{

Console.WriteLine("Невірний формат дати.");

}

// === Завдання 3 ===

Console.WriteLine("\nЗавдання 3. Обробка 3 дат:");

Console.Write("Введіть 3 дати через кому (формат: dd.MM.yyyy,dd.MM.yyyy,...): ");

string input = Console.ReadLine();

string[] dateParts = input.Split(',');

if (dateParts.Length != 3)

{

Console.WriteLine("Треба ввести рівно 3 дати.");

return;

}

DateTime[] dates = new DateTime[3];

bool allParsed = true;

for (int i = 0; i < 3; i++)

{

if (!DateTime.TryParseExact(dateParts[i].Trim(), "dd.MM.yyyy", CultureInfo.InvariantCulture, DateTimeStyles.None, out dates[i]))

{

allParsed = false;

break;

}

}

if (!allParsed)

{

Console.WriteLine("Помилка у форматі дат. Використовуйте dd.MM.yyyy");

return;

}

// a) рік з найменшим номером

int minYear = dates.Min(d => d.Year);

Console.WriteLine($"a) Найменший рік: {minYear}");

// b) весняні дати (березень, квітень, травень)

Console.WriteLine("b) Весняні дати:");

foreach (var d in dates)

{

if (d.Month >= 3 && d.Month <= 5)

{

Console.WriteLine(d.ToString("dd.MM.yyyy"));

}

}

// c) найпізніша дата

DateTime latest = dates.Max();

Console.WriteLine("c) Найпізніша дата: " + latest.ToString("dd.MM.yyyy"));

}

static string ReplaceBits(string input, int startPos)

{

if (startPos < 0 || startPos >= input.Length)

{

return input;

}

char[] chars = input.ToCharArray();

for (int i = startPos; i < chars.Length; i++)

{

if (chars[i] == '0') chars[i] = '1';

else if (chars[i] == '1') chars[i] = '0';

}

return new string(chars);

}

}

1. **Результати роботи програм.**

Задача 1

