

簡答題

Ch8 2.現在有一個 C#字串 `str`，請寫出字串方法 `ToUpper()`、`Substring(2,4)`和 `IndexOf("程式")`的輸出結果，如下所示：`string str = "Visual C# 程式設計範例教本";`

VISUAL C# 程式設計範例教本

sual

10

Ch8 6.請簡單說明什麼是搜尋與排序?請問搜尋方法依照搜尋的資料可以分為哪兩種?

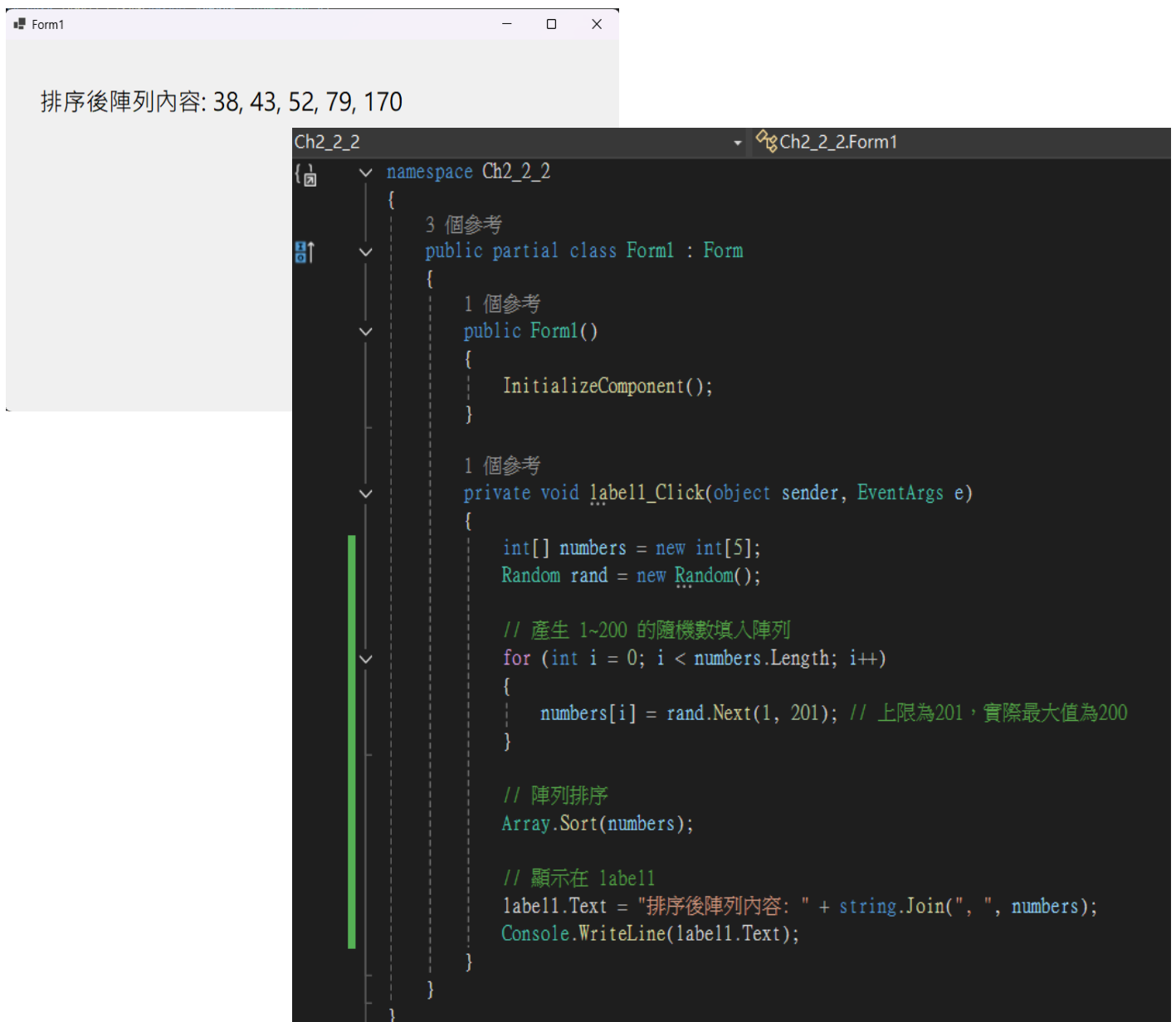
搜尋是在資料中找尋是否存在與特定的值或稱鍵值相同的資料；

排序是將一些工作依照特定原則排列籌遞增或遞減順序來排序。

分為 沒有排序的資料 以及 已經排序的資料。

實作題

Ch8 2.請建立 C#應用程式宣告 5 個元素的一維陣列後，使用亂數類別來產生陣列的元素值，其範圍是 1~200 的整數，然後將陣列內容排序後，顯示在標籤控制項



The screenshot displays a Windows Forms application window titled "Form1". Inside the window, a label shows the text "排序後陣列內容: 38, 43, 52, 79, 170". Below the window, the Visual Studio code editor shows the source code for `Ch2_2_2.Form1`. The code declares a `Form1` class that inherits from `Form`. It includes a constructor `Form1()` that calls `InitializeComponent()`. A click event handler `label1_Click` is attached to a `label1` control. In this handler, an array of 5 integers is created, populated with random values between 1 and 201 using `Random.Next(1, 201)`. The array is then sorted using `Array.Sort(numbers)`, and the sorted values are displayed in `label1` using `string.Join(", ", numbers)`. A `Console.WriteLine` statement also outputs the sorted array.

```
Ch2_2_2
{
    namespace Ch2_2_2
    {
        3 個參考
        public partial class Form1 : Form
        {
            1 個參考
            public Form1()
            {
                InitializeComponent();
            }

            1 個參考
            private void label1_Click(object sender, EventArgs e)
            {
                int[] numbers = new int[5];
                Random rand = new Random();

                // 產生 1~200 的隨機數填入陣列
                for (int i = 0; i < numbers.Length; i++)
                {
                    numbers[i] = rand.Next(1, 201); // 上限為201，實際最大值為200
                }

                // 陣列排序
                Array.Sort(numbers);

                // 顯示在 label1
                label1.Text = "排序後陣列內容: " + string.Join(", ", numbers);
                Console.WriteLine(label1.Text);
            }
        }
    }
}
```

Ch8 4.請分別建立 arrMin()和 arrMax() 函數傳入整數陣列，傳回值是陣列的最小值和最大值，請建立 C# 應用程式的表單介面讓使用者輸入 6 個數字，然後找出其中的最小值和最大值。

```
Form1.cs  x Form1.cs [設計]
Ch8_2_4  Ch8_2_4.Form1

3 個參考
public partial class Form1 : Form
{
    1 個參考
    public Form1()
    {
        InitializeComponent();
    }
    1 個參考
    public void bubbleSort(int[] data)
    {
        int i, j, l;
        int n = data.Length;

        for (i = 0; i < n - 1; i++)
        {
            for (j = 0; j < n - 1 - i; j++)
            {
                if (data[j + 1] < data[j])
                {
                    l = data[j + 1];
                    data[j + 1] = data[j];
                    data[j] = l;
                }
            }
        }
    }

    1 個參考
    private void Form1_Load(object sender, EventArgs e)
    {
        textBox1.Focus();
    }

    1 個參考
    private void button1_Click(object sender, EventArgs e)
    {
        int[] data = new int[6];
        data[0] = Convert.ToInt32(textBox1.Text);
        data[1] = Convert.ToInt32(textBox2.Text);
        data[2] = Convert.ToInt32(textBox3.Text);
        data[3] = Convert.ToInt32(textBox4.Text);
        data[4] = Convert.ToInt32(textBox5.Text);
        data[5] = Convert.ToInt32(textBox6.Text);
        bubbleSort(data);
        textBox1.Text = data[0].ToString();
        textBox2.Text = data[1].ToString();
        textBox3.Text = data[2].ToString();
        textBox4.Text = data[3].ToString();
        textBox5.Text = data[4].ToString();
        textBox6.Text = data[5].ToString();
        label1.Text = "最小值：" + data[0];
        label2.Text = "最大值：" + data[5];
    }
}
```

簡答題

Ch9 1.請簡單說明物件導向的應用程式開發和傳統應用程式開發的差異?

前者是將資料和操作一起思考，主要找出參與中相互之間的關係並於分配後合力解決問題；後者則是將資料和操作分開來思考，主要在於找出解決辦法的演算法來建立後續的程序或函數。

Ch9 6.請舉例說明 **private**、**protected** 和 **public** 三種修飾子的用途和差異?甚麼是「工具方法」(Utility Methods)?

都是作用於成員變數或方法之中的，差異在指定成員的存取層級不同，其用途分別為 **private** 只能在類別本身呼叫或存取；**protected** 可以在類別本身和其子類別存取或呼叫；**public** 是此類別建立物件對外的使用者介面。

工具方法是指一些執行通用、可重複使用任務的方法。

實作題

Ch9 2.

```
Ch9_2
using System;

3 個參考
public class Box //這裡是類別
{
    private double Width;
    private double Height;
    private double Length;

    1 個參考
    public Box(double width, double height, double length) //建構子
    {
        Width = width;
        Height = height;
        Length = length;
    }

    1 個參考
    public double Volume() //計算體積
    {
        return Width * Height * Length;
    }

    1 個參考
    public double Area() //計算面積
    {
        return 2 * (Width * Height + Height * Length + Width * Length);
    }
}
```

```
0 個參考
public class Program //主程式 類別
{
    0 個參考
    public static void Main(string[] args) //函數
    {
        Console.WriteLine("輸入長度:");
        double length = double.Parse(Console.ReadLine());
        Console.WriteLine("輸入寬度:");
        double width = double.Parse(Console.ReadLine());
        Console.WriteLine("輸入高度:");
        double height = double.Parse(Console.ReadLine());

        Box box = new Box(length, width, height); //將Box類別中的資料呼叫出來並套用主程式的長寬高

        Console.WriteLine("Volume: " + box.Volume());
        Console.WriteLine("Area: " + box.Area());
    }
}
```

Ch9 4.

```
Ch9_4
using System;
5 個參考
public class PhoneList
{
    public string HomePhone;
    public string BusinessPhone;
    public string CellPhone;

    1 個參考
    public PhoneList(string homePhone, string businessPhone, string cellPhone)
    {
        HomePhone = homePhone;
        BusinessPhone = businessPhone;
        CellPhone = cellPhone;
    }
}

3 個參考
public class Cards
{
    private string Name;
    private string Occupation;
    private string Age;
    public PhoneList Phone;
    private string Email;

    1 個參考
    public Cards(string name, string occupation, string age, string email, PhoneList phone)
    {
        Name = name;
        Occupation = occupation;
        Age = age;
        Email = email;
        Phone = phone;
    }
}
```

```
1 個參考
public void GetCard()
{
    Console.WriteLine("Name : " + Name);
    Console.WriteLine("Occupation : " + Occupation);
    Console.WriteLine("Age : " + Age);
    Console.WriteLine("Email : " + Email);
    Console.WriteLine("Phone : ");
    Console.WriteLine("    Home : " + Phone.HomePhone);
    Console.WriteLine("    Business : " + Phone.BusinessPhone);
    Console.WriteLine("    Cell : " + Phone.CellPhone);
}

0 個參考
public static void Main(string[] args)
{
    Console.Write("輸入住家電話 : ");
    string homePhone = Console.ReadLine();
    Console.Write("輸入公司電話 : ");
    string businessPhone = Console.ReadLine();
    Console.Write("輸入行動電話 : ");
    string cellPhone = Console.ReadLine();
    PhoneList phone = new PhoneList(homePhone, businessPhone, cellPhone);

    Console.Write("輸入姓名 : ");
    string Name = Console.ReadLine();
    Console.Write("輸入職業 : ");
    string Occupation = Console.ReadLine();
    Console.Write("輸入年齡 : ");
    string Age = Console.ReadLine();
    Console.Write("輸入電子郵件 : ");
    string Email = Console.ReadLine();
    Console.WriteLine();
    Cards card = new Cards(Name, Occupation, Age, Email, phone);
    card.GetCard();
}
}
```