

第八章

簡答 2

```
2     string str = "Visual C# 程式設計範本教本";
3
4     string upperStr = str.ToUpper();
5     Console.WriteLine(upperStr);
6
7     string substr = str.Substring(2, 4);
8     Console.WriteLine(substr);
9
10    int index = str.IndexOf("程式");
11    Console.WriteLine("IndexOf(\"程式\"): " + index);
```



簡答 6

在資料中找尋是否存在與鍵值相同的資料

分為二元搜尋法和線性搜尋法

實作 2

```
11 namespace WindowsFormsApp1
12 {
13     public partial class Form1 : Form
14     {
15         private int i;
16
17         public Form1()
18         {
19             InitializeComponent();
20         }
21
22         private void button1_Click(object sender, EventArgs e)
23         {
24             int[] numbers = new int[5];
25
26             Random rand = new Random();
27
28             for (int i = 0; i < numbers.Length; i++)
29             {
30                 numbers[i] = rand.Next(1, 201);
31             }
32             string output = "原始陣列內容:\n";
33             output += string.Join(" ", numbers) + "\n";
34
35             Array.Sort(numbers);
36
37             output += "排序後陣列內容:\n";
38             output += string.Join(" ", numbers);
39
40             label1.Text = output;
41         }
42     }
43 }
```

Form1

原始陣列內容:
82 93 166 143 13
排序後陣列內容:
13 82 93 143 166

排序

實作 4

```

11 namespace WindowsFormsApp3
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19         int arrMin(int[] arr)
20         {
21             return arr[0];
22         }
23         int arrMax(int[] arr)
24         {
25             return arr[arr.Length - 1];
26         }
27         private void button1_Click(object sender, EventArgs e)
28         {
29             int[] num = new int[6];
30             num[0] = Convert.ToInt32(textBox1.Text);
31             num[1] = Convert.ToInt32(textBox2.Text);
32             num[2] = Convert.ToInt32(textBox3.Text);
33             num[3] = Convert.ToInt32(textBox4.Text);
34             num[4] = Convert.ToInt32(textBox5.Text);
35             num[5] = Convert.ToInt32(textBox6.Text);
36
37             Array.Sort(num);
38             label7.Text = "最大值:" + Convert.ToString(arrMax(num));
39             label8.Text = "最小值:" + Convert.ToString(arrMin(num));
40         }
41     }
42 }

```

Form1

數字1: 78

數字2: 9

數字3: 45

數字4: 34

數字5: 25

數字6: 44

最大值:78

最小值:9

確認

第九章

問答 1

傳統應用程式開發是將資料和操作分開來思考，著重如何找出解決問題的演算法來建立程序或函數，物件導向程式開發是將資料和操作一起思考，主要是找出參與物件和其他物件之間的關係

問答 6

private：程式變數或方法只能在類別本身呼叫或存取，若類別沒有特別宣告，預設就是 **private**

public：成員變數或方法是此類別建立物件對外的使用介面，可以讓 **c#** 程式碼呼叫物件的成員方法或存取成員變數

protect：成員變數或方法可以在類別本身和其子類別存取或呼叫，類別的子類別就是繼承

工具方法指的是一組不屬於特定物件實例，而是提供通用輔助功能的函式或方法。這些方法通常用來執行常見的、重複性的任務

實作 2

```

2  public class Box
3  {
4      public double Width;
5      public double Height;
6      public double Length;
7
8      public Box(double width, double height, double length)
9      {
10         Width = width;
11         Height = height;
12         Length = length;
13     }
14
15     public double Volume()
16     {
17         return Width * Height * Length;
18     }
19
20     public double Area()
21     {
22         return 2 * (Width * Length + Width * Length + Height * Length);
23     }
24 }

```

BOX
- Width: double - Height: double - Length: double
+ Box(w, h, l) + Volume(): double + Area(): double

實作 4

```
3 public class Cards
4 {
5     public class PhoneList //內部類別
6     {
7         public string HomePhone; //住家電話
8         public string BusinessPhone; //公司電話
9         public string CellPhone; //手機電話
10        public PhoneList(string homePhone, string businessPhone, string cellPhone)
11        {
12            this.HomePhone = homePhone;
13            this.BusinessPhone = businessPhone;
14            this.CellPhone = cellPhone;
15        }
16    }
17    public string Name; //姓名
18    public string Occupation; //職業
19    public byte Age; //年齡
20    public PhoneList Phone; //電話
21    public string Email; //電子郵件
22    public Cards(string name, string occupation, byte age, string homePhone, string businessPhone,
23                string cellPhone, string email) //建構子
24    {
25        this.Name = name;
26        this.Occupation = occupation;
27        this.Age = age;
28        this.Phone = new PhoneList(homePhone, businessPhone, cellPhone);
29        this.Email = email;
30    }
31    public string GetCard()
32    {
33        return $"姓名: {Name}\n 職位: {Occupation}\n 年齡: {Age}\n 住家電話: {Phone.HomePhone}\n" +
34            $"公司電話: {Phone.BusinessPhone}\n 手機電話: {Phone.CellPhone}";
35    }
36 }
```