# 第一部分

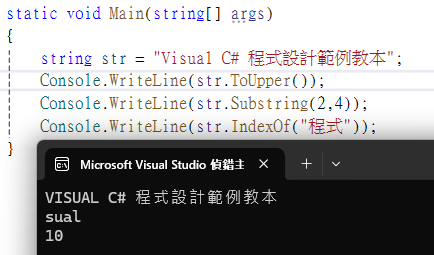
1. 第8章 簡答題2

Ans：

ToUpper()：VISUAL C# 程式設計範例教本

Substring(2,4)：sual

IndesOF(“程式”)：10



1. 第8章 簡答題6

Ans：

「排序」（Sorting）和「搜尋」（Searching）在計算機科學屬於資料結構與演算法的範疇。電腦有相當多的執行時間都是在處理資料的排序和搜尋，排序和搜尋實際應用在資料庫系統、編譯程式和作業系統之中。

* 陣列以整數值的大小，將陣列內容依遞增的順序來排序。其排序結果如下所示：

data[0]=34 data[1]=45 data[2]=78 data[3]=89

陣列data[]已經完成排序，其大小順序如下所示：

data[0] < data[1] < data[2] < data[3]

* 搜尋是在資料中找出是否存在與鍵值相同的資料，如果資料存在，就進行後續的資料處理。例如：查尋電話簿是為了找朋友的電話號碼，然後與他連絡。在書局找書也是為了找到後買回家閱讀。
* 搜尋方法依照搜尋的資料分為兩種，如下所示：
  + 沒有排序的資料：針對沒有排序的資料執行搜尋，我們需要從資料的第1個元素開始比較，從頭到尾以確認資料是否存在。
  + 已經排序的資料：因為資料已經排序，所以搜尋就不需從頭開始一一比較。例如：在電話簿找電話，相信沒有人是從電話簿的第1頁開始找，而是直接從姓名出現的頁數開始找，因為電話簿已經依照姓名進行排序。

1. 第8章 實作題2

Ans：



1. 第8章 實作題4

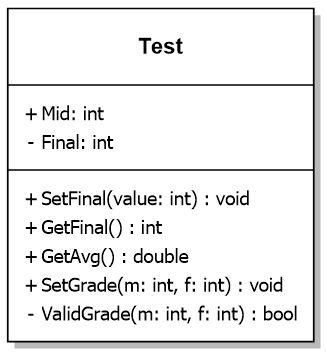
Ans：



1. 第9章 簡答題6

Ans：

* private修飾子：成員變數或方法只能在類別本身呼叫或存取，如果沒有使用修飾子，預設是private。
* protected修飾子：成員變數或方法可以在類別本身和其子類別存取或呼叫，類別的子類別稱為繼承。
* public修飾子：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。



public class Test {

public int Mid;

private int Final;

public void SetFinal(int value) { … }

public int GetFinal() { … }

public string GetAvg() { … }

public void SetGrade(int m, int f) { … }

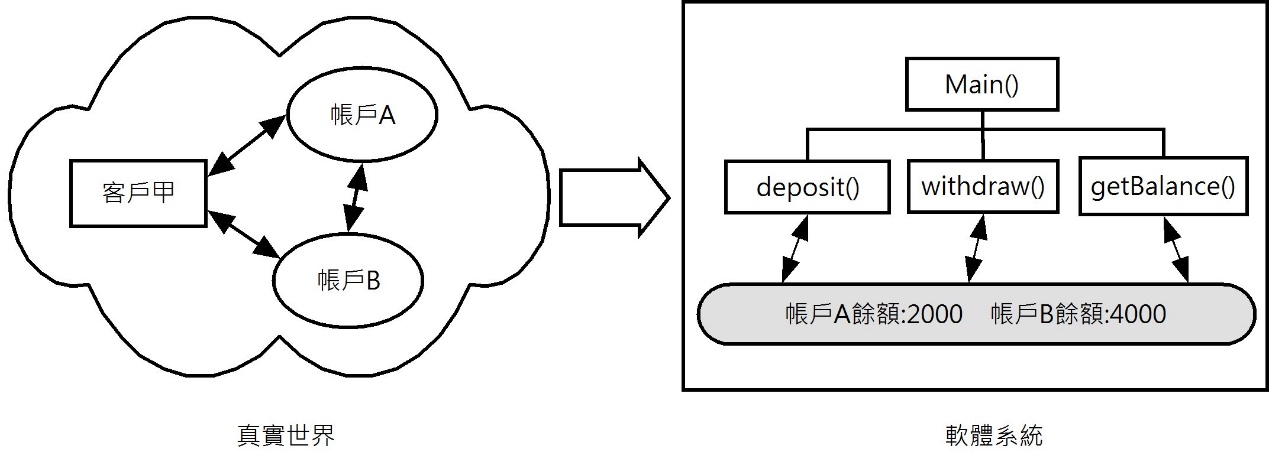
private bool ValidGrade(int m, int f) { … }

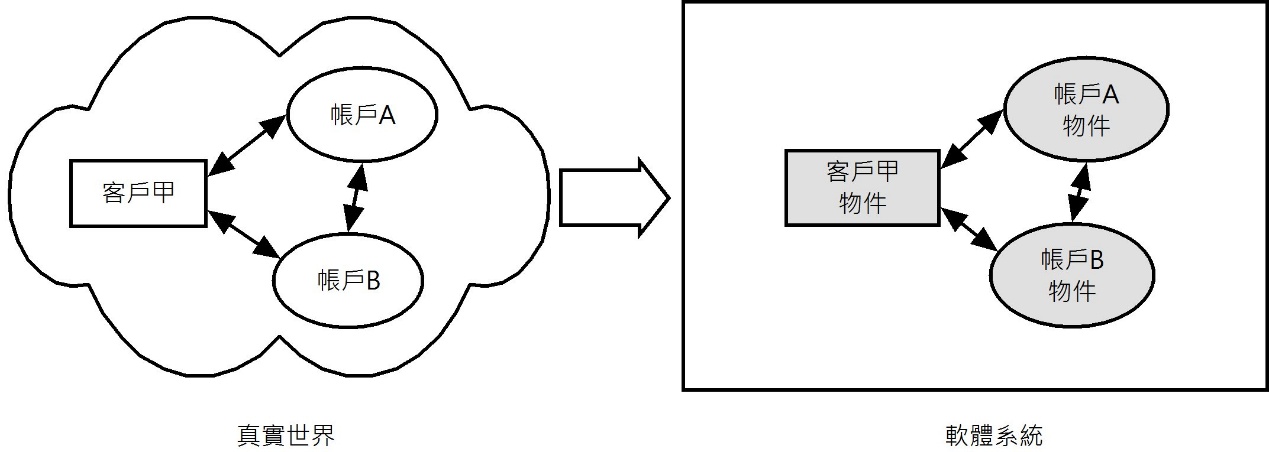
}

* 除了ValidGrade()方法宣告成private外，其他方法都是類別對外的使用介面。ValidGrade()方法因為只提供給類別的SetGrade()成員方法呼叫，所以宣告成private，這種方法也稱為「工具方法」（Utility Methods）。

1. 第9章 簡答題1

Ans：

* 傳統的應用程式開發是將資料和操作分開來思考，著重於如何找出解決問題的程序或函數，即演算法。
  + 例如：一家銀行的客戶甲擁有帳戶A和B兩個帳戶，客戶甲在查詢帳戶A的餘額後，從帳戶A提出1000元，然後將1000元存入帳戶B。
  + 
* 物件導向的應用程式開發是將資料和操作一起思考，其主要工作是找出參與物件和物件之間的關係，並且透過這些物件的通力合作來解決問題。
  + 例如：針對上一節相同的銀行存提款問題，使用物件導向應用程式開發建立的模型，如下圖所示：



1. 第9章 實作題2

Ans：

|  |
| --- |
| Box |
| -Width: double  -Height: double  -Length: double |
| +Box(width: double, height: double, length: double)  +Volume(): double  +Area(): double |

public class Box

{

private double width;

private double height;

private double length;

public Box( double width, double height, double length)

{

this.width = width;

this.height = height;

this.length = length;

}

public double Volume()

{

return width \* height \* length;

}

public double Area()

{

Return 2\* (width \* height + width \* length + height \* length);

}

}

1. 第9章 實作題4

Ans：

|  |
| --- |
| Cards |
| -Name: string  -Occupation: string  -Age: int  -phone: PhoneList  -Email: string |
| +Cards(name: string, occupation: string, age: int, phone: PhoneList, email: string)  +GetCard(): void |
|  |
| PhoneList |
| -HomePhone: string  -BusinessPhone: string  -CellPhone: string |
| +PhoneList(homephone : string, businessphone: string, cellphone: string) |

public class Cards{

private string name;

private string occupation;

private int age;

private PhoneList phone;

private string email;

public Cards(string name, string occupation, int age, PhoneList phone, string email)

{

this.name = name;

this.occupation = occupation;

this.age = age;

this.phone = phone;

this.email = email;

}

public void GetCard()

{

Console.WriteLine($"Name: {name}");

Console.WriteLine($"Occupation: {occupation}");

Console.WriteLine($"Age: {age}");

Console.WriteLine($"Home Phone: {phone.GetHomePhone()}");

Console.WriteLine($"Business Phone: {phone.GetBusinessPhone()}");

Console.WriteLine($"Cell Phone: {phone.GetCellPhone()}");

Console.WriteLine($"Email: {email}");

}

}

public class PhoneList

{

private string homePhone;

private string businessPhone;

private string cellPhone;

public PhoneList(string homePhone, string businessPhone, string cellPhone)

{

this.homePhone = homePhone;

this.businessPhone = businessPhone;

this.cellPhone = cellPhone;

}

}