

# 11. Classes and Objects

- 物件導向概念
- Java的物件導向設計
- Class定義
- Object實體化
- Constructors
- Access Modifiers

相信您一定聽過物件導向(Object-Oriented)這個名詞，但具體的意義又是什麼呢？基本上，我們可以將物件導向視為一種新的思維、新的想法，應用在系統分析、設計上，就產生了OOA(Object-Oriented Analysis物件導向分析)與OOD(Object-Oriented Design物件導向設計)；應用在程式語言，則有了OOP(Object-Oriented Programming物件導向程式設計)的誕生。目前C++、C#、Java、Object C等都是十分流行的物件導向程式語言。我們將先介紹物件導向的觀念與精神，再就Java語言本身的物件導向相關的程式語法與特性做一說明。

## 11.1 物件導向概念

□Programming is problem solving.□

傳統的程式設計概念，是將程式設計視為問題求解的過程：針對各種的應用問題，定義所需要的資料項目與處理問題的邏輯、流程與步驟、條件與限制，並撰寫程式碼以取得資料項目、進行資料處理，最後將問題的解答輸出。我們在進行設計程式以解決應用問題時，腦海中所思考的是：

要取得哪些資料？如何取得？資料的型態為何？要用什麼樣的資料結構來存放資料？資料的範圍為何？取得資料後該進行哪些資料處理？要使用什麼方法來處理？有沒有特定的處理流程、條件、步驟或限制？資料處理完成後，又要輸出哪些資訊給使用者？格式為何？要求為何？

我們可以說這是一種「以資料為核心」的思維方式。

至於現代的物件導向程式設計則是以「物件」為核心，將問題抽象化描述為一個由物件所構成的環境，透過物件的行為、屬性的改變、物件彼此間的互動、訊息的傳遞，來完成問題求解。我們所思考的往往是以下問題：

在應用問題所存在的真實或虛擬世界中，有哪些種類的物件存在？  
這些物件有哪些屬性？有哪些狀態？有哪些行為？  
物件與物件間有沒有關係？有沒有互動？問題的需求牽涉到那些物件？

這種以「物件為核心」的思維方式，就是物件導向程式設計的概念。

### 11.1.1 Objects

在Java的世界，萬事萬物都是「物件(objects)」，不論是真實或虛擬世界中有形或無形的事物都被視為物件：

對於供應鏈管理系統而言：特定的企業、產品、客戶、供應商、物流業者等是物件  
對於生產管理系統而言：特定的原料、生產線、作業員、品管員、生產計畫等是物件  
對於銀行存放款系統而言：特定的銀行、帳戶、貸款、信用卡等是物件  
對於學校的資訊系統而言：特定的學校、系所、班級、課程、學生等是物件  
對於線上遊戲的軟體而言：特定的女巫、怪獸、寶物、地圖、商店、玩家等是物件  
對於視窗系統系統而言：特定的程式視窗、對話盒、功能表、按鈕等是物件

與問題相關或有助於解決問題的物件加入到所建構的物件導向世界中，但和解決問題無關的物件屬性和物件行為會被省略。

例如學生成績管理系統當然會把學生視為物件，其屬性包含學生姓名、學號、成績等(但不包含學生的身高、體重等屬性)，其行為包含課程選修、加選、退選，成績登錄與查詢等(但不含社團活動、設備借用等行為)。

例如人事管理系統會把員工視為物件，其屬性包含了姓名、部門、職級、薪資等資料(但不包含員工的血型、星座等屬性)，其行為則包含了加班申請、出差申請或上班簽到等(但不含吃飯、睡覺等行為)。

物件所應該擁有的屬性與行為完全是依據系統的要求而決定的，在不同的軟體系統中也許會有同樣的物件，但其屬性和行為可能完全不同。

□Programming is problem solving□!

物件導向程式設計(object-oriented programming□OOP)將所要解決的問題，描繪成一個物件導向世界，凡是和問題相關或有助於解決問題的物件都存在於這個世界中。問題的處理邏輯、流程與步驟、條件與限制等需求，將會對映到物件的行為與屬性，透過物件彼此間的互動與訊息的傳遞，問題就可以被解決。

### 11.1.2 Classes

物件導向程式設計就是要建構一個虛擬的物件導向世界，在這個世界中存在著一些物件，透過這些物件的行為、屬性的改變、彼此的互動與訊息的傳遞，就可以完成特定的系統目的。問題是，在建構這個世界之前，您有沒有先定義這些物件的行為與屬性呢？

類別(class)就是物件的抽象定義，包含了屬性與行為的定義。

有了類別的定義後，我們才能在虛擬的物件導向世界中產生物件。

以紅龜粿的製作為例，必須先有紅龜粿的模板，才能大量生產出一模一樣的紅龜粿！



Fig. 1

再以整數變數的宣告□int x=5;□為例，變數x之所以能在程式中用來代表一個整數，是因為程式語言先行定義了int這個型態(包含它的數值範圍以及可進行的操作等)。

所以要在物件導向的程式中使用物件，您必須先行定義其所屬的類別，才能據以產生物件供程式使用。這

個產生物件的動作稱為「實體化(instantiate)[]而所產生出的物件則被稱為「實體(instance)[]

關於Java語言如何定義類別與產生物件，後續會再加以說明。

From:

<https://junwu.nptu.edu.tw/dokuwiki/> - Jun Wu的教學網頁

國立屏東大學資訊工程學系

CSIE, NPTU

Total: 252807

Permanent link:

<https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=java:classobject>

Last update: **2019/07/02 15:01**

