物件導向

什麼是物件導向？

物件導向程式(Object-oriented programming)

- 提高軟體的重用性、靈活性和擴充功能性。
- 程式中物件可以互相存取相關的資料

低耦合度

- 物件本身可以自由修改與變化，但不會影響其他物件
- 外部程式的修改也不會影響到物件本身的功能與運作

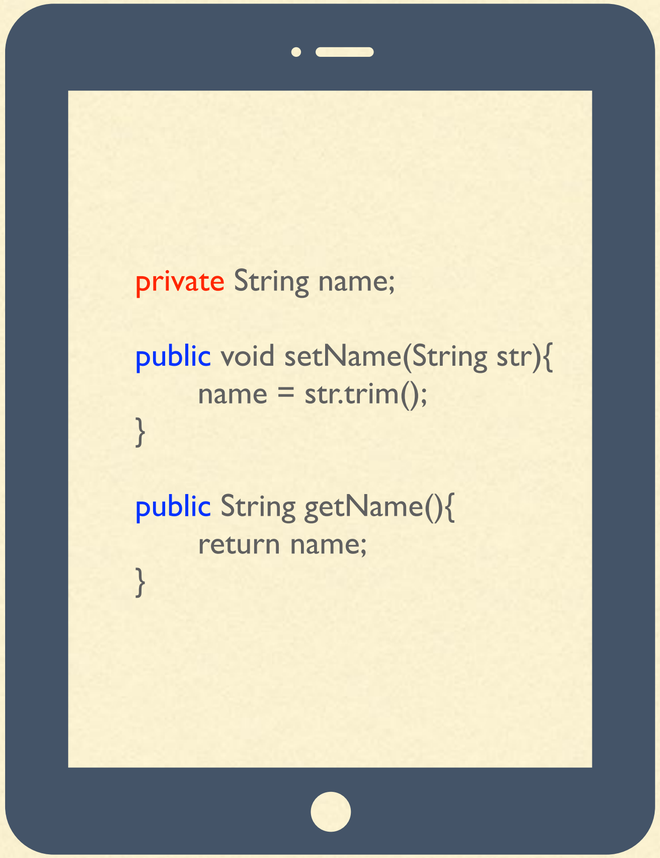
WORLD TRADE ORGANIZATION

物件與類別(Object and class)

- 類別是定義物件的一種型態，沒有實體
  - 屬性、方法
- 物件是某一個類別的實體(instance)

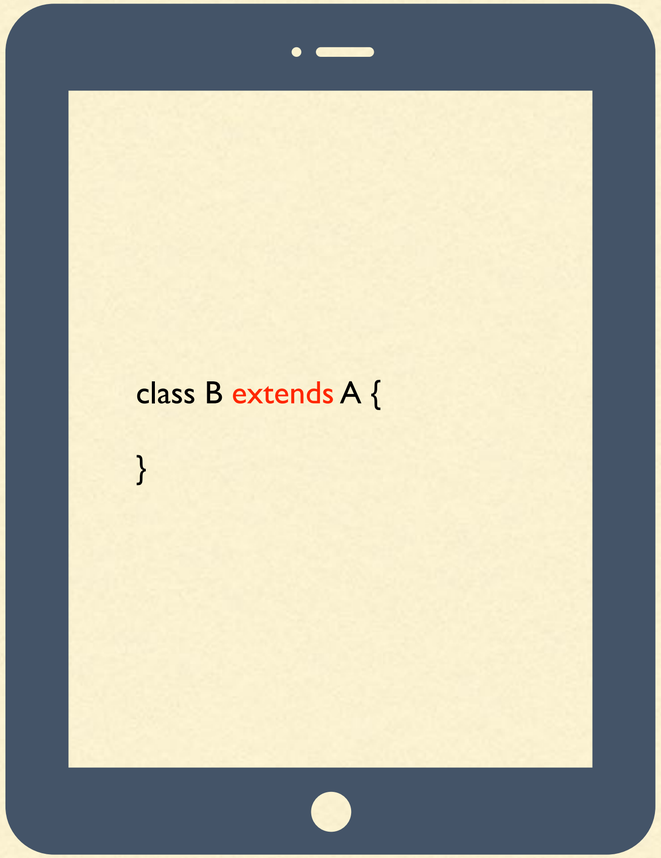
封裝(Encapsulation)

- 將實作與外部物件感興趣的操作介面分開
- 定義物件的操作介面和隱藏實作細節
- 使用private, public等修飾子控制



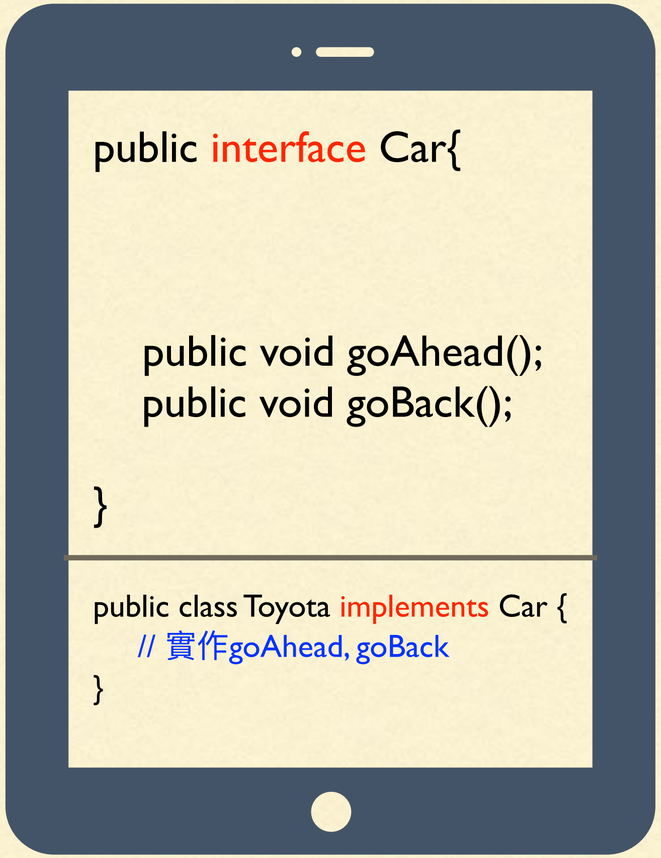
繼承(Extends)

- 基於某個父類別對物件的定義加以擴充
- 子類別可以增加新的方法，也可以覆寫父類別的方法



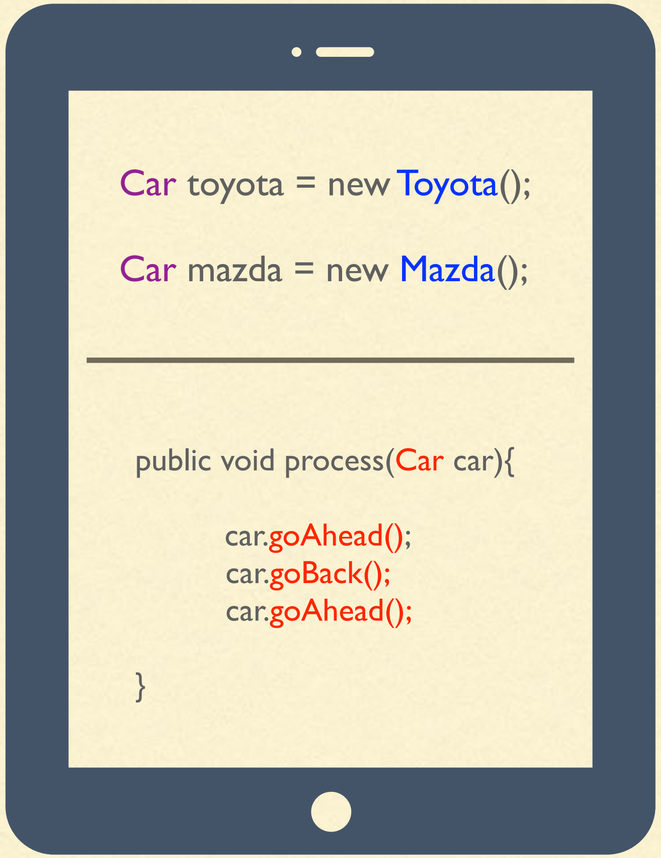
介面(Interface)

- 描述不同類別的共同行為
- 實作與介面分開，以便讓同一界面但不同的實作的物件能以一致的面貌讓外界存取
- interface, implements



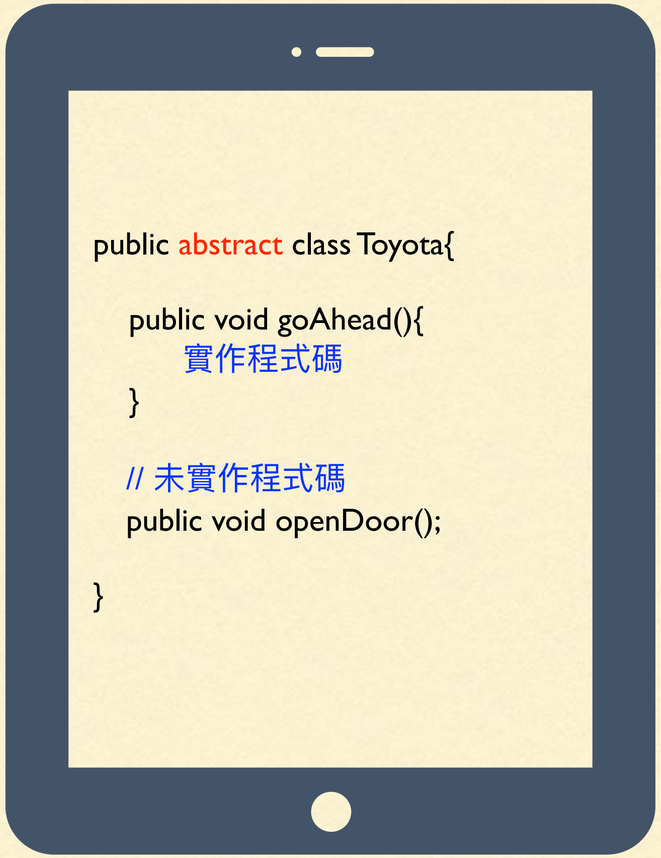
多型(Polymorphism)

- 同一個資料型態，操作不同物件實例
- 降低對資料型態操作的依賴層度
- 可用介面和父類別宣告物件型態



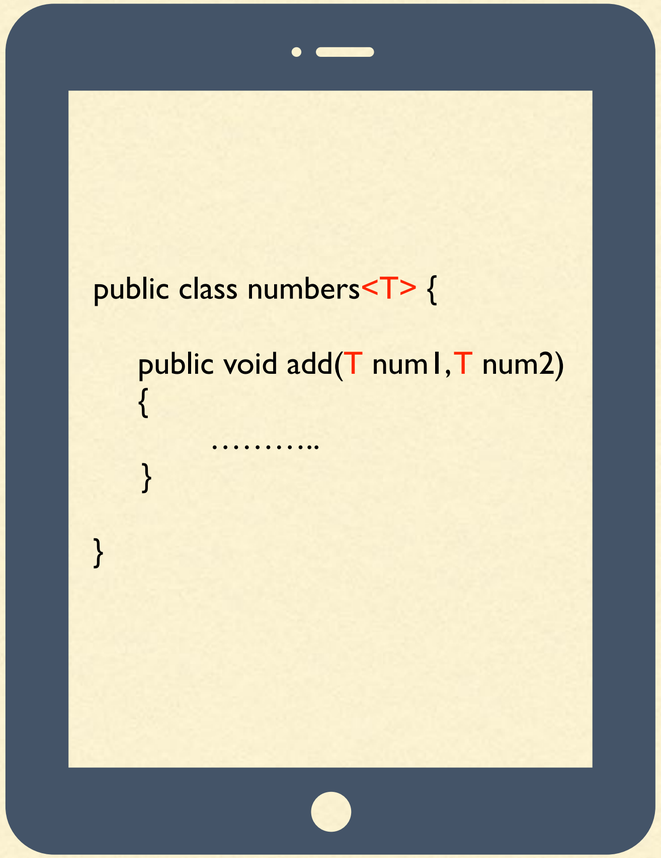
抽象類別(Abstract class)

- 包含抽象方法的類別，稱為抽象類別
- 子類別必須實作抽象類別的抽象方法
- 不可以被實體化



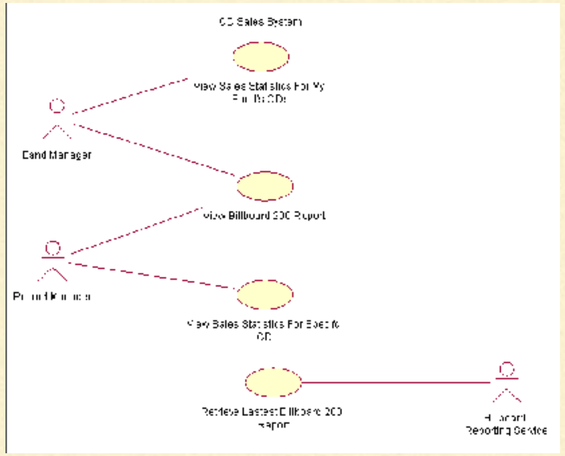
泛型(Generic)

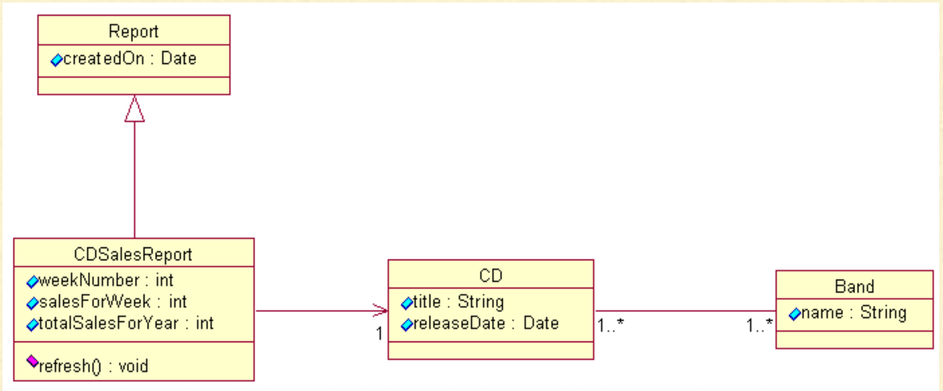
- 泛型是多型(polymorphism)的一種技巧
- 當編譯期間無法確定程式碼的撰寫方式，而是依照執行期間的狀況而決定
- 不用因為資料型別的限制而實作多種方法，只需要一種方法即可



UML(Unified Modeling Language)

- IT 技術人員能夠閱讀、傳達系統組織架構的設計規格
- 使用案例圖（Use-case diagram）
- 類別圖（Class diagram）
- 循序圖（Sequence diagram）
- 狀態圖（Statechart diagram）





Code Gym codegym.tech

設計模式(Design pattern)

- 設計模式是在解決問題中的經驗彙整
- 將反覆出現的各種問題提出解決的方案
- GoF提出了23種經典設計模式