

本地練習 (5 題)

本章為自學延伸，非課程考試範圍。 想多練物件導向的人再寫。每題寫成一個 `.java` (含 `main`)，用 `./run.sh 你的檔.java` 編譯執行；卡住再看 [參考解答/](#)。本練習聚焦 CH10：介面 `implements`、抽象類別與抽象方法、多型陣列、向上轉型與動態繫結、多重介面。檔名須與 `public class` 同名。

第 4 題 向上轉型 + 動態繫結

設計 `Animal` (方法 `String speak()` 回 `"...")`，子類別 `Dog` (回 `"汪")`、`Cat` (回 `"喵")` 覆寫之。在 `main` 用 `Animal a = new Dog();` 印一次、再 `a = new Cat();` 印一次，看同一變數依物件跑不同版本 (預期 `汪`、`喵`)。

```
1. `Animal` 含 `public String speak() { return "..."; }` 2. `Dog`、`Cat` 各 `@Override`  
public String speak()` 3. `Animal a = new Dog(); println(a.speak()); a = new Cat();`  
println(a.speak());`
```


易錯：以為宣告成 `Animal` 就跑 `Animal` 版 (動態繫結看物件、跑子類別版)。參考解答：參考解答/UpcastBinding.java

第 5 題 多重介面

定義 `Flyable { void fly(); }` 與 `Swimmable { void swim(); }`，讓 `Duck` 同時 `implements` 兩者（逗號分隔）。在 `main` 建一隻 `Duck` 並各呼叫 `fly()`、`swim()`（預期 `鴨子拍翅飛`、`鴨子划水游`）。

1. 兩個介面各一個方法
2. `class Duck implements Flyable, Swimmable`（逗號分隔，兩個都要 `@Override`）
3. `main` 內 `Duck d = new Duck(); d.fly(); d.swim();`

易錯：多重介面用 `extends` 接（類別實作介面用 `implements`）；只實作其中一個介面的方法（兩個都要）。參考解答：[參考解答/MultiInterfaceDuck.java](#)