# 比較整理

## 重載與重寫

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 重載 | | | 重寫 |
| 名稱 | overload | | | Override |
| 物件 | 方法 | | | 方法, 屬性 |
| 判別 | 相同 | 可同可不同 | 不同 | 方法名稱, 輸出值型態, 參數個數, 參數型態, 參數排序皆相同 |
| 方法名稱 | 輸出值型態 | 參數個數  參數型態  參數排序 |
| 範圍 | 同一個類別 | | | 子類別 |
| 權限 | 沒有權限控制 | | | 子類別覆寫的方法不能比父類別更高的權限 |
| static, private, final方法 | 可 | | | 不可 |
| 功能 | 增加程式可讀性 | | | 對父類別的方法實作或修改 |
| 進行時間 | compile | | | run |
| 速度 | 快 | | | 慢 |

## this和super

|  |  |  |
| --- | --- | --- |
|  | this | super |
| 呼叫屬性 | 呼叫本類別的屬性，找不到則向父類別尋找 | 呼叫父類別的屬性 |
| 呼叫方法 | 呼叫本類別的方法，找不到則向父類別尋找 | 呼叫父類別的方法 |
| 呼叫建構子 | this()只呼叫本類別建構子 | super()只呼叫父類別建構子 |
|  | 只在建構子第一行 | |
|  | 只能使用一次 | |
|  | 不能兩個一起使用 | |
|  | 都不使用時，預設為super() | |
| 物件 | 現在(正在呼叫該屬性、方法)的物件 |  |

## 抽象類別與介面

|  |  |  |
| --- | --- | --- |
| 比較 | **抽象類別** | **介面** |
| 關鍵字 | abstract class | interface |
| 組成 | 屬性, 全域常數, 構造方法, 普通方法, 抽象方法, static方法 | 全域常數, 抽象方法 |
| 權限(存取)訪問權限 | 各種權限皆可 | 只能使用public權限 |
| 使用 | 必須有子類別, 子類別繼承抽象類別 | 介面必須有子類別, 子類別實作介面 |
| 子類別宣告 | class 子類別 extends 抽象類別 | class 子類別 implements 介面A, 介面B |
| 關係 | 抽象類別實作多個介面 | 介面不能繼承抽象類別  介面可以繼承多個介面 |
| 設計模式 | 範本設計模式 | 工廠設計, 代理設計 |
| 兩者合用可以做轉接器 | |
| 繼承關係 | 子類別只能繼承一個父類別，單繼承 | 子類別可以實作多個介面  不受單一繼承限制 |
| 方法預設行為 | 可 | 不可 |
| 實作 | 如果子類別不是抽象類別，則必須提供抽象類別中所有聲明的方法的實作。 | 子類別需要提供介面所有聲明的方法的實作 |
| 建構子 | 抽象類別可以有建構子 | 介面不能有建構子 |
| 與正常Java類的區別 | 除了不能產生物件，抽象類別是一種類別 | 介面是完全不同的類型 |
| main方法 | 可以有main方法，且可以執行 | 沒有main方法 |
| 速度 | 快 | 慢 |
| 增添方法 | 抽象類別增添新方法，可以提供預設的實作。 | 介面增添方法，必須改變實作該介面的類別。 |
| 應用原則 | 屬於類別的本質不可或缺 | 非必要 |

權限由低到高：public, protected, default, private

**應用原則：與類別本質相關者，以抽象類別設計；類別外掛者，以介面設計為主。**

窗戶：打開：打開方法：左右拉, 上下拉, 直立軸旋轉, 水準軸旋轉

關閉：

防盜：

隔音：

杯子：空間

材質

保溫

密封

冷氣：開

關

溫控

風量

控制方式：面板控制、線控、遙控、智慧控制…

暖氣

除濕

人類：頭

手

腳

心

衣

褲

帽

眼鏡