

CH3

111360138 電子三甲 蔣安聖

讀書會

組員：

111360127 林煒哲

111360137 許鎧晏

111360138 蔣安聖

討論時間:2024/11/12 晚上 10 點

地點 Discord

詢問 GPT 問題



1. `this.num` 與不加 `this` 的 `num` 有什麼差別？



在 Java 中，`this.num` 和 `num` 在功能上是相同的。在大多數情況下，可以直接使用 `num`，而不需要特別加上 `this`。

差異說明

- `this.num`： `this` 關鍵字表示當前物件的實例，因此 `this.num` 明確表示屬於該物件的 `num` 屬性。在某些情況下（如變數名稱與參數名稱重疊時），`this` 可以用來避免混淆。
- `num`：若沒有與方法參數重疊的變數名，則直接使用 `num` 即可，Java 會自動將其視為該物件的屬性。

使用情境

1. 當參數名稱與屬性名稱相同時：例如在構造方法中，`this` 是必要的來區分屬性和參數變數。

java

複製程式碼

```
class Car {  
    int num;  
    Car(int num) {  
        this.num = num; // this.num 指的是屬性，右邊的 num 是參數  
    }  
}
```

2. 避免模糊性：在不確定是否有名稱重疊或需要明確指向當前實例時，可以用 `this` 提高可讀性。

在這段程式碼中，由於 `num` 和 `gas` 沒有與其他名稱重疊的情況，因此 `this.num` 和 `num` 在效果上是相同的。

2. Java 中可以宣告相同名稱且參數資料型態相同，只有參數數目不同的函式嗎？



可以的，Java 支援方法重載 (overloading)，允許在同一個類別中宣告多個相同名稱的方法，但它們的參數列表必須不同。參數列表不同的情況包括：

1. 參數數目不同。
2. 參數的資料型態不同。
3. 參數的排列順序不同（僅在資料型態不同時有效）。

範例

以下範例展示了透過不同的參數數目來進行重載：

```
java 複製程式碼

public class OverloadingExample {

    // 方法1：沒有參數
    void show() {
        System.out.println("沒有參數");
    }

    // 方法2：一個 int 參數
    void show(int num) {
        System.out.println("一個參數：" + num);
    }

    // 方法3：兩個 int 參數
    void show(int num1, int num2) {
        System.out.println("兩個參數：" + num1 + ", " + num2);
    }
}
```

在這裡，`show()` 方法被重載了三次：

- 無參數的 `show()`。
- 一個 `int` 參數的 `show(int num)`。
- 兩個 `int` 參數的 `show(int num1, int num2)`。

當呼叫這些方法時，Java 編譯器會根據傳入參數的數量來判斷應該使用哪個方法，因此可以確定它們之間沒有模糊性。

3. Java 中的類別庫與 C 語言中的函式庫一樣嗎？



Java 中的「類別庫」和 C 語言中的「函式庫」在本質上是相似的，但在結構和功能上有一些重要的不同：

1. 組織結構：

- Java 的類別庫包含許多類別、介面和方法，通常按封裝的概念組織在一起，並由不同的套件（package）來劃分。例如，`java.util` 包含集合框架，`java.io` 包含輸入/輸出操作。類別庫中的每個類別或介面都可以包含多個方法，並支援面向對象（OOP）特性。
- C 的函式庫則是由函式和變數組成的集合，以程式碼文件（如 `.c` 和 `.h` 檔案）形式存在，並依賴於頭文件來管理。它們主要是獨立的函式和變數的集合，通常不支援物件導向結構。

2. 資料封裝與可重用性：

- Java 類別庫支援物件導向設計，使用類別來封裝資料和方法，並且通過繼承、封裝、多型等特性增加了可重用性和靈活性。
- C 函式庫中函式和變數是直接的函式調用或變數存取。它們提供了基本的程式碼重用，但無法提供類別的封裝和繼承等特性。

3. 執行方式：

- Java 類別庫運行在 Java 虛擬機（JVM）上，並通過字節碼來執行。這些類別在 Java 程式執行期間可以動態載入。
- C 函式庫通常是靜態或動態鏈接庫（例如 `.lib` 或 `.dll` 文件）。它們會在編譯階段（靜態鏈接）或執行階段（動態鏈接）與 C 程式聯結。

4. 支援的功能：

- Java 類別庫包含豐富的工具包支援，例如 GUI、網路、檔案 I/O、集合框架等功能，使開發 Java 應用程式更加方便。
- C 函式庫通常提供更底層的函式，比如數學運算、標準輸入/輸出、字串處理等。它的函式庫對硬體和操作系統的抽象較少，通常需自行進行更底層的資源管理。

總結

Java 類別庫和 C 函式庫在概念上都是為了支援程式開發並提高重用性，但 Java 的類別庫提供了更多的面向對象支援、組織方式以及高層次的功能封裝，而 C 的函式庫則更加低層和直接。

心得

在這次的練習中，我們學到了 Java 語言最核心的東西，類別，在物件導向程式設計中，類別是一個很關鍵的工具，寫程式時往往會需要用到這個工具，不過我自己是學 C 的，因此對於物件導向的東西都不是很了解，若是撰寫結構體我都還可以掌握，不過類別對我來說就是全新的東西了，但上完課之後我覺得其實跟結構差不多，並不會很難，也都有邏輯，像是公私成員、多載與建構式，我都能從程式中看出邏輯，因此對我來說學習類別不會像是背函式庫名字那麼單調。

GitHub 程式連結與截圖

<https://github.com/DolphinBlast/JavaHW06>

