

基礎1

[Main.java](#)

[Ex01.java](#)

[Ex02.java](#)

[Main.java](#)

[Ex03.java](#)

[Main.java](#)

[bmi](#)

[Ex04.java](#)

[Main.java](#)

[自我挑戰:設計功能 計算總分\(\)、計算平均\(\)](#)

[自我挑戰:攝氏 換算 華氏](#)

[運算子](#)

[關係比較運算](#)

[邏輯運算](#)

[指定運算](#)

[if 條件 \(流程控制\)](#)

[亂數 Math.random\(\)](#)

[浮點數 轉換 整數](#)

[範例:](#)

[Ex06.java](#)

[自我挑戰:使用 亂數 產生英文與數學成績, 計算總分與平均, 最後顯示評語](#)

[自我挑戰:銀行帳戶](#)

[自我挑戰:猜密碼](#)

基礎2

[傳值 參數\(parameter\) 引數\(argument\) return 返回值](#)

[區域變數 作用域\(範圍\)](#)

[Scanner 使用者輸入資料](#)

[工具.java](#)

[自我挑戰:學生 使用 工具 計算總分與平均](#)

[自我挑戰:使用 傳值 返回值 設計帳戶存提款](#)

物件導向1

[物件類別](#)

[this 關鍵字](#)

[封裝 設值 取值](#)

[LocalDate.of\(\) 建立日期](#)

[建構子\(constructor\)](#)

[自我挑戰](#)

陣列

[靜態陣列 Array \(容量大小固定\)](#)

[動態陣列 ArrayList 容器 \(關鍵字搜尋 java api arraylist 查閱官方文件\)](#)

[尋找陣列裡物件](#)

[學生.java](#)

[班級.java](#)

[班級Test.java](#)

不可變物件 (immutable)

[String](#)

[StringBuilder](#)

[自我挑戰:不可變 Student](#)

物件導向2

[介面 interface](#)

[CharSequence 字串介面](#)

[自我挑戰: 使用 CharSequence 設計 Student 名字](#)

[MVC 設計模式](#)

[TextView.java](#)

[StudentModel.java](#)

[Controller.java](#)

[Test.java](#)

[物件導向3](#)

[繼承](#)

[View.java](#)

[TextView.java](#)

[Button.java](#)

[Test.java](#)

[ViewGroup.java](#)

[自我挑戰: 在 ViewGroup 設計 findViewById\(\)](#)

[自我挑戰: EditText \(能輸入文字的View\)](#)

[EditText.java](#)

[測試程式](#)

[序列化](#)

[Serializable 介面](#)

[學生.java](#)

[班級.java](#)

[AppData.java](#)

[Test存檔.java](#)

[Test讀檔.java](#)

[自我挑戰: 完成 銀行物件 序列化 \(銀行包含客戶、帳戶\)](#)

[映射容器 HashMap](#)

[自我挑戰: 設計 ExtraData 存放 App 額外資料](#)



殷志忠 Chih-Chung Yin
Google / Oracle 原廠認證講師
teach.yin@gmail.com



ORACLE
Certified Professional
Java SE 8 Programmer

ORACLE
Certified Expert
Java EE 6 Web
Component Developer

ORACLE
Certified Professional
Java EE 7-
Application Developer

基礎1

```
// Java 以類別為單位, 寫法 class XXX
// Main 通常 主要程式
// { } 代表程式範圍

// 木葉村.鳴人.影分身()
// 木葉村.鳴人.螺旋丸()

// 類別名稱
class Main {

    // 主功能 程式執行起點
    public static void main(String[] args) {

        System.out.println("主技能執行了");
        xxxx();
    }

    // 次功能
    public static void xxxx() {
        System.out.println("次技能執行了");
    }
}
```

Main.java

```
// 類別名稱
class Main {

    // 主功能 程式執行起點
    public static void main(String[] args) {

        Ex01.顯示訊息();

    }
}
```

Ex01.java

```
class Ex01 {  
  
    // 功能  
    public static void 顯示訊息() {  
  
        System.out.println("歡迎光臨");  
  
    }  
}
```

Ex02.java

```
class Ex02 {  
    public static void 計算() {  
        System.out.println("英文成績");  
        System.out.println(100);  
        System.out.println("數學成績");  
        System.out.println(99);  
        System.out.println("總分");  
        System.out.println(100+99);  
    }  
}
```

Main.java

```
class Main {  
  
    // 主功能 程式執行起點  
    public static void main(String[] args) {  
  
        //Ex01.顯示訊息();  
        Ex02.計算();  
    }  
  
}
```

Ex03.java

```
class Ex03 {  
    // 類別裡面由 資料 與 功能 組成  
  
    // 資料(變數)  
    //     String 文字類型(字串)  
    //     int     整數類型  
    //     double 浮點數類型(不精確的小數, 近似值)
```

```

public static String 名字;
public static int 英文成績;
public static int 數學成績;

// 功能
public static void 設定初值() {
    System.out.println("--- 設定初值 ---");
    名字 = "未命名";
    英文成績 = 0;
    數學成績 = 0;
}

public static void 顯示資料() {
    System.out.println("--- 顯示資料 ---");
    System.out.println(名字);
    System.out.println(英文成績);
    System.out.println(數學成績);
}

public static void 修改資料() {
    System.out.println("--- 修改資料 ---");
    名字 = "Tom";
    英文成績 = 100;
    數學成績 = 99;
}
}

```

Main.java

```

class Main {
    // 主功能 程式執行起點
    public static void main(String[] args) {
        Ex03.設定初值();
        Ex03.顯示資料();
        Ex03.修改資料();
        Ex03.顯示資料();
    }
}

```

bmi

<http://depart.femh.org.tw/dietary/3opd/bmi.htm>

Ex04.java

```

class Ex04 {

    // 類別裡面由 資料 與 功能 組成

    // 資料(變數)
    //     String 文字類型(字串)
    //     int    整數類型
    //     double 浮點數類型(不精確的小數, 近似值)

```

```

public static double cm;
public static double kg;
public static double m;
public static double bmi;

// 功能
public static void 設定初值() {
    System.out.println("--- 設定初值 ---");
    cm = 170.5;
    kg = 70.5;
    m = cm / 100.0;
    bmi = kg / (m * m);
}

public static void 顯示資料() {
    System.out.println("--- 顯示資料 ---");
    // 當 字串 + 數字 , 數字會自動加入字串 , 最終產生一個更長的字串
    System.out.println("cm: " + cm);
    System.out.println("kg: " + kg);
    System.out.println("m: " + m);
    System.out.println("bmi: " + bmi);
}
}

```

Main.java

```

class Main {
    // 主功能 程式執行起點
    public static void main(String[] args) {
        Ex04.設定初值();
        Ex04.顯示資料();
    }
}

```

自我挑戰: 設計功能 計算總分()、計算平均()

自我挑戰: 攝氏 換算 華氏

溫標換算公式

攝氏至華氏換算

$$[^{\circ}F] = [^{\circ}C] \times 9/5 + 32$$

運算子

https://www.w3schools.com/java/java_operators.asp

關係比較運算

- >
- >=
- <
- <=
- == 等於
- != 不等於

邏輯運算

- && 且
- || 或
- ! 非

指定運算

- =
- +=
- -=
- *=
- /=

if 條件 (流程控制)

https://www.w3schools.com/java/java_conditions.asp

```
public static void 評語1() {  
    System.out.println("--- 評語1 ---");  
  
    if(平均 >= 90.0) {  
        System.out.println("超棒");  
    }  
    if(平均 >= 80.0) {  
        System.out.println("很棒");  
    }  
    if(平均 >= 70.0) {  
        System.out.println("棒");  
    }  
    if(平均 < 70.0) {  
        System.out.println("不棒");  
    }  
}
```

```
public static void 評語2() {  
    System.out.println("--- 評語2 ---");  
  
    if(平均 >= 90.0) {  
        System.out.println("超棒");  
    }  
    else if(平均 >= 80.0) {  
        System.out.println("很棒");  
    }  
}
```

```

    }
    else if(平均 >= 70.0) {
        System.out.println("棒");
    }
    else {
        System.out.println("不棒");
    }
}

public static void 評語3() {

    System.out.println("--- 評語3 ---");

    if(bmi < 27) {
        System.out.println("過重");
    } else if(bmi < 24) {
        System.out.println("正常");
    } else if(bmi < 18.5) {
        System.out.println("過輕");
    } else {
        System.out.println("肥胖");
    }
}

public static void 評語4() {

    System.out.println("--- 評語4 ---");

    if(bmi < 18.5) {
        System.out.println("過輕");
    } else if(bmi < 24) {
        System.out.println("正常");
    } else if(bmi < 27) {
        System.out.println("過重");
    } else {
        System.out.println("肥胖");
    }
}

```

亂數 **Math.random()**

https://www.w3schools.com/java/java_math.asp

產生 0.0(含)~ 1.0(不含)之間的隨機浮點數

浮點數 轉換 整數

https://www.w3schools.com/java/java_type_casting.asp

語法：

(int) 浮點數

範例：

```

(int)(3.14)
(int)((100.0 + 99.0) / 2.0)

```


Ex06.java

```
class Ex06 {
    public static double r;
    public static int eng;

    public static void 設定初值() {
        r = Math.random(); // 0.0 <= Math.random() < 1.0
    }

    public static void 顯示結果() {
        System.out.println("亂數 " + r);
    }

    public static void 產生英文成績() {
        //eng = (int)(Math.random() * 100.0);
        eng = (int)(r * 101.0);
    }

    public static void 顯示英文成績() {
        System.out.println("英文成績 " + eng);
    }

    public static void 跑流程() {
        設定初值();
        顯示結果();
        產生英文成績();
        顯示英文成績();
    }
}
```

自我挑戰: 使用 亂數 產生英文與數學成績, 計算總分與平均, 最後顯示評語

自我挑戰: 銀行帳戶

- 變數
 - 帳戶名稱
 - 帳戶餘額
- 功能
 - 設定初始餘額 5000
 - 存款
 - 提款
 - 顯示帳戶資訊

自我挑戰: 猜密碼

- 變數
 - 密碼答案 (自行指定3位數密碼)
 - 電腦猜密碼 (亂數產生)
- 功能
 - 設定密碼答案
 - 電腦亂數猜密碼
 - 比對密碼是否猜中

基礎2

傳值 參數(parameter) 引數(argument) return 返回值

https://www.w3schools.com/java/java_methods_param.asp

執行功能時, 將資料傳遞給對方

區域變數 作用域(範圍)

https://www.w3schools.com/java/java_scope.asp

限功能內部使用, 功能執行結束後自動消失

Scanner 使用者輸入資料

https://www.w3schools.com/java/java_user_input.asp

工具.java

// 宣告使用 Scanner 掃描器

```
import java.util.Scanner;
```

```
public class 工具 {
```

```
    // 全域變數
```

```
    // private      私有的    不開放其他類別使用
```

```
    // System.in    代表鍵盤
```

```
    // new          建立物件
```

```
    private static Scanner sc = new Scanner(System.in);
```

```
    // int          整數      方法返回值 (結果) 若沒有返回值 以 void 表示
```

```
    // 輸入成績 方法名稱
```

```
    public static int 輸入成績() {
```

```
        System.out.println("請輸入成績: ");
```

```
        int x; // 區域變數, 限於功能內部使用
```

```
        x = sc.nextInt(); // 輸入整數
```

```
        return x; // return代表結束 x代表返回值
```

```
    }
```

```
    // 傳入訊息 為參數, 用來接收對方傳遞過來的值
```

```
    public static int 輸入成績v2(String 傳入訊息) {
```

```
        System.out.println(傳入訊息);
```

```
        int x; // 區域變數, 限於功能內部使用
```

```
        x = sc.nextInt(); // 輸入整數
```

```
        return x; // return代表結束 x代表返回值
```

```
    }
```

```
    public static int 加總(int 傳入整數1, int 傳入整數2) {
```

```
        int x; // 區域變數, 限於功能內部使用
```

```
        x = 傳入整數1 + 傳入整數2;
```

```
        return x; // return代表結束 x代表返回值
```

```
    }
```

```
    public static double 平均(int 傳入整數1, int 傳入整數2) {
```

```
        // 自我挑戰
```

```
    }
```

```
}
```

測試工具.java

```
/*
    殷志忠老師 (TeacherYin.com)
*/

public class 測試工具 {

    private static void 測試輸入() {
        System.out.println("第1次輸入成績 忽略 結果");
        // 呼叫 輸入成績() 忽略 返回結果
        工具.輸入成績();

        System.out.println("第2次輸入成績 接收 結果");
        // 呼叫 輸入成績() 接收 返回結果
        int 結果; // 區域變數, 限於功能內部使用
        結果 = 工具.輸入成績();
        System.out.println("收到結果: " + 結果);

        System.out.println("第3次輸入成績 傳值並接收結果");
        // 呼叫 輸入成績v2() 傳值並接收結果
        結果 = 工具.輸入成績v2("請輸入英文成績: ");
        System.out.println("收到結果: " + 結果);

        System.out.println("第4次輸入成績 傳值並接收結果");
        // 呼叫 輸入成績v2() 傳值並接收結果
        結果 = 工具.輸入成績v2("請輸入數學成績: ");
        System.out.println("收到結果: " + 結果);
    }

    private static void 測試加總() {
        System.out.println("第1次 測試加總 傳值並接收結果");
        int 結果;
        結果 = 工具.加總(100, 99);
        System.out.println("收到結果: " + 結果);
    }

    private static void 測試平均() {
        // 自我挑戰
    }

    public static void main(String[] args) {

        測試輸入();
        測試加總();
        測試平均();

    }
}
```

自我挑戰: 學生 使用 工具 計算總分與平均

```
/*
    殷志忠老師 (TeacherYin.com)
*/

public class 學生 {

    public static String 名字;
    public static int 英文;
    public static int 數學;

    public static void 顯示資料() {
        System.out.println(名字);
        System.out.println("英文:" + 英文);
        System.out.println("數學:" + 數學);
        System.out.println("總分:" + 工具.加總(英文, 數學));
        System.out.println("平均:" + 工具.平均(英文, 數學));
    }

}
```

自我挑戰: 使用 傳值 返回值 設計帳戶存提款

// 測試範例

```
boolean 結果;

結果 = 存款( 5000 );

if(結果) { // 簡化 if(結果 == true)
    System.out.println("存款 5000 成功");
} else {
    System.out.println("存款 5000 失敗");
}
```

物件導向1

物件類別

https://www.w3schools.com/java/java_classes.asp

- 物件類別 像是範本(規格), 專門設計給 物件 使用
- 給物件使用的 資料變數 或 方法, 不寫 **static**
- 物件透過 **new** 創造, 通常會指定變數來代表它

this 關鍵字

https://www.w3schools.com/java/ref_keyword_this.asp

- **this** 代表目前執行程式的物件對象

```

public class Student {

    // 物件變數(沒有static)
    public String name; // 預設值 null (空值)
    public int eng;     // 預設值 0
    public int math;    // 預設值 0

    // 物件方法(沒有static)
    public void show() {
        System.out.println("name " + this.name);
        System.out.println("eng " + this.eng);
        System.out.println("math " + this.math);
        System.out.println("tot " + 工具.加總(this.eng, this.math));
        System.out.println("avg " + 工具.平均(this.eng, this.math));
    }
}

```

測試程式

```

public class StudentTest {

    public static void main(String[] args) {

        // 宣告 物件變數
        Student a;
        Student b;

        // new 建立物件
        // = 設定給變數來代表它
        a = new Student();
        a.name = "Tom";
        a.eng = 100;
        a.math = 99;
        a.show(); // 執行 show 的物件對象是 a 即 this = a

        b = new Student();
        b.show(); // 執行 show 的物件對象是 b 即 this = b
    }
}

```

封裝 設值 取值

https://www.w3schools.com/java/java_encapsulation.asp

// 封裝類別:適度隱藏資訊, 避免向外界暴露資料, 導致設定不合理值

```

public class 日期 {

    // 利用 private 宣告為私有, 其他類別程式無法看到
    // = 0 可省略不寫 (整數預設為0)

    private int 年 = 0;
    private int 月 = 0; // 1 ~ 12
    private int 日 = 0; // 1 ~ 28,29,30,31 (目前簡化設計, 一律為30天)
}

```

// setter 指專門用來設定值的方法 如 set年() set月() set日()

```
public void set年(int 年) {
```

```
    // this 通常省略不寫
```

```
    // 當 參數(區域變數) 年 與 物件變數 年 同名時, 靠 this 來區別
```

```
    this.年 = 年;
```

```
}
```

```
public void set月(int 月) {
```

```
    if(月 < 1 || 月 > 12) {
```

```
        System.out.println("月份設定失敗: " + 月);
```

```
        System.out.println("[原因] 月份必須 1 ~ 12");
```

```
        return; // 結束 不返回值
```

```
    } else {
```

```
        this.月 = 月;
```

```
    }
```

```
    System.out.println("月份設定成功: " + 月);
```

```
}
```

```
public void set日(int 日) {
```

```
    // 自我挑戰
```

```
}
```

```
public void 顯示資料() {
```

```
    // && 且 || 或
```

```
    if (this.年 == 0 || this.月 == 0 || this.日 == 0) {
```

```
        System.out.println("日期無效 無法顯示");
```

```
    } else {
```

```
        System.out.println(this.年 + "/" + this.月 + "/" + this.日);
```

```
    }
```

```
}
```

```
}
```

測試程式

```
public class 測試日期 {
```

```
    public static void main(String[] args) {
```

```
        // 宣告 物件變數
```

```
        日期 d;
```

```
        // 建立物件(儲存格)
```

```
        d = new 日期();
```

```
        d.顯示資料(); // 日期無效
```

```
        // 設定資料 (年月日是 public 可能填入不合理值)
```

```
        // 年月日是 private 無法使用私有資料
```

```
        // d.年 = 2021;
```

```
        // d.月 = 16; 嘗試設定不合理值
```

```
        // d.日 = 25;
```

```

d.set年(2021);
d.set月(16); // 月份 16 不合理, 設定會失敗
d.set日(25);
d.顯示資料();

d.set月(6);
d.顯示資料();
}
}

```

LocalDate.of() 建立日期

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html#of-int-int-int->

建構子(constructor)

https://www.w3schools.com/java/java_constructors.asp

- 建構子就是類別當中的方法, 用於設定 物件資料(變數) 初值, 搭配 **new** 使用
- 如果沒有寫建構子, 會自動採用預設建構子, 雖然程式碼中看不到
- 建構子 方法名稱 必須與類別同名, 且不能寫 **void**

```

public class Date {
    private int year;
    private int month;
    private int day;

    // 建構子 (帶有3個參數 y m d)
    public Date(int year, int month, int day) {
        this.year = year;
        this.month = month;
        this.day = day;
        System.out.println("建構子 完成");
    }

    // 取值方法 getter
    public int getYear() { return this.year; }

    // 自我挑戰 getMonth()

    // 自我挑戰 getDay()

    // 字串方法, 通常用於 System.out.println() 顯示物件值, 規定命名為 toString(), 注意 S 為大寫
    public String toString() {
        return this.year + "/" + this.month + "/" + this.day;
    }
}

```

測試程式

```

public class DateTest {
    public static void main(String[] args) {

        // 建立物件 建構子完成初值設定
        Date d1 = new Date(2023,1,1);
        Date d2 = new Date(2023,2,2);
    }
}

```

```

// 測試 字串方法 toString()
System.out.println(d1.toString()); // .toString() 通常省略不寫
System.out.println(d2.toString()); // .toString() 通常省略不寫

// 測試 getter 取值
System.out.println("d2 年 = " + d2.getYear() );
System.out.println("d2 月 = " + d2.getMonth() );
System.out.println("d2 日 = " + d2.getDay() );
}
}

```

自我挑戰

- 建立2個帳戶物件
- 用 建構子 設定 帳戶名稱、餘額
- 用 toString() 輸出資料

陣列

靜態陣列 Array (容量大小固定)

https://www.w3schools.com/java/java_arrays.asp

動態陣列 ArrayList 容器 (關鍵字搜尋 java api arraylist 查閱官方文件)

https://www.w3schools.com/java/java_arraylist.asp

- 能自動調整容量(大小)
- 使用 add() 加入物件
- 使用 for-each 迴圈讀取所有物件

```
import java.util.ArrayList;
```

```
public class 班級 {
```

```
    public String 名稱;
    private ArrayList<學生> 學生們;
```

```
// 建構子
```

```
public 班級() {
    this.學生們 = new ArrayList();
    System.out.println("班級 建構完成");
}
```

```
public void 加入學生(學生 x) {
    this.學生們.add(x);
    System.out.println("加入學生 " + x.name);
}
```

```
public void 顯示資料() {
    System.out.println("----- " + this.名稱 + " 顯示資料 -----");
    for (學生 x : this.學生們) {
        x.show();
    }
}
```

```
}
```


測試程式

```
/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class 班級Test {

    學生 x1;
    學生 x2;
    學生 x3;
    班級 c1;
    班級 c2;

    x1 = new 學生();
    x1.name = "Tom";
    x1.eng = 100;
    x1.math = 99;

    x2 = new 學生();
    x2.name = "Amy";
    x2.eng = 90;
    x2.math = 95;

    x3 = new 學生();
    x3.name = "Jack";
    x3.eng = 80;
    x3.math = 85;

    c1 = new 班級();
    c1.名稱 = "android班";
    c1.加入學生(x1);
    c1.加入學生(x2);
    c1.顯示資料();

    c2 = new 班級();
    c2.名稱 = "python班";
    c2.加入學生(x1);
    c2.加入學生(x2);
    c2.加入學生(x3);
    c2.顯示資料();
}
}
```

尋找陣列裡物件

學生.java

```
public class 學生 {
    // 物件變數(沒有static)
    public int id;
    public String name;
    public int eng;
    public int math;
}
```

```

// 物件方法(沒有static)
public void show() {
    System.out.println("id " + this.id);
    System.out.println("name " + this.name);
    System.out.println("eng " + this.eng);
    System.out.println("math " + this.math);
    System.out.println("-----");
}
}

```

班級.java

```

import java.util.ArrayList;

public class 班級 {

    public String 名稱;
    private ArrayList<學生> 學生們;

    // 建構子
    public 班級() {
        this.學生們 = new ArrayList();
        System.out.println("班級 建構完成");
    }

    public void 加入學生(學生 x) {
        this.學生們.add(x);
        System.out.println("加入學生 " + x.name);
    }

    public void 顯示資料() {
        System.out.println("----- " + this.名稱 + " 顯示資料 -----");
        for (學生 x : 學生們) {
            x.show();
        }
    }

    public 學生 find學生ById(int id) {
        for (學生 x : 學生們) {
            if(x.id == id) {
                return x;
            }
        }
        return null; // 找無學生, 回傳 null (空值)
    }
}

```

班級Test.java

```
public class 班級Test {  
  
    public static void main(String[] args) {  
  
        學生 x1, x2;  
        班級 c;  
  
        x1 = new 學生();  
        x1.id = 1;  
        x1.name = "Tom";  
        x1.eng = 100;  
        x1.math = 99;  
  
        x2 = new 學生();  
        x2.id = 2;  
        x2.name = "Amy";  
        x2.eng = 95;  
        x2.math = 96;  
  
        c = new 班級();  
        c.名稱 = "甲班";  
        c.加入學生(x1);  
        c.加入學生(x2);  
        c.顯示資料();  
  
        // 測試 找學生  
        學生 x;  
        x = c.find學生ById(1);  
        // x = c.find學生ById(3);  
  
        if (x == null) {  
            System.out.println("找不到");  
        } else {  
            System.out.println("找到 " + x.name);  
        }  
    }  
}
```

不可變物件 (immutable)

[維基百科: 不可變物件](#)

物件被創造之後，它的狀態不可改變。

String

<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

StringBuilder

<https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html>

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class 不可變字串 {

    public static void main(String[] args) {
        // String 不可變字串 (可共用)
        String s1 = "Tom";
        String s2 = "Tom";
        String s3 = new String("Tom");
        // s1 == s2 若為相等, 表示兩個變數參考同一份資料(共用)
        System.out.println("s1 == s2 is " + (s1 == s2)); // true
        System.out.println("s1 == s3 is " + (s1 == s3)); // false

        // StringBuilder 可變字串
        StringBuilder sb;
        sb = new StringBuilder();
        // append 加入字串
        sb.append("name: ");
        sb.append(s1);
        sb.append(" eng: ");
        sb.append(100);

        // 可變字串 轉 不可變字串
        String s = sb.toString();
        System.out.println(s);

        // 可變字串簡化寫法 +
        String s4;
        s4 = "name: " + s1 + " eng: " + 100;
        System.out.println(s4);
    }
}

```

自我挑戰: 不可變 Student

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */

public class Student {
    private String name;
    private int eng;
    private int math;

    // constructor
    public Student(String name, int eng, int math) {
        this.name = name;
        this.eng = eng;
        this.math = math;
    }
}

```

```

// 不提供 setter

// getter
public String getName() {
    return name;
}

public int getEng() {
    return eng;
}

public int getMath() {
    return math;
}

public String toString() {
    return "name=" + name + ", eng=" + eng + ", math=" + math;
}
}

```

測試程式

```

public class StudentTest {
    public static void main(String[] args) {
        Student a;
        a = new Student("Tom", 100, 99);
        // 因為 Student 不可變, 只能讀取值, 無法修改值, 更安全的設計
        System.out.println(a.getName());
        System.out.println(a.getEng());
        System.out.println(a.getMath());
        System.out.println(a.toString());
    }
}

```

物件導向2

介面 interface

- 介面(合約)定義功能, 由類別實現(**implements**)介面要求的功能
- 用介面宣告變數, 能增加程式設計彈性, 因為介面是抽象型別(例如:USB介面、載具等都是抽象型別)
- 類別強調屬於什麼種類, 介面強調有什麼功能(雖然介面也可當作種類)

CharSequence 字串介面

<https://docs.oracle.com/javase/8/docs/api/java/lang/CharSequence.html>

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class 介面Test {

    public static void main(String[] args) {

        // String 與 StringBuilder 兩個類別都實現了 CharSequence 介面
        String s = "Amy";

        StringBuilder sb = new StringBuilder();
        sb.append("Jack");

        // CharSequence 字串介面 定義字串有哪些功能
        CharSequence cs;
        cs = s; // cs 參考(共用) s 字串
        System.out.println("長度(字數): " + cs.length());
        System.out.println("產生不可變字串: " + cs.toString());
        System.out.println("位置0文字: " + cs.charAt(0));

        cs = sb; // cs 參考(共用) sb 字串
        System.out.println("長度(字數): " + cs.length());
        System.out.println("產生不可變字串: " + cs.toString());
        System.out.println("位置0文字: " + cs.charAt(0));
    }
}

```

自我挑戰: 使用 **CharSequence** 設計 **Student** 名字

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class Student {

    private CharSequence name; // 可以是任何一種字串類型(可變或不可變)
    private int eng;
    private int math;

    // constructor
    public Student(CharSequence name, int eng, int math) {
        this.name = name;
        this.eng = eng;
        this.math = math;
    }

    // getter
    public CharSequence getName() {
        return name;
    }

    public int getEng() {
        return eng;
    }
}

```

```

public int getMath() {
    return math;
}

public String toString() {
    // + name 時 .toString() 可省略不寫
    return "name=" + name.toString() + ", eng=" + eng + ", math=" + math;
}
}

```

測試程式

```

public class StudentTest {

    public static void main(String[] args) {

        String s;
        StringBuilder sb;

        s = "Amy";
        sb = new StringBuilder("Jack");

        Student st1;
        st1 = new Student(s, 100, 99);

        Student st2;
        st2 = new Student(sb, 90, 95);

        System.out.println(st1.toString());
        System.out.println(st2.toString());
    }

}

```

MVC 設計模式

<https://zh.wikipedia.org/zh-tw/MVC>

MVC模式(Model-view-controller)是軟體工程中的一種軟體架構模式，把軟體系統分為三個基本部分：模型(Model)、視圖(View)和控制器(Controller)。

TextView.java

```

public class TextView {

    private CharSequence text;

    // getter
    public CharSequence getText() {
        return text;
    }
}

```

```

// setter
public void setText(CharSequence text) {
    this.text = text;
    this.show();
}

private void show() {
    System.out.println("----TextView----");
    System.out.println(this.text);
}
}

```

StudentModel.java

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class StudentModel {
    private String name;
    private int eng;
    private int math;

    public StudentModel(String name, int eng, int math) {
        this.name = name;
        this.eng = eng;
        this.math = math;
    }

    public String getName() {
        return name;
    }

    public int getEng() {
        return eng;
    }

    public int getMath() {
        return math;
    }
}

```

Controller.java

```

public class Controller {

    private StudentModel model;
    private TextView view;

    public void create() {
        model = new StudentModel("Tom", 100, 99);
        view = new TextView();
    }
}

```



```

public void run() {
    String s;
    s = model.getName() + "\n" + model.getEng() + "\n" + model.getMath();
    view.setText(s);
}
}

```

Test.java

```

public class Test {

    public static void main(String[] args) {
        Controller c;
        c = new Controller();
        c.create();
        c.run();
    }
}

```

物件導向3

繼承

[繼承\(維基百科\)](#)

繼承可以使得子類具有父類的各種屬性和方法，而不需要再次編寫相同的代碼。
 在令子類繼承父類的同時，可以重新定義某些資料，並重寫某些功能，即覆蓋父類原有資料和功能。

View.java

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class View extends Object { // 可省略繼承 Object, 預設自動繼承 Object

    private int id;
    private int width;
    private int height;

    public View(int id, int width, int height) {
        this.id = id;
        this.width = width;
        this.height = height;
    }

    public int getId() {
        return id;
    }

    public int getWidth() {
        return width;
    }
}

```

```

    public int getHeight() {
        return height;
    }

    public void show() {
        System.out.println("----" + this.getClass().getSimpleName() + "----");
        System.out.println("id: " + id + " width: " + width + " height: " + height);
    }
}

```

TextView.java

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class TextView extends View { // extends 繼承

    private CharSequence text;

    public TextView(int id, int width, int height) {
        super(id, width, height); // super() 呼叫父類別建構子
    }

    public CharSequence getText() {
        return text;
    }

    public void setText(CharSequence text) {
        this.text = text;
    }

    @Override // 覆寫 改寫父類別方法
    public void show() {
        super.show(); // super 呼叫父類別
        if(text == null) { // 若為空值
            System.out.println("");
        } else {
            System.out.println("text: " + text);
        }
    }
}

```

Button.java

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class Button extends TextView {

    public Button(int id, int width, int height) {
        super(id, width, height);
    }
}

```

```

public void click() {
    System.out.println("----click----");
    System.out.println(this.getText() + " 被按下了");
}
}

```

Test.java

```

public class Test {

    public static void main(String[] args) {

        TextView tv1 = new TextView(1, 50, 20);
        tv1.setText("訊息");

        TextView tv2 = new TextView(2, 50, 20);
        tv2.setText("請按OK");

        Button btn = new Button(3, 50, 20);
        btn.setText("OK");

        tv1.show();
        tv2.show();
        btn.show();

        btn.click();
    }
}

```

ViewGroup.java

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class ViewGroup extends View {
    private ArrayList<View> views;

    public ViewGroup(int id, int width, int height) {
        super(id, width, height);
        this.views = new ArrayList();
    }

    public void addView(View v) {
        this.views.add(v);
    }

    @Override // 覆寫 改寫父類別方法
    public void show() {
        for (View v : views) {
            v.show();
        }
    }
}

```

自我挑戰：在 ViewGroup 設計 findViewById()

測試程式

```
/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */

public class Test3 {

    public static void main(String[] args) {
        TextView tv1;
        TextView tv2;
        Button btn;
        ViewGroup vg;

        tv1 = new TextView(1, 50, 20);
        tv1.setText("訊息");

        tv2 = new TextView(2, 50, 20);
        tv2.setText("請按OK");

        btn = new Button(3, 50, 20);
        btn.setText("OK");

        vg = new ViewGroup(4, 50, 60);
        vg.addView(tv1);
        vg.addView(tv2);
        vg.addView(btn);
        vg.show();

        // 尋找按鈕
        Button b;
        b = (Button) vg.findViewById(3); // (Button) 將返回值 View 轉 Button
        b.click();
    }
}
```

自我挑戰：EditText (能輸入文字的View)

EditText.java

```
import java.util.Scanner;

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
```

```

public class EditText extends TextView {

    private Scanner sc;

    public EditText(int id, int width, int height) {
        super(id, width, height);
        sc = new Scanner(System.in);
    }

    public void input() {
        System.out.print("請輸入資料: ");
        String s;
        s = sc.next();    // 輸入字串資料
        this.setText(s);  // setText() 從 TextView 繼承
    }
}

```

測試程式

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class Test4 {

    public static void main(String[] args) {

        TextView tv_cm;    // 文字訊息 如:身高(cm)
        EditText et_cm;    // 輸入文字資料 如: 170
        TextView tv_result; // 顯示結果訊息 如:您身高為 1.7 公尺

        // 文字訊息
        tv_cm = new TextView(1, 50, 20);
        tv_cm.setText("身高(cm)");
        tv_cm.show();
        // 輸入文字資料
        et_cm = new EditText(2, 50, 20);
        et_cm.show();
        et_cm.input();

        // 取得輸入文字
        String s_cm;
        s_cm = et_cm.getText().toString();

        // 字串 轉 整數
        int i_cm;
        i_cm = Integer.parseInt(s_cm);

        // 公分換算公尺
        double d_m;
        d_m = i_cm / 100.0;

        // 顯示結果訊息
        String s_result;
        s_result = "您身高為 " + d_m + " 公尺";
    }
}

```

```

        tv_result= new TextView(3, 50, 20);
        tv_result.setText( s_result );
        tv_result.show();
    }
}

```

序列化

[序列化 \(維基百科\)](#)

序列化(serialization)在**計算機科學**的資料處理中, 是指將**資料結構**或**物件**狀態轉換成可取用格式(例如存成檔案, 存於緩衝, 或經由網路中傳送), 以留待後續在相同或另一台計算機環境中, 能恢復原先狀態的過程。

Serializable 介面

<https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html>

Serializable介面沒有要求實現任何功能(空介面), 作用像是類別的標籤, 讓系統能辨識類別有介面。

學生.java

```

import java.io.Serializable;
/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class 學生 implements Serializable { // Serializable 資料可序列化
    public String name;
    public int eng;
    public int math;

    public String getName() {
        return name;
    }
    public int getEng() {
        return eng;
    }
    public int getMath() {
        return math;
    }
    @Override
    public String toString() {
        return "name=" + name + ", eng=" + eng + ", math=" + math;
    }
}

```

班級.java

```

import java.io.Serializable;
import java.util.ArrayList;

```

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class 班級 implements Serializable { // Serializable 資料可序列化
    public String 名稱;
    public ArrayList<學生> 學生們;

    // 建構子
    public 班級() {
        this.學生們 = new ArrayList();
        System.out.println("班級 建構完成");
    }

    public void 加入學生(學生 x) {
        this.學生們.add(x);
        System.out.println("加入學生 " + x.name);
    }

    public void 顯示資料() {
        System.out.println("----- " + this.名稱 + " 顯示資料 -----");
        for (學生 x : 學生們) {
            System.out.println(x.toString());
        }
    }
}

```

AppData.java

```

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class AppData {

    private 班級 c;
    private final String filename; // final 建構子初值後禁止修改

    public AppData() {
        filename = "app.data"; // 沒指定路徑, 預設為專案資料夾
    }

    public void 存檔() throws Exception {
        FileOutputStream fos = new FileOutputStream(filename);
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(this.c); // this.c 須先透過 set班級() 完成設定
        oos.close();
    }
}

```

```

        fos.close();
    }

    public void 讀檔() throws Exception {
        FileInputStream fis = new FileInputStream(filename);
        ObjectInputStream ois = new ObjectInputStream(fis);
        this.c = (班級) ois.readObject(); // (班級) 將讀到的 Object 轉 班級
        ois.close();
        fis.close();
    }

    // setter and getter
    public void set班級(班級 c) {
        this.c = c;
    }

    public 班級 get班級() {
        return c;
    }
}

```

Test存檔.java

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class Test存檔 {
    public static void main(String[] args) {
        學生 x1;
        學生 x2;
        班級 c1;

        x1 = new 學生();
        x1.name = "Tom";
        x1.eng = 100;
        x1.math = 99;

        x2 = new 學生();
        x2.name = "Amy";
        x2.eng = 90;
        x2.math = 95;

        c1 = new 班級();
        c1.名稱 = "android班";
        c1.加入學生(x1);
        c1.加入學生(x2);
        c1.顯示資料();

        AppData appData;
        appData = new AppData();
        appData.set班級(c1);
    }
}

```



```

    try {
        appData.存檔();
        System.out.println("存檔成功");
    } catch(Exception e) {
        System.out.println("存檔失敗");
        System.out.println(e.toString());
        // e.printStackTrace(); // 完整錯誤訊息
    }
}
}
}

```

Test讀檔.java

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class Test讀檔 {

    public static void main(String[] args) {
        班級 c1;
        AppData appData;
        appData = new AppData();
        try {
            appData.讀檔();
            System.out.println("讀檔成功");
            c1 = appData.get班級();
            c1.顯示資料();
        } catch(Exception e) {
            System.out.println("讀檔失敗");
            System.out.println(e.toString());
            // e.printStackTrace(); // 完整錯誤訊息
        }
    }
}

```

自我挑戰: 完成 銀行物件 序列化 (銀行包含客戶、帳戶)

映射容器 HashMap

- https://www.w3schools.com/java/java_hashmap.asp
- <https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html>

```

import java.util.ArrayList;
import java.util.HashMap;

```

```

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */
public class TestMap {

```

```

public static void main(String[] args) {

    // key        字串類型
    // value      任意物件類型
    String key;
    Object value;

    // Map 映射容器 (透過 key 管理 value)
    HashMap<String, Object> map;
    map = new HashMap();

    // 放資料 put()
    key = "Tom";
    value = new Student("Tom", 100, 99);
    map.put(key, value);

    key = "100";
    value = "中正區";
    map.put(key, value);

    key = "list";
    value = new ArrayList(); // 即 ArrayList<Object>
    map.put(key, value);

    key = "map2";
    value = new HashMap(); // 即 HashMap<Object, Object>
    map.put(key, value);

    // 取資料 get()
    //      可能需要資料轉型, 當得到的資料與變數類型不同時)
    //      例如 HashMap 的 value 宣告為 Object(抽象類型), 可能是任何東西
    Student st;
    st = (Student) map.get("Tom");

    String area;
    area = (String) map.get("100");

    ArrayList<Object> list;
    list = (ArrayList<Object>) map.get("list");

    HashMap<Object, Object> map2;
    map2 = (HashMap<Object, Object>) map.get("map2");
}
}

```

自我挑戰: 設計 **ExtraData** 存放 App 額外資料

```

import java.util.HashMap;

/**
 *
 * 殷志忠老師 (TeacherYin.com)
 */

public class ExtraData {

```

```

private HashMap<String, Object> map;

public ExtraData() {
    map = new HashMap();
}

public void putExtra(String name, int value) {
    map.put(name, value);
}

public int getIntExtra(String name) {
    return (int) map.get(name);
}

public void putExtra(String name, CharSequence value) {
    map.put(name, value);
}

public CharSequence getCharSequenceExtra(String name) {
    return (CharSequence) map.get(name);
}

public void putExtra(String name, Object value) {
    map.put(name, value);
}

public Object getObjectExtra(String name) {
    return map.get(name);
}
}

```

測試程式

```

import java.util.ArrayList;

/**
 * 殷志忠老師 (TeacherYin.com)
 */
public class TestExtra {
    public static void main(String[] args) {

        ArrayList<Student> list;
        list = new ArrayList();
        list.add(new Student("Jack", 80, 85));
        list.add(new Student("Amy", 70, 75));

        ExtraData data;
        data = new ExtraData();

        // 寄資料
        data.putExtra("成績", 100);
        data.putExtra("名字", "Tom");
        data.putExtra("陣列", list);
    }
}

```

```
// 取資料
int 成績 = data.getIntExtra("成績");
CharSequence 名字 = data.getCharSequenceExtra("名字");
ArrayList<Student> 陣列 = (ArrayList<Student>) data.getObjectExtra("陣列");

System.out.println(成績);
System.out.println(名字);
for (Student x : 陣列) {
    System.out.println(x.toString());
}
}
```