

Java 程式設計進階

static 關鍵字

鄭安翔

ansel_cheng@hotmail.com

課程大綱

- 1) 類別成員 **static** 關鍵字
 - 類別屬性 **static attribute**
 - 類別方法 **static method**
- 2) 常用 **static** 元件
- 3) 類別進階

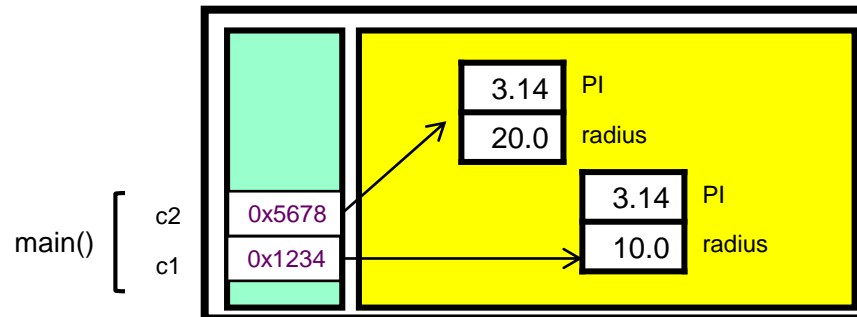
實體成員

- 實體成員 non-static (instance) member
 - 實體屬性：每個物件各自擁有一份資料
 - 實體方法：需透過特定物件來操作
 - 物件實體化後,物件屬性才存在,物件方法才可使用

Circle 範例

```
01 public class Circle {
02     public double PI = 3.14;
03     public double radius;
04     public Circle(double r) {
05         radius = r;
06     }
07     public double getArea() {
08         return PI * radius * radius;
09     }
10     public void draw() {
11         .....
12     }
13 }
```

```
01 public class CircleTest {
02     public static void main(String[] args) {
03         Circle c1 = new Circle(10);
04         System.out.println("半徑10的圓,面積為"+c1.getArea());
05         c1.draw();
06         Circle c2 = new Circle(20);
07         System.out.println("半徑20的圓,面積為"+c2.getArea());
08         c2.draw();
09     }
10 }
11 }
```



只使用實體成員的不足

- **Circle** 類別只使用實體屬性及方法的不足
 - 每個 **Circle** 物件都有相同屬性 **PI**
 - 記憶體空間浪費
 - 維護不易
 - 建立 **Circle** 物件才能計算面積
 - 記憶體空間浪費
 - `Circle.getArea(double radius)` ?

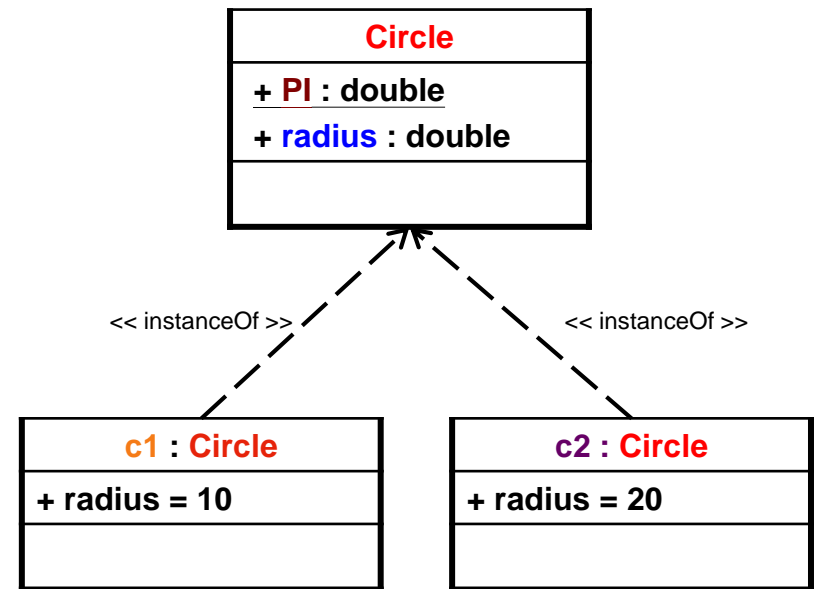
類別成員

■ 類別成員 **static (class) member**

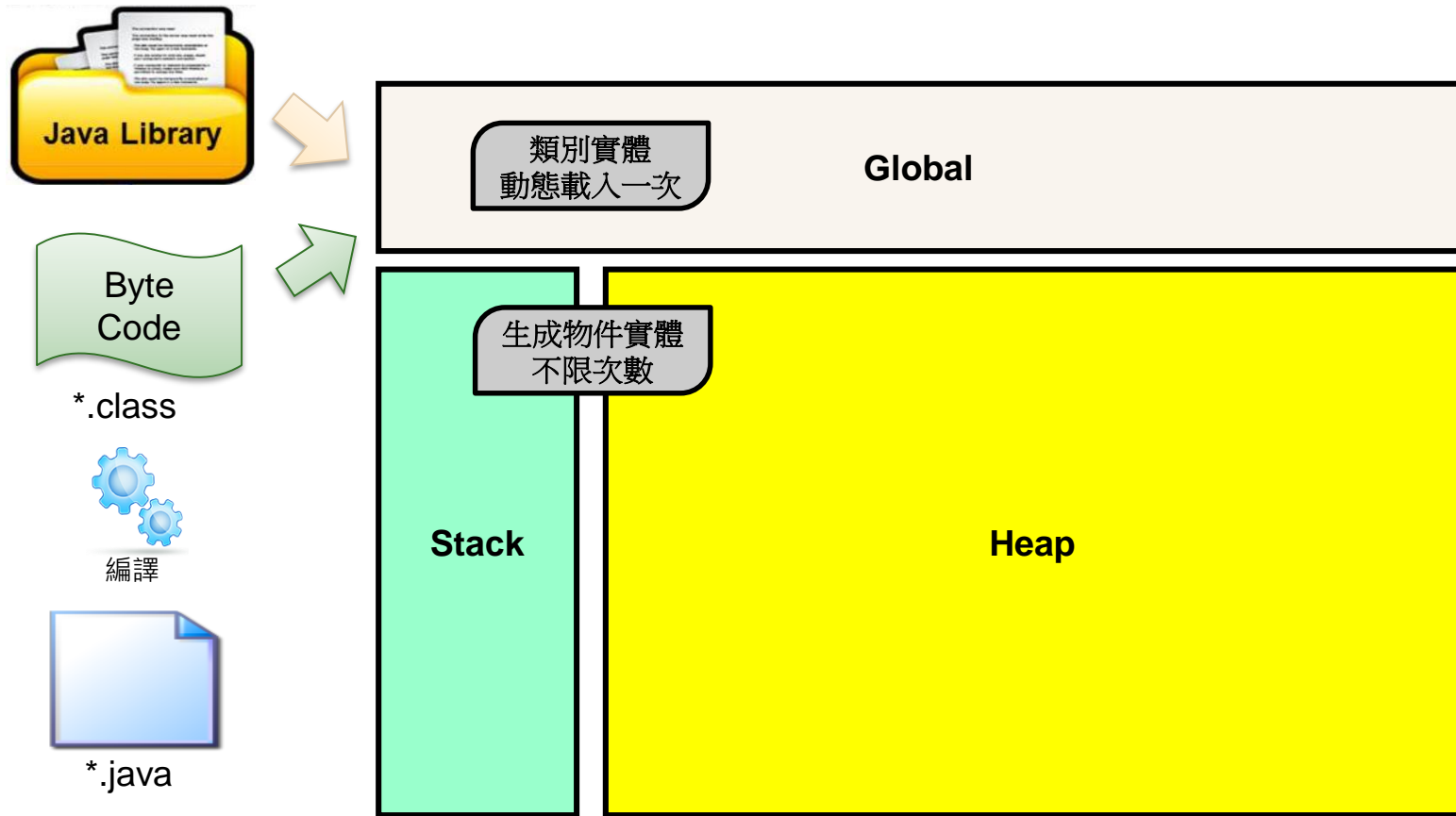
- 類別屬性：不伴隨物件,由同類別所有物件共享
- 類別方法：不需特定物件,可由類別來操作
- **Java**用 **static** 修飾字來宣告類別屬性及類別方法

類別屬性

- 用來表示同一類別所有物件共用的屬性欄位
 - 類似全域變數 Global Variable
- 語法
 - [modifiers] **static** type name
- 可用兩種方法存取
 - **ClassName.attribute;**
 - **objectName.attribute;**



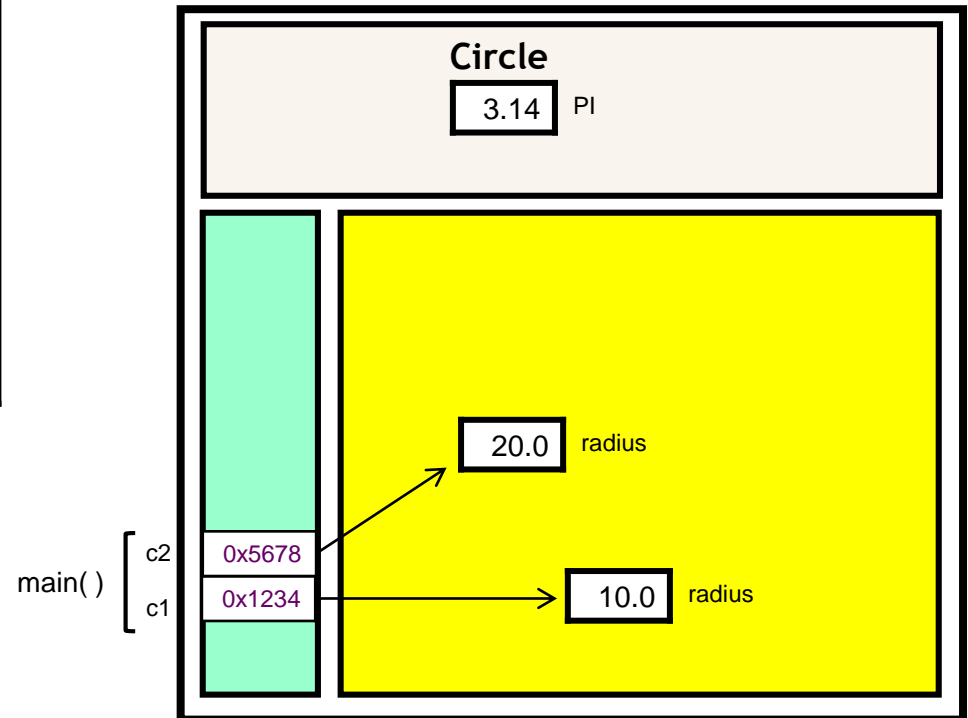
Java 記憶體配置



類別屬性 static attribute

```
01 public class Circle {  
02     public static double PI = 3.14;  
03     public double radius;  
04     public Circle(double r) {  
05         radius = r;  
06     }  
07     public double getArea() {  
08         return PI * radius * radius;  
09     }  
10     public void draw() {  
11         .....  
12     }  
13 }
```

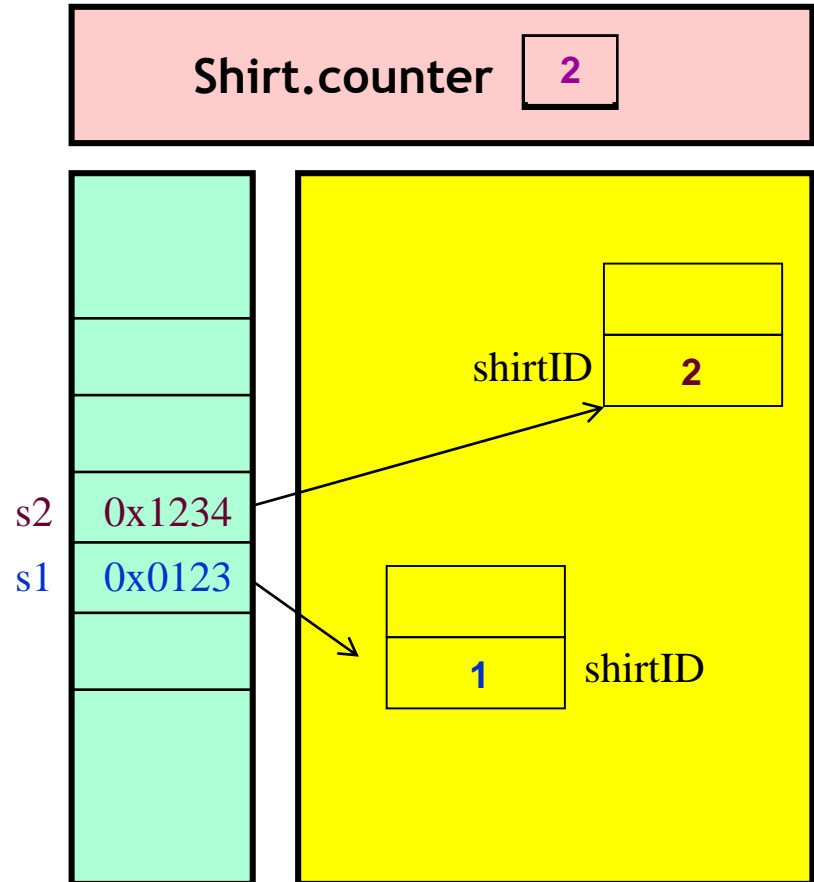
```
01 public class CircleTest {  
02     public static void main(String[] args) {  
03         Circle c1 = new Circle(10);  
04         Circle c2 = new Circle(20);  
05         System.out.print("圓周率:"+Circle.PI);  
06     }  
07 }  
08 }
```



類別屬性 static attribute

```
01 public class Shirt {  
02     public static int counter = 0;  
03     public int ShirtID;  
04  
05     public Shirt() {  
06         counter++;  
07         shirtID = counter;  
08     }  
09 }
```

```
01 public class TestShirt {  
02     public static void main(String [] args) {  
03         Shirt s1 = new Shirt();  
04         Shirt s2 = new Shirt();  
05     }  
06 }
```



類別方法 static method

■ 用途

- ❑ 在類別/物件不存在前,即可呼叫
- ❑ 存取static variable
- ❑ 提供utility 工具方法：執行結果只與傳入參數有關

■ 語法

[modifiers] static return_type name([argu_list]) { }

■ 用法

- ❑ ClassName.method();
- ❑ objectName.method();

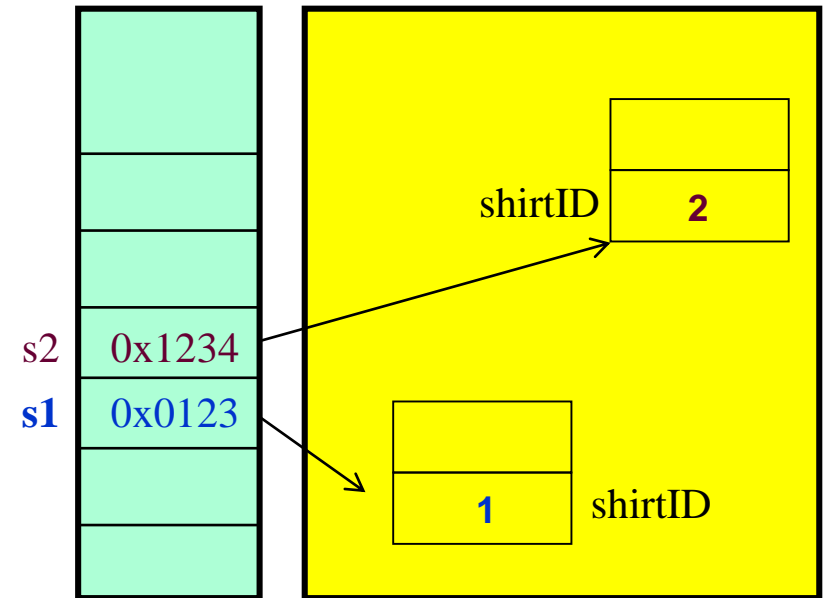
類別方法 static method

```
01 public class Shirt {
02     public static int counter = 0;
03     public int shirtID;
04
05     public Shirt() {
06         counter++;
07         shirtID = counter;
08     }
09
10     public static int getTotalCount() {
11         return counter;
12     }
13 }
```

```
01 public class TestShirt {
02     public static void main(String [] args) {
03         System.out.println("number of Shirt is "
04             + Shirt.getTotalCount());
05         Shirt s1 = new Shirt();
06         System.out.println("number of Shirt is "
07             + s1.getTotalCount());
08         Shirt s2 = new Shirt();
09         System.out.println("number of Shirt is "
10             + s2.getTotalCount());
11     }
12 }
```

Shirt.counter

2



A screenshot of a Windows command prompt window titled "系統管理員: 命令提示字元". The window shows the following commands and output:

```
c:\JavaClass>javac TestShirt.java

c:\JavaClass>java TestShirt
number of Shirt is 0
number of Shirt is 1
number of Shirt is 2

c:\JavaClass>
```

類別方法 static method

```
01 public class Shirt {
02     public static int counter = 0;
03     public int shirtID;
04
05     public Shirt() {
06         counter++;
07         shirtID = counter;
08     }
09
10     public static int getTotalCount() {
11         return counter;
12     }
13
14     public static String convertShirtSize( int numericalSize) {
15         if(numericalSize < 10){
16             return "S" ;
17         } else if(numericalSize < 14){
18             return "M" ;
19         } else if(numericalSize < 18){
20             return "L" ;
21         } else {
22             return "XL" ;
23         }
24     }
25
26 }
```

```
01 public class TestShirt {
02     public static void main(String [] args) {
03         .....
04         System.out.println("歐洲尺碼14號請買" +
05                             Shirt.convertShirtSize(14);
06     }
07 }
08 }
```

課程大綱

- 1) 類別成員 **static** 關鍵字
- 2) 常用 **static** 元件
 - **System**類別
 - **Math**類別
- 3) 類別進階

常用 static 元件 - System 類別

欄位摘要

static PrintStream	err	“標準” 錯誤輸出串流。
static InputStream	in	“標準” 輸入串流。
static PrintStream	out	“標準” 輸出串流。

方法摘要

static void	arraycopy (Object src, int srcPos, Object dest, int destPos, int length)	從指定源陣列中複製一個陣列，複製從指定的位置開始，到目標陣列的指定位置結束。
static String	clearProperty (String key)	移除指定鍵指示的系統屬性。
static Console	console ()	返回與當前 Java 虛擬機器關聯的唯一 Console 物件（如果有）。
static long	currentTimeMillis ()	返回以毫秒為單位的當前時間。
static void	exit (int status)	終止當前正在運行的 Java 虛擬機器。
static void	gc ()	運行垃圾回收器。
static Map < String , String >	getenv ()	返回一個不能修改的當前系統環境的字元串映射視圖。
static String	getenv (String name)	獲取指定的環境變數值。
static Properties	getProperties ()	確定當前的系統屬性。
static String	getProperty (String key)	獲取指定鍵指示的系統屬性。
static String	getProperty (String key, String def)	獲取用指定鍵描述的系統屬性。
static SecurityManager	getSecurityManager ()	獲取系統安全介面。
static int	identityHashCode (Object x)	返回給定物件的雜湊碼，該程式碼與預設的方法 hashCode () 返回的程式碼一樣，無論給定物件的類別是否覆寫 hashCode ()。

常用 static 元件 - System 類別

<code>static Channel</code>	<code>inheritedChannel()</code> 返回從創建此 Java 虛擬機器的實體中繼承的通道。
<code>static void</code>	<code>load(String filename)</code> 從作為動態資源庫的本地檔案系統中以指定的檔案名載入程式碼檔案。
<code>static void</code>	<code>loadLibrary(String libname)</code> 載入由 libname 參數指定的系統資源庫。
<code>static String</code>	<code>mapLibraryName(String libname)</code> 將一個資源庫名稱映射到特定於平臺的、表示本機資源庫的字元串中。
<code>static long</code>	<code>nanoTime()</code> 返回最準確的可用系統計時器的當前值，以毫微秒為單位。
<code>static void</code>	<code>runFinalization()</code> 運行處於掛起終止狀態的所有物件的終止方法。
<code>static void</code>	<code>runFinalizersOnExit(boolean value)</code> 已過時。 該方法具有固有的不安全性。它可能對正在使用的物件調用終結方法，而其他執行緒同時正在操作這些物件，從而導致不正確的行為或死鎖。
<code>static void</code>	<code>setErr(PrintStream err)</code> 重新分派“標準”錯誤輸出串流。
<code>static void</code>	<code>setIn(InputStream in)</code> 重新分派“標準”輸入串流。
<code>static void</code>	<code>setOut(PrintStream out)</code> 重新分派“標準”輸出串流。
<code>static void</code>	<code>setProperties(Properties props)</code> 將系統屬性設置為 Properties 參數。
<code>static String</code>	<code>setProperty(String key, String value)</code> 設置指定鍵指示的系統屬性。
<code>static void</code>	<code>setSecurityManager(SecurityManager s)</code> 設置系統安全性。

常見工具方法 Math Utilities

欄位摘要

static double	E	比任何其他值都更接近 e （即自然對數的底數）的 double 值。
static double	Pi	比任何其他值都更接近 π （即圓的周長與直徑之比）的 double 值。

方法摘要

static double	abs (double a)	返回 double 值的絕對值。
static float	abs (float a)	返回 float 值的絕對值。
static int	abs (int a)	返回 int 值的絕對值。
static long	abs (long a)	返回 long 值的絕對值。
static double	acos (double a)	返回一個值的反餘弦；返回的角度範圍在 0.0 到 π 之間。
static double	asin (double a)	返回一個值的反正弦；返回的角度範圍在 $-\pi/2$ 到 $\pi/2$ 之間。
static double	atan (double a)	返回一個值的反正切；返回的角度範圍在 $-\pi/2$ 到 $\pi/2$ 之間。
static double	atan2 (double y, double x)	將矩形坐標 (x, y) 轉換成極坐標 (r, theta)，返回所得角 theta。
static double	cbrt (double a)	返回 double 值的立方根。
static double	ceil (double a)	返回最小的（最接近負無窮大）double 值，該值大於等於參數，並等於某個整數。

static double	cos (double a)	返回角的三角餘弦。
static double	cosh (double x)	返回 double 值的雙曲線餘弦。
static double	exp (double a)	返回歐拉數 e 的 double 次方的值。
static double	expm1 (double x)	返回 $e^x - 1$ 。
static double	floor (double a)	返回最大的（最接近正無窮大）double 值，該值小於等於參數，並等於某個整數。
static int	getExponent (double d)	返回 double 表示形式中使用的無偏指數。
static int	getExponent (float f)	返回 float 表示形式中使用的無偏指數。
static double	hypot (double x, double y)	返回 $\sqrt{x^2 + y^2}$ ，沒有中間溢位或下溢。
static double	IEEEremainder (double f1, double f2)	按照 IEEE 754 標準的規定，對兩個參數進行餘數運算。
static double	log (double a)	返回 double 值的自然對數（底數是 e ）。
static double	log10 (double a)	返回 double 值的底數為 10 的對數。

常見工具方法 Math Utilities

static double	max (double a, double b) 返回兩個 double 值中較大的一個。
static float	max (float a, float b) 返回兩個 float 值中較大的一個。
static int	max (int a, int b) 返回兩個 int 值中較大的一個。
static long	max (long a, long b) 返回兩個 long 值中較大的一個。
static double	min (double a, double b) 返回兩個 double 值中較小的一個。
static float	min (float a, float b) 返回兩個 float 值中較小的一個。
static int	min (int a, int b) 返回兩個 int 值中較小的一個。
static long	min (long a, long b) 返回兩個 long 值中較小的一個。
static double	nextAfter (double start, double direction) 返回第一個參數和第二個參數之間與第一個參數相鄰的浮點數。
static float	nextAfter (float start, double direction) 返回第一個參數和第二個參數之間與第一個參數相鄰的浮點數。
static double	nextUp (double d) 返回 d 和正無窮大之間與 d 相鄰的浮點值。
static float	nextUp (float f) 返回 f 和正無窮大之間與 f 相鄰的浮點值。
static double	pow (double a, double b) 返回第一個參數的第二個參數次方的值。
static double	random () 返回帶正號的 double 值，該值大於等於 0.0 且小於 1.0。
static double	rint (double a) 返回最接近參數並等於某一整數的 double 值。
static long	round (double a) 返回最接近參數的 long。
static int	round (float a) 返回最接近參數的 int。

static double	scalb (double d, int scaleFactor) 返回 $d \times 2^{\text{scaleFactor}}$ ，其捨入方式如同將一個正確捨入的浮點值乘以 double 值集合中的一個值。
static float	scalb (float f, int scaleFactor) 返回 $f \times 2^{\text{scaleFactor}}$ ，其捨入方式如同將一個正確捨入的浮點值乘以 float 值集合中的一個值。
static double	signum (double d) 返回參數的符號函數；如果參數為 0，則返回 0；如果參數大於 0，則返回 1.0；如果參數小於 0，則返回 -1.0。
static float	signum (float f) 返回參數的符號函數；如果參數為 0，則返回 0；如果參數大於 0，則返回 1.0；如果參數小於 0，則返回 -1.0。
static double	sin (double a) 返回角的三角正弦。
static double	sinh (double x) 返回 double 值的雙曲線正弦。
static double	sqrt (double a) 返回正確捨入的 double 值的正平方根。
static double	tan (double a) 返回角的三角正切。
static double	tanh (double x) 返回 double 值的雙曲線餘弦。
static double	toDegrees (double angrad) 將用弧度表示的角轉換為近似相等的用角度表示的角。
static double	toRadians (double angdeg) 將用角度表示的角轉換為近似相等的用弧度表示的角。
static double	ulp (double d) 返回參數的 ulp 大小。
static float	ulp (float f) 返回參數的 ulp 大小。

課程大綱

- 1) 類別成員 **static** 關鍵字
- 2) 常用 **static** 元件
- 3) 類別進階
 - 類別方法存取規則
 - 靜態區塊

實體成員 vs. 類別成員

- 實體成員可以存取類別成員
 - 物件實體可存取類別共享屬性
 - 實體方法可以呼叫類別方法

```
01 public class CircleTest {  
02     public static void main(String[] args) {  
03         System.out.println("圓周率:" + Circle.PI);  
04         System.out.println("圓周率:" + c1.PI);  
05         System.out.println("半徑10的圓,面積為"+  
06             Circle.areaFormula(10));  
07         Circle c1 = new Circle(10);  
08         System.out.println("c1面積為" + c1.getArea());  
09     }  
10 }  
11 }
```

```
01 public class Circle {  
02     public static double PI = 3.14;  
03     public double radius;  
04     public Circle(double r) {  
05         radius = r;  
06     }  
07     public static double areaFormula(double r){  
08         return PI * r * r;  
09     }  
10     public double getArea() {  
11         return areaFormula(radius);  
12     }  
13     public void draw() {  
14         .....  
15     }  
16 }
```

實體方法可以
呼叫類別方法

類別方法 static method

■ 存取規則

□ 類別方法內只可以使用

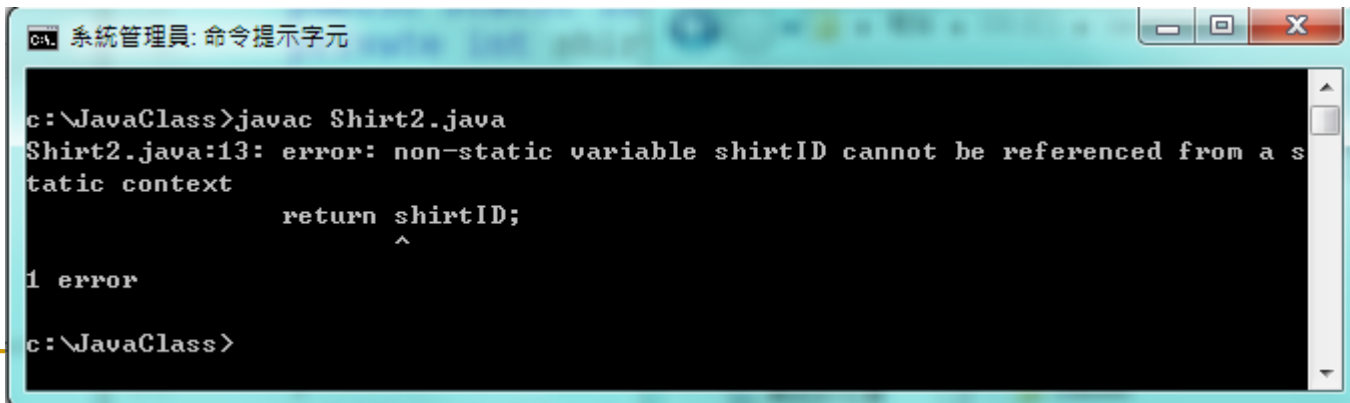
- 區域變數 local variables
- 方法傳入參數 parameter
- 類別屬性 static variables
- 其他類別方法 static method

□ 類別方法內不可以存取非靜態 (實體)成員

- ~~■ 物件屬性 instance variable~~
- ~~■ 物件方法 instance method~~
- ~~■ this~~

類別方法 static method

```
01 public class Shirt2 {  
02     public static int counter = 0;  
03     private int shirtID;  
04  
05     public Shirt() {  
06         counter++;  
07         shirtID = counter;  
08     }  
09     public static int getTotalCount(){  
10         return counter;  
11     }  
12     public static int getShirtID(){  
13         return shirtID;  
14     }  
15 }
```



系統管理員: 命令提示字元

```
c:\JavaClass>javac Shirt2.java  
Shirt2.java:13: error: non-static variable shirtID cannot be referenced from a s  
tatic context  
        return shirtID;  
                ^  
1 error  
  
c:\JavaClass>
```

靜態區塊 static block

- 靜態區塊通常用來初始類別變數的內容
- 定義在類別內,方法之外,前面加**static**修飾的程式區塊{ }
- 只在類別初次載入實執行一次

```
01 public class Shirt3 {  
02     public static int counter;  
03     static {  
04         counter = Integer.getInteger("ShirtCounter").intValue();  
05     }  
06 }
```

```
01 public class TestStaticInit {  
02     public static void main(String[] args) {  
03         System.out.println("Shirt Counter = "+ Shirt3.counter);  
04     }  
05 }
```

java -DShirtCounter=101 TestStaticInit



```
C:\JavaClass>java -DShirtCounter=101 TestStaticInit  
Shirt counter = 101  
C:\JavaClass>
```