

第九章

PHP物件導向設計-繼承與多型

繼承



- 繼承是將一類別所有成員與方法拷貝至其他類別的技術，不需要重複定義相同的成員與方法，提高程式碼的再利用性。
- 我們可將已有的基礎類別叫做父類別或基本類別，新的類別叫做子類別或衍生類別。

繼承特性

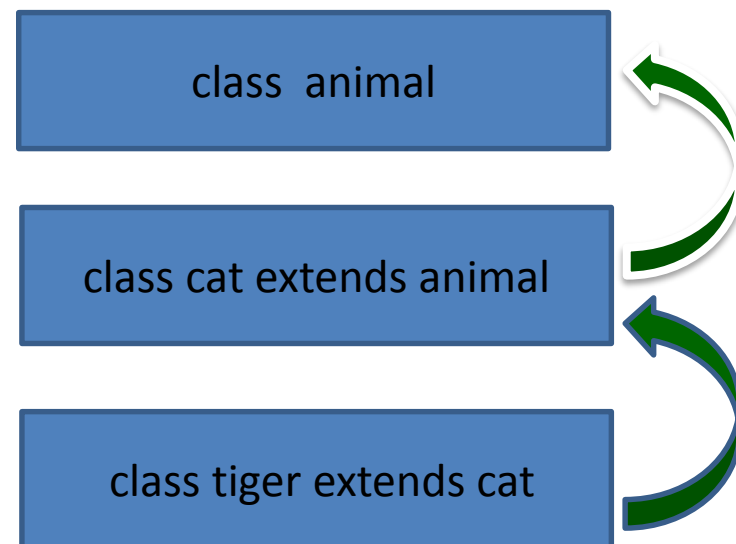
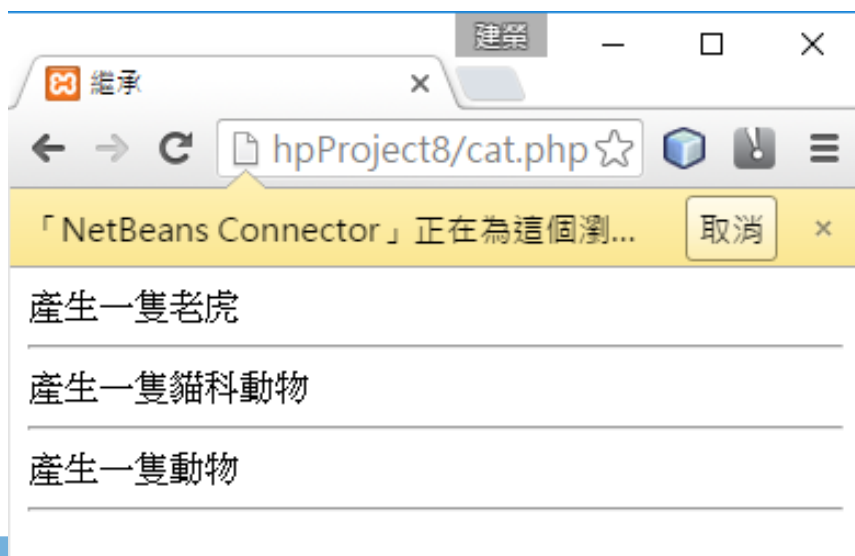


- PHP內允許一個子類別只能從一個父類別中繼承資料，PHP不支援多重繼承，建立一個子類別時可使用extends繼承一個父類別。
- 子類別可繼承父類別內public與protected權限成員與方法，private權限成員與方法是不能被繼承的。

繼承特性



- 老虎具有貓科動物趾端具銳利而彎曲的爪，及以伏擊的方式獵捕其它動物的特性，而貓科動物具有動物會動的生物特性。所以我們可讓老虎繼承於貓科，而貓科繼承於動物，設計三個類別，而這三個類別內均有建構子。練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat.php」）。



繼承特性

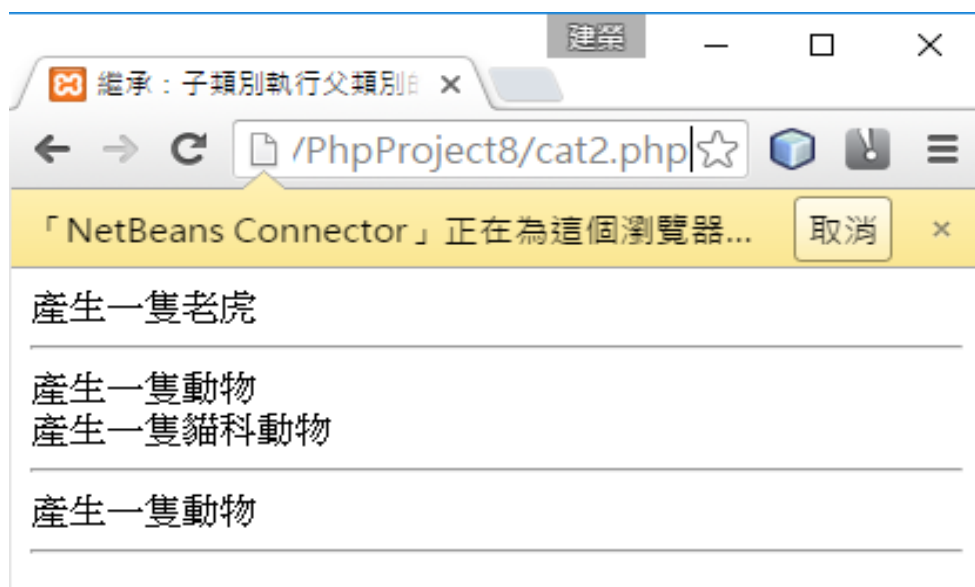


- 第07行開始建立cat類別，繼承animal類別，第09行建立cat類別的建構子；第12行開始建立tiger類別，繼承cat類別，第14行建立tiger類別的建構子；第17行建立animal類別，第19行建立animal類別的建構子。
- PHP產生物件時不會自動執行父類別內的建構子與解構子：
 - 如果要執行父類別的建構子，請用「parent::__construct();」方式呼叫
 - 如果要執行父類別的解構子，請用「parent::__destruct();」方式呼叫

繼承特性



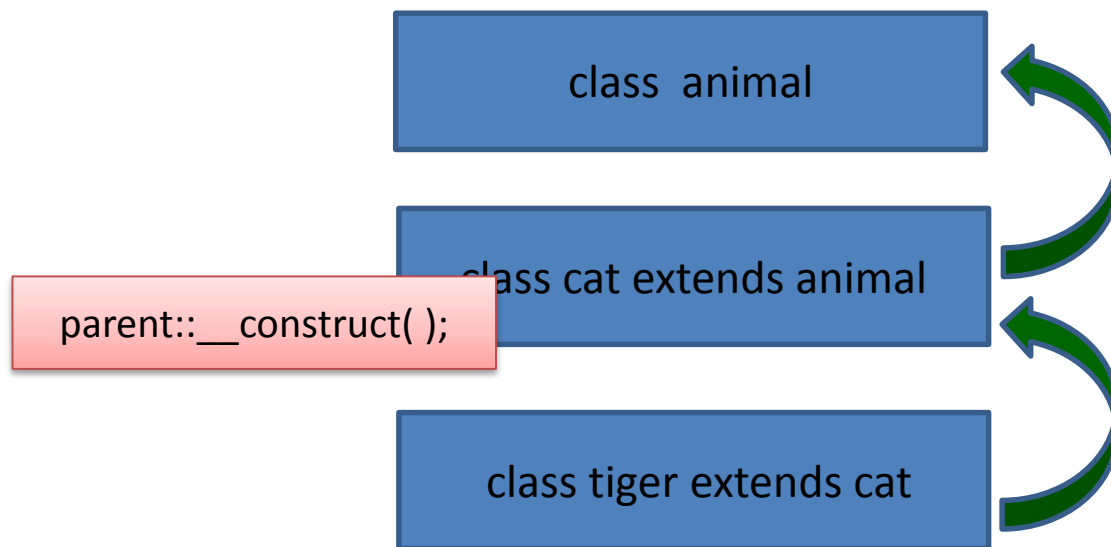
- 若cat類別內呼叫animal類別建構子，執行時會如何呈現呢？請將cat.php另存新檔為cat2.php，我們修改cat這個類別內的語法，來了解會有甚麼變化，在此僅列出cat與tiger的語法，其他語法請參考cat.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat2.php」）。



繼承特性



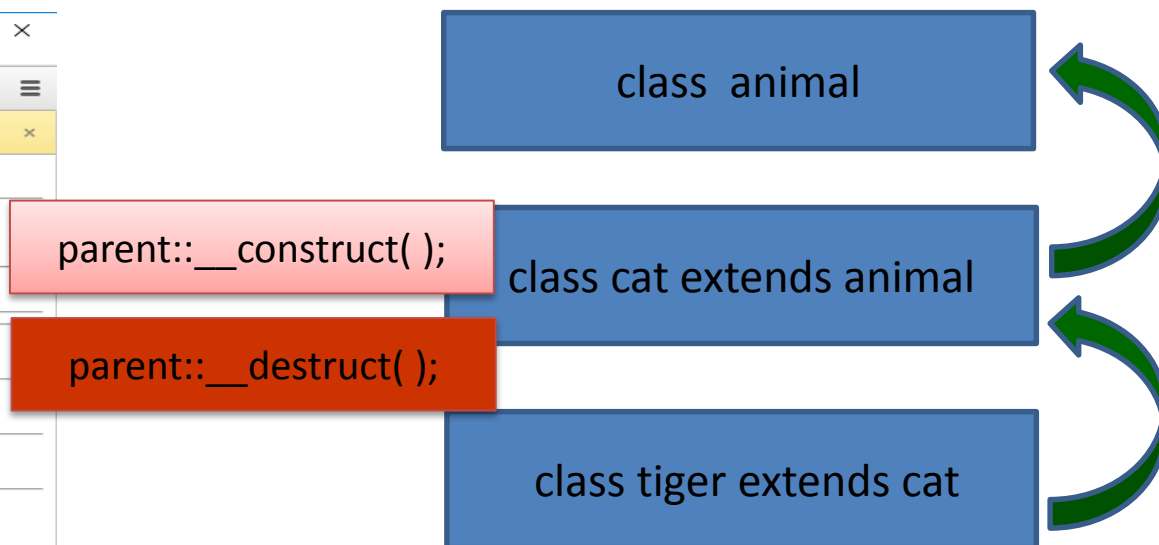
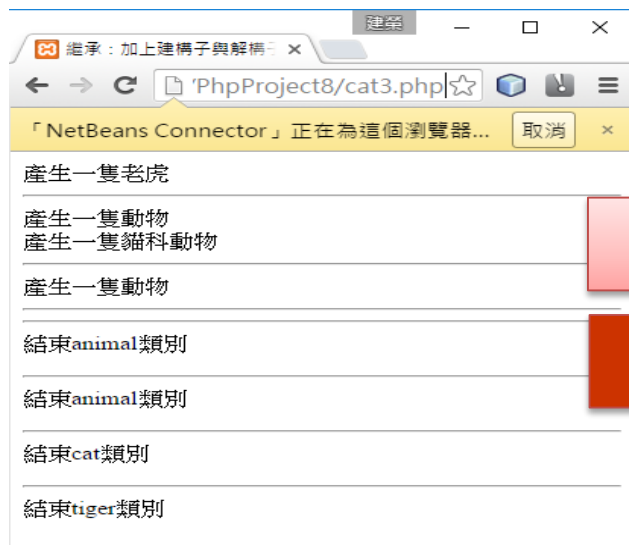
- 由於第09行內呼叫父類別的建構子，所以cat類別建構子執行前會先執行父類別的建構子，再執行自己的建構子。
- 請留意tiger是cat的子類別，不過並沒有執行cat父類別的建構子，代表說PHP語法不會自動執行父類別的建構子語法。



繼承特性



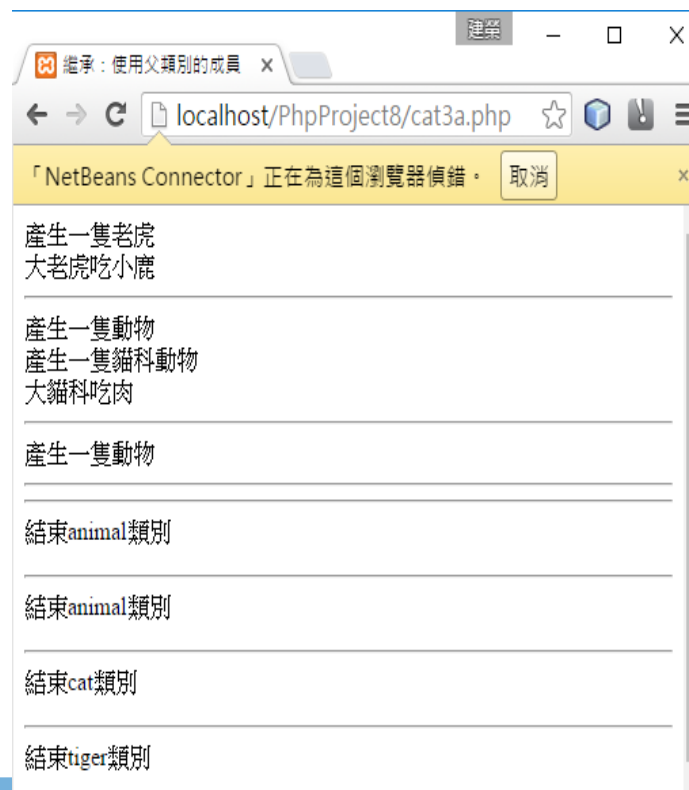
- 若cat類別內呼叫animal類別建構子與解構子，執行時會如何呈現呢？請將cat2.php另存新檔為cat3.php，我們修改cat這個類別內的語法，來了解會有甚麼變化，在此僅列出cat與tiger的語法，其他語法請參考cat2.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat3.php」）。



繼承特性



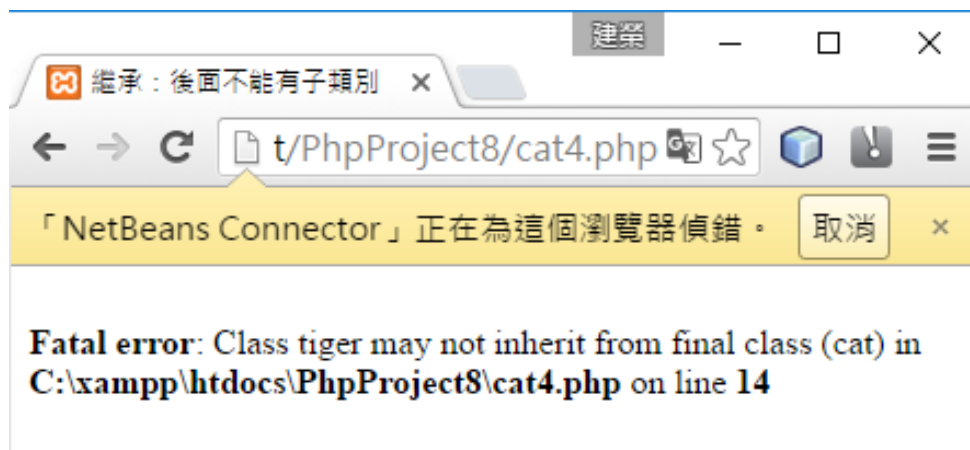
- 若要使用父類別內的成員，請於子類別內以\$this->父類別內的成員方式呼叫即可。
- 請練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat3a.php」）。



繼承特性



- 若希望類別後不要有子類別，可於class之前加上final，該類別就不會有其他類別作為子類別情形。
- 在此僅列出cat與tiger的語法，其他語法請參考cat2.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat4.php」）。



抽象類別

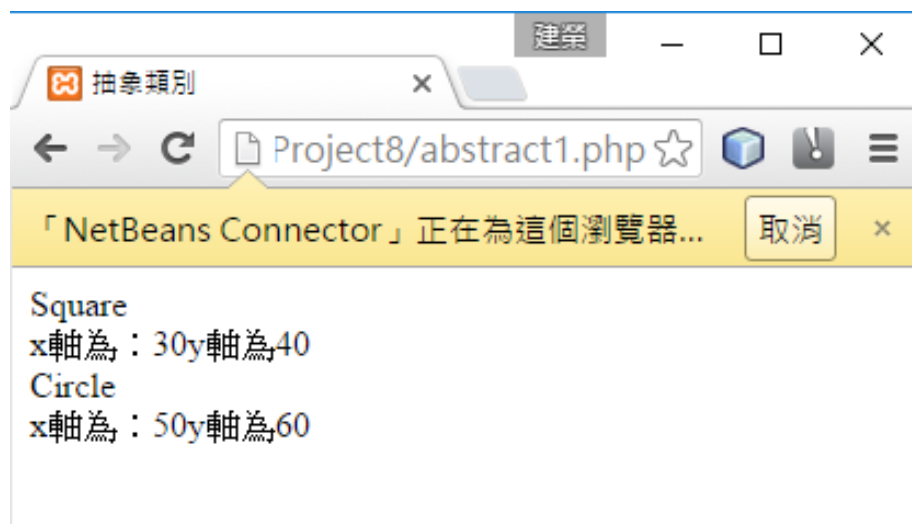


- 子類別可以使用父類別的成員與方法，父類別是否可以強制子類別一定要實作指定的方法呢？
- 抽象類別可讓您規劃類別時，確保子類別內一定會規劃指定的方法。
- PHP使用abstract 關鍵字來定義抽象類別和抽象方法，抽象類別無法被實體化產生物件。
- 繼承抽象類別的類別可被實體化產生物件，而子類別內一定得建立與父類別抽象方法同名方法。

抽象類別



- 請建立Square類別與Circle類別，並強制這兩個類別必須執行draw()方法。因為這兩個類別有具體必須實作的方法，所以規劃了Shape類別為父類別，並改為抽象類別型式，加入draw()抽象方法。
- 請練習觀察網頁執行結果（檔案名稱：（檔案名稱：「PhpProject8」資料夾內「abstract1.php」））。



抽象類別



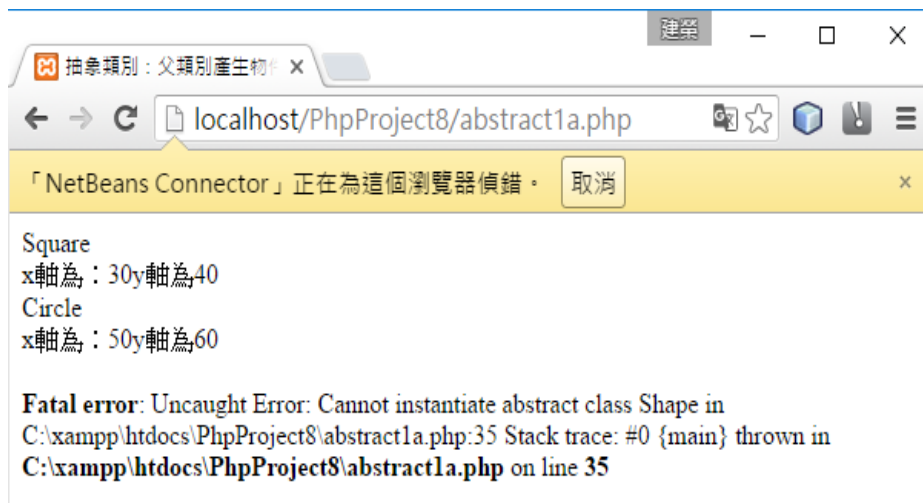
- 第07行宣告了Shape類別，請留意class之前加上「abstract」，代表這是抽象類別。
- 抽象類別內可放入抽象方法。
- 第13行宣告了一個draw()方法，而此方法於function之前加上「abstract」，代表這是抽象方法，所以抽象方法名稱之後直接使用()後接上分號，抽象方法內不寫內容。

```
07 abstract class Shape {  
08   protected $x, $y;  
09   public function setCenter($x, $y) {  
10     $this->x = $x;  
11     $this->y = $y;  
12   }  
13   abstract function draw( );  
14 }
```

抽象類別



- 抽象類別不能產生物件，所以抽象類別若產生物件，會產生什麼樣的回應呢？
- 請由abstract1.php另存新檔為abstract1a.php，在此僅列出程式流程語法，其他語法請參考abstract1.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「abstract1a.php」）。



抽象類別

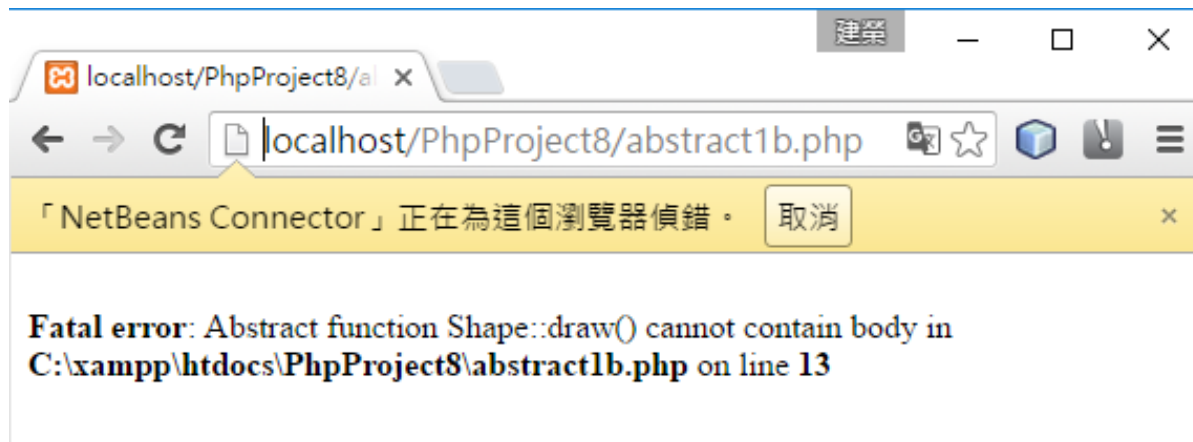


- 因為Shape類別是抽象類別，所以第35行語法由類別產生物件，就會產生錯誤訊息。
- 抽象類別不能產生物件，抽象方法內也不可以有內容。

抽象類別



- 若抽象方法內有內容，會產生什麼樣的回應呢？請由 abstract1.php 另存新檔為 abstract1b.php，在此僅列出 abstract function draw 函數語法，其他語法請參考 abstract1.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「abstract1b.php」）。



介面



- 當您規劃一個公司產品生產管理系統時，發現到C產品可以擁有A原料及B原料的多種行為方法，類別規劃時該如何進行呢？畢竟繼承只能有一個類別，而A、B、C之間也沒有繼承關係存在。
- 介面interface 是一個共同的操作環境，不同類別只要實作介面就可以操作同樣的行為，可以讓完全沒關係的數個類別來實作它，使完全不同的類別間可以有相同的部份。

介面



- 介面只定義方法，不可以宣告成員或屬性，所以介面不可以產生物件，要產生物件必須要類別去引用介面，物件才能帶入介面所定義的方法。
- 所有方法必須是public權限及abstract觀念。一個類別可以實做多個介面，一個介面也可以繼承多個介面，以擴充類別的功能。
- 介面宣告的語法為「interface 介面名稱」，而類別實做介面的語法為「class 類別名稱 implements 介面名稱」。

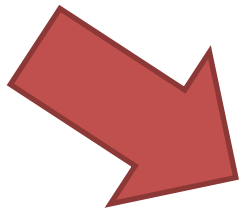
介面

- 飛機與汽車並無親屬關係，但兩者都是機械的一種，所以類別內可設計一個機械的介面，提供setname與operation兩個方法給類別引用後實作。
- 如何於PHP內表達飛機與汽車的資訊呢？

```
07 interface machine
08 {
09     public function setname($name);
10     public function operation( );
11 }
```



```
12 class plain implements machine
13 {
22 }
```

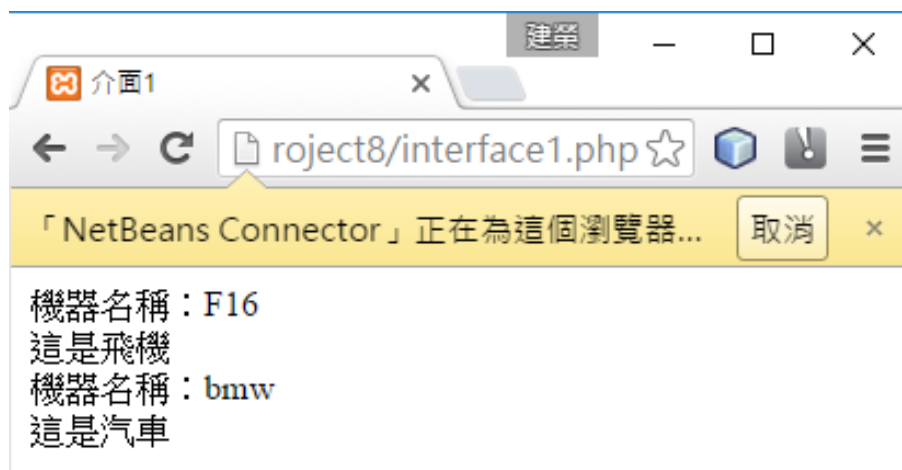


```
23 class car implements machine
24 {
33 }
```

介面



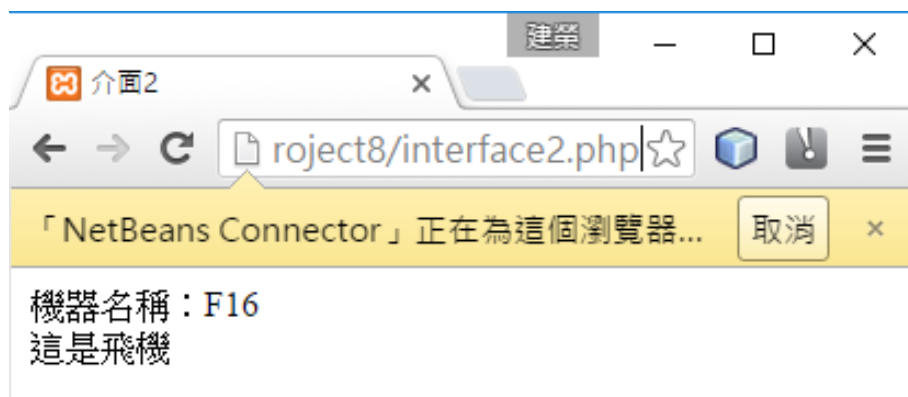
- 請練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface1.php」）。



介面



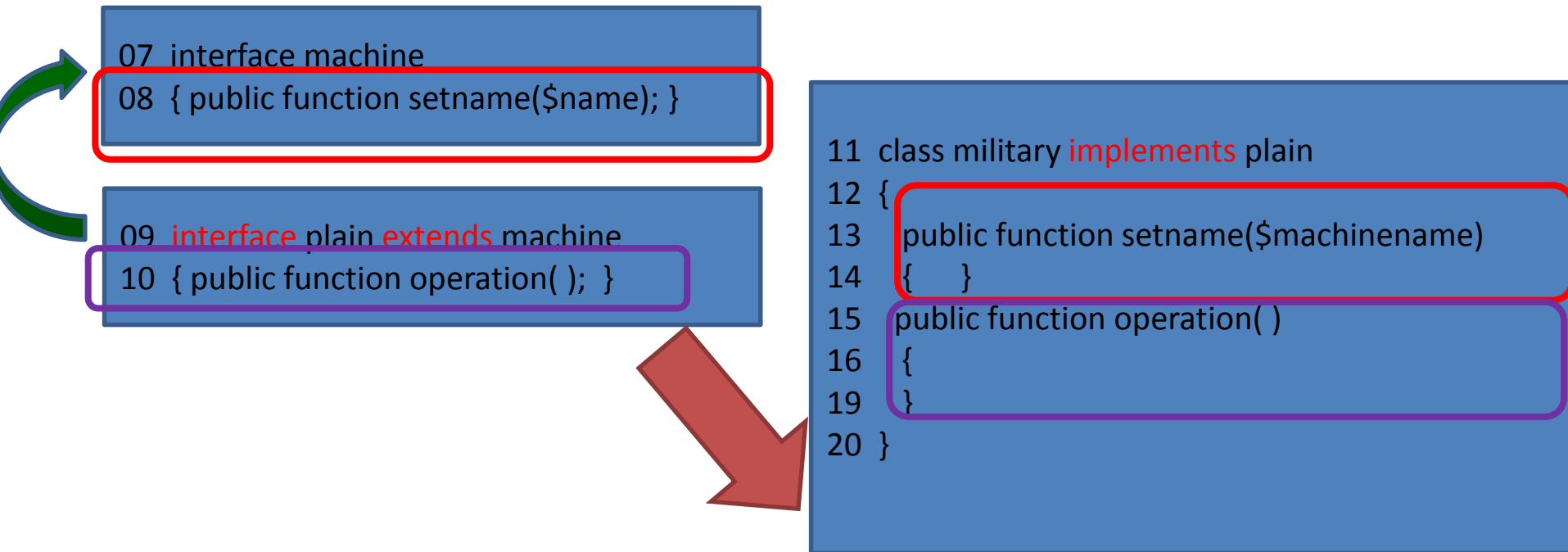
- 介面之間也可使用繼承的方式擴充功能，而類別再透過引用子介面方式擁有這兩個介面提供的功能。
- 請依練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface2.php」）。



介面



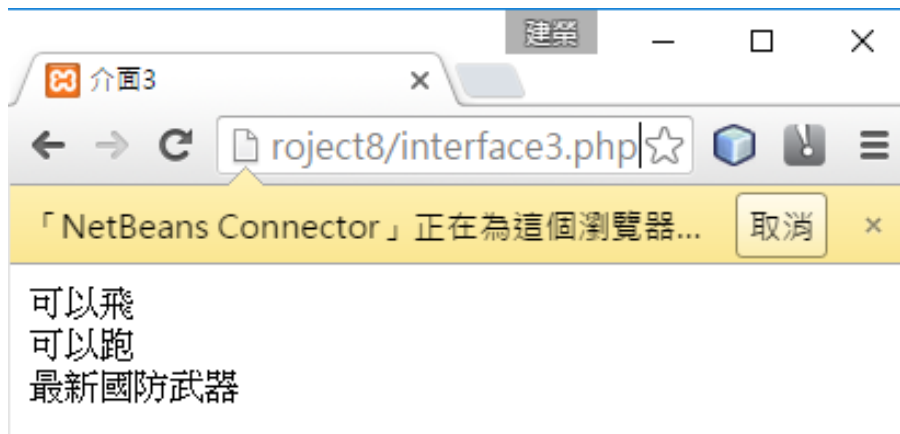
- 請依練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface2.php」）。



介面



- 介面可以繼承多個介面，例如army介面想要繼承plain與car兩個介面，請依以下練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface3.php」）。



介面



- 介面可以繼承多個介面，例如army介面想要繼承plain與car兩個介面，請依以下練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface3.php」）。

```
07 interface plain
08 { public function fly( );}
```

```
09 interface car
10 { public function run( ); }
```

```
11 interface army extends plain , car
12 { public function fire( ); }
```

```
13 class arms implements army
14 {
15     public function fly( )
16     { echo "可以飛"."<br>"; }
17     public function run( )
18     { echo "可以跑"."<br>"; }
19     public function fire( )
20     { echo "最新國防武器"."<br>"; }
21 }
```



多型特性



- 多型指呼叫名稱相同的函數，卻在不同情況下有不同的表現。
- 多型包括了方法過載 (Overloading) 與方法蓋寫 (Overriding) 兩種，不過PHP不支援方法過載。

方法過載



- 方法過載指類別內可定義多個同名的方法，依方法接收參數的類型與數量不同來判斷如何執行。
- 但PHP並不支援方法過載，所以PHP類別內請勿設計兩個同名的方法。

方法過載



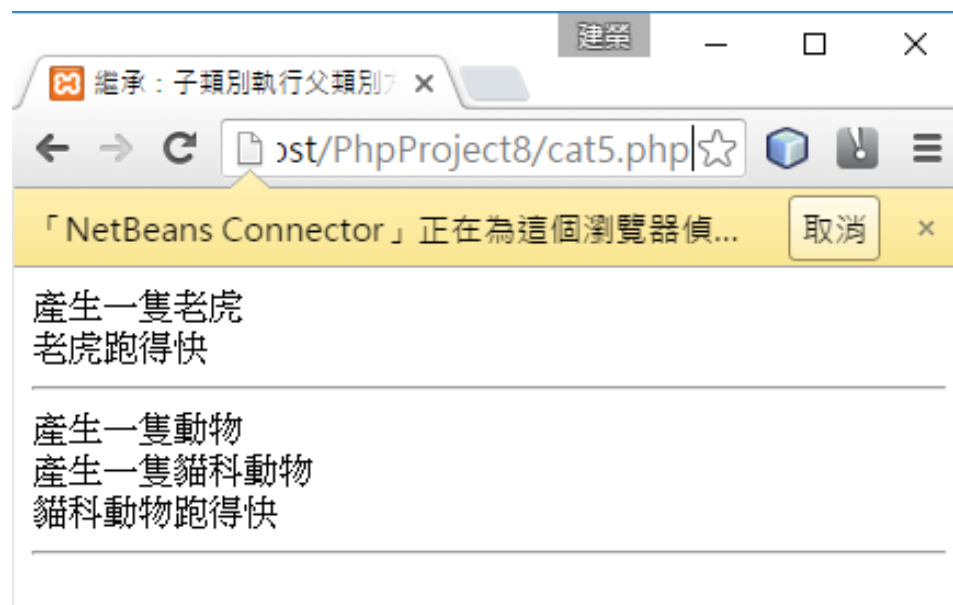
- 若類別內有兩個同名方法，網頁會如何呈現呢？請由box4.php另存新檔為box6.php，在此僅列出兩個同名的方法語法，其他語法請參考box4.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「box6.php」）。

```
11 function __construct($a,$b,$c) {  
12     $this->length1=$a;  
13  
14     Method or function "getdata" has already been declared  
15     ----  
16     (Alt-Enter shows hints)  
17 function getdata() {  
18     return $this->breadth1*$this->height1*$this->length1;  
19 }  
20 function getdata($a,$b) {  
21     return $this->breadth1*$this->height1*$this->length1;  
22 }
```

方法覆寫



- 當您不滿意由父類別那裡繼承到的方法時，您可以寫一個同名方法，就可將他覆蓋。
- 請依練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat5.php」）。



方法覆寫



- cat類別是tiger類別的父類別，而cat類別於第13行設計了run()方法，tiger類別於第19行也設計了run()方法。
- 當第27行由tiger類別產生的物件執行run()方法，您會發現到系統會執行tiger類別設計的run()方法，而不是父類別所提供的方法。

方法覆寫



- 所以當父子類別都有相同的方法時，依子類別產生的物件會執行子類別的方法，不會執行父類別的方法。

```
08 class cat extends animal {  
09   public function __construct( )  
13   public function run( )  
15 }
```

```
16 class tiger extends cat{  
17   public function __construct( )  
19   public function run( )  
21 }
```



```
26 $tiger1 = new tiger;  
27 $tiger1->run( );
```

方法覆寫



- 父類別內方法若為public權限，子類別產生的物件想要執行父類別的方法，有兩種方式。
- 第一種是子類別內不要有同名方法，子類別產生的物件就會直接執行父類別內的方法，請由cat5.php另存新檔為cat6.php，在此僅列出tiger類別內的語法，其他語法請參考cat5.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat6.php」）。

方法覆寫

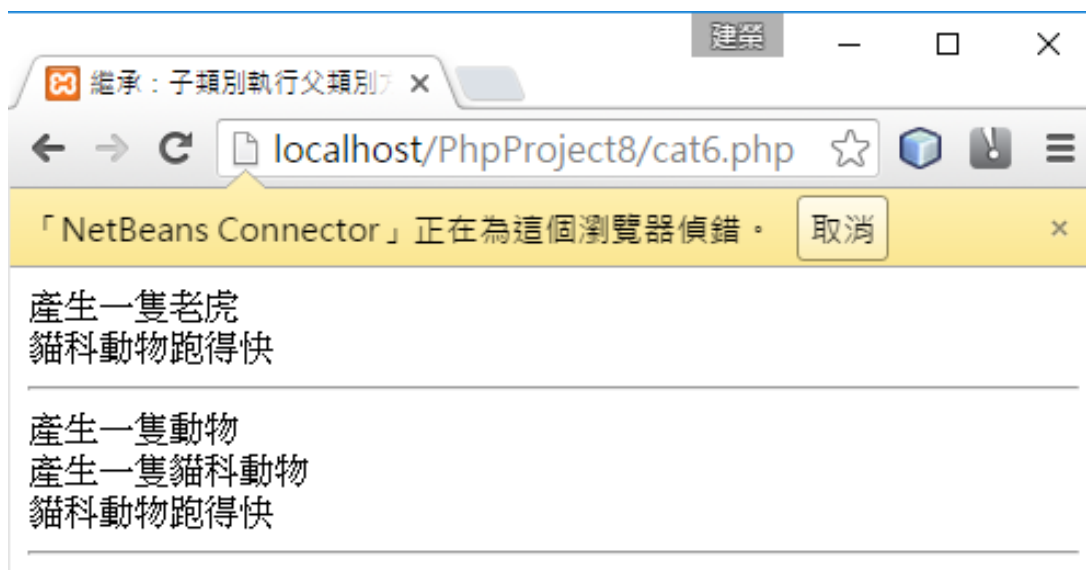


- 父類別內方法若為public權限，子類別產生的物件想要執行父類別的方法，有兩種方式：
- 第一種是子類別內不要有同名方法，子類別產生的物件就會直接執行父類別內的方法。

方法覆寫



- 請由cat5.php另存新檔為cat6.php，在此僅列出tiger類別內的語法，其他語法請參考cat5.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat6.php」）。



方法覆寫



- 請由cat5.php另存新檔為cat6.php，在此僅列出tiger類別內的語法，其他語法請參考cat5.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat6.php」）。

```
08 class cat extends animal {  
09     public function __construct( )  
13     public function run( )  
15 }
```

```
16 class tiger extends cat{  
17     public function __construct( )  
  
21 }
```

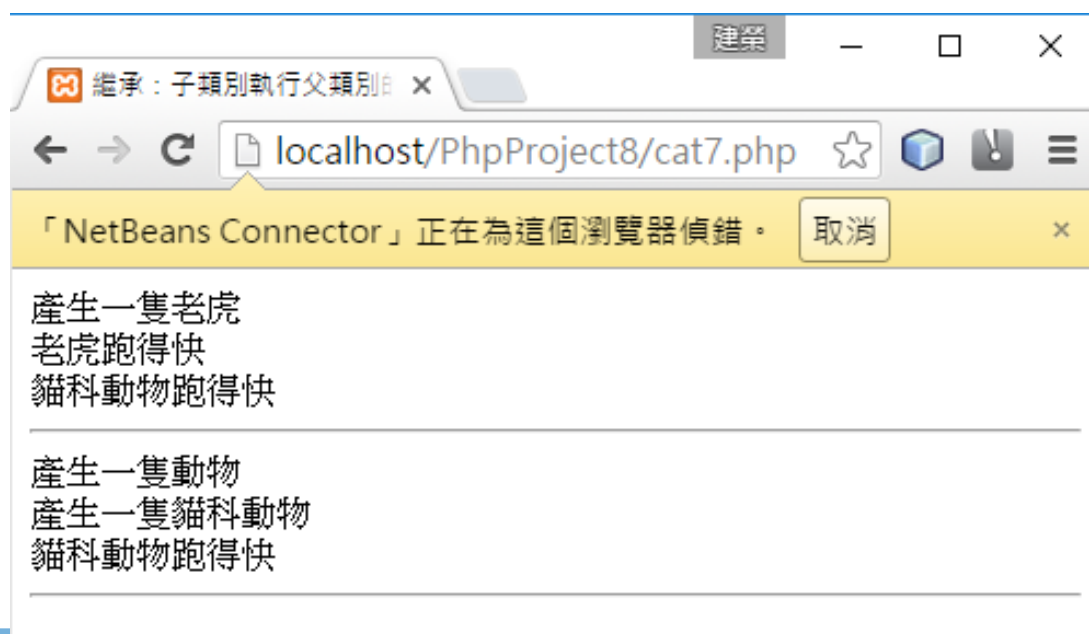


```
26 $tiger1 = new tiger;  
27 $tiger1->run( );
```

方法覆寫



- 第二種是子類別內以「parent::方法名稱()」方式呼叫，子類別產生的物件就會執行父類別內的方法。
- 請由cat5.php另存新檔為cat7.php，在此僅列出tiger類別內的語法，其他語法請參考cat5.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat7.php」）。



方法覆寫



- cat類別是tiger類別的父類別，而cat類別於第13行設計了run()方法，tiger類別於第19行也設計了run()方法。
- tiger類別於run()方法內執行「parent::run()」語法(第22行)，執行父類別的run()方法。

方法覆寫



- 當tiger類別產生的物件執行run()方法，您會發現到系統除執行子類別，也就是tiger類別的run()方法外，也執行父類別所提供的方法。

```
08 class cat extends animal {  
09   public function __construct( )  
13   public function run( )  
15 }
```

```
16 class tiger extends cat{  
17   public function __construct( )  
19   public function run( )  
20   {  
21       echo "老虎跑得快". "<br>";  
22       return parent::run( );  
23   }
```

```
26 $tiger1 = new tiger;  
27 $tiger1->run( );
```

新增物件導向功能

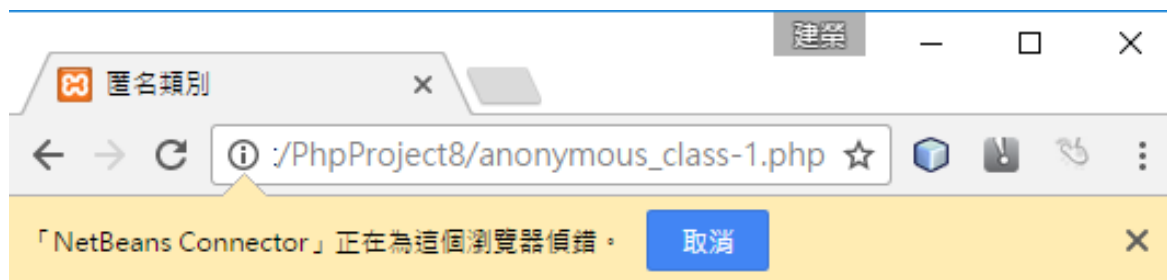


- PHP7於物件導向這部分功能如下：
 - 新增匿名類別
 - 修改use方法
 - 新增call方法
 - 新增例外處理
- 新增的例外處理於10-1節介紹，而call方法將於第11-3-3節介紹，本章將介紹其他項目。

新增匿名類別



- PHP7支援匿名類別，這種類別沒有類別名稱，直接使用new方式建立物件。
- 通常用於只需要使用一次物件的情況下進行。請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「anonymous_class-1.php」）。



修改use方法



- PHP可以使用載入其他網頁的namespace命名空間內的常數、函數、類別與靜態方法，在此我們規劃了第一個library檔案（檔案名稱：「PhpProject8」資料夾內「lib1.php」）。
- 接著規劃第二個library檔案（檔案名稱：「PhpProject8」資料夾內「lib2.php」）。

修改use方法



- 於PHP7之前的版本您得使用多次的use語法才能呼叫後使用（檔案名稱：「PhpProject8」資料夾內「uselib.php」）。

```
05 use App\Lib1\MyClass as C1;  
06 use App\Lib1\MyClass2 as C2;  
07 use App\Lib2\MyClass as C3;  
08 use function App\Lib1\MyFunction as Cf1;  
09 use function App\Lib1\MyFunction2 as Cf2;  
10 use function App\Lib2\MyFunction as Cf3;  
11 use const App\Lib1\MYCONST1 as Cont1;  
12 use const App\Lib1\MYCONST2 as Cont2;  
13 use const App\Lib2\MYCONST as Cont3;
```

修改use方法



- PHP7可以使用一個use語法導入同一個命名空間內的類別、方法與常數（檔案名稱：「PhpProject8」資料夾內「uselib2.php」）。

```
05 use App\Lib1\{MyClass as C4,MyClass2 as C5};  
06 use function App\Lib1\{MyFunction as Cf4,MyFunction2 as Cf5};  
07 use const App\Lib1\{MYCONST1 as Cont4,MYCONST2 as Cont5};
```

問題與討論



- 1.如何呼叫父類別的建構子與解構子?
- 2.如何使用父類別內的成員?
- 3.請問抽象類別可以產生物件嗎?
- 4.請問抽象方法可以有內容嗎?
- 5.請問子類別一定要實作抽象方法嗎?
- 6.何謂介面?
- 7. PHP支援方法過載嗎?
- 8.請解釋甚麼是匿名類別。