



Chapter [6]

類別class
與封裝

類別是抽象的，而物件就是類別的實體。在這一章我們將介紹物件類別圖來解釋類別。

物件導向程式設計 (object oriented programming 簡稱 OOP) 有幾個特色，分別為類別 class、繼承 inheritance、多型 polymorphism。

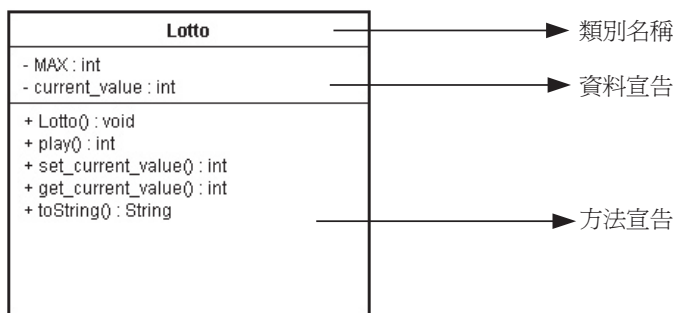
資料封裝指的是在類別成員 (資料欄或方法) 宣告存取控制。使用公用 public、私有者 private、預設或受保護 protected，私有者 private 只有本類別的方法允許使用，預設 default 是只有在相同套件中才能存取，保護者 protected 只有在本類別其衍生類別 (子類別) 或相同套件中的類別可以存取，公用者 public 任何類別都允許存取。

衍生類別為本類別的次類別 (subclass) 或稱為子類別、延伸類別 (extended class)，本類別稱為子類別的父類別 (super class) 或稱為父類別 (祖先類別)。

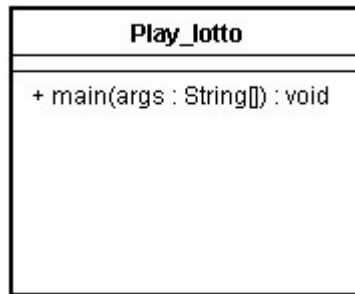
6-1 類別

這是一個樂透遊戲，我們使用兩個骰子來玩。

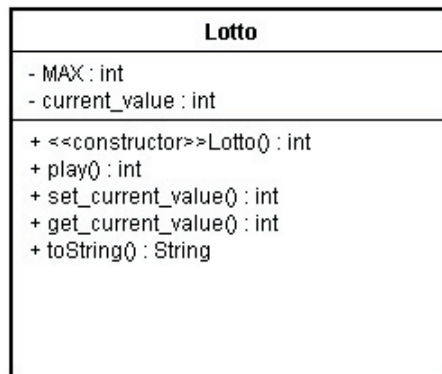
類別是由資料和方法所組成。類別名稱就是類別的名子；資料宣告就是儲存在每個類別物件裏的資料；方法就是每個物件所提供的服務。我們定義骰子為 Lotto 類別。



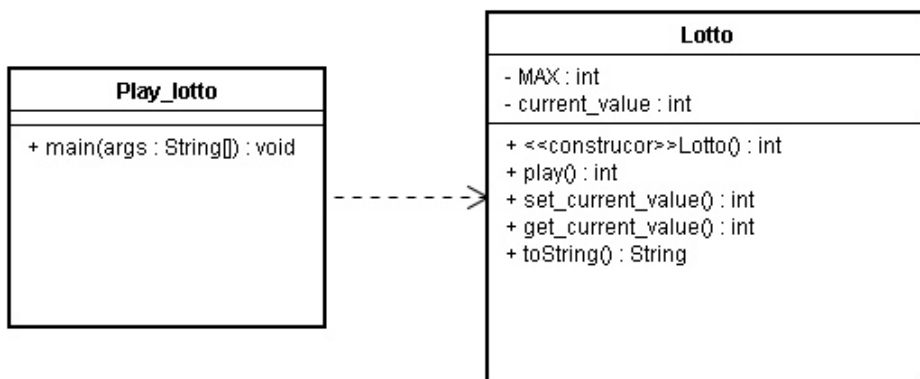
`public static void main(String[] args)` 是我們程式起始的地方，只有宣告 `main()` 函數的類別才是我們程式開始的地方。我們定義玩樂透遊戲 (丟骰子) 為 `Play_lotto` 類別。樂透遊戲會丟兩個骰子。



Lotto 類別包含了兩個私有 private(符號 -) 資料，分別是整數 int 資料型態的 MAX 和整數 int 資料型態的 current_value。MAX 資料代表骰子最大的點數，而 current_value 則代表目前骰子的點數。Lotto() 函數為 Lotto 類別的建構子 constructor。Lotto 類別也包含了四個成員函數，分別是 play() 函數，它為公用的 public(符號 +)，回傳值是整數 int；set_current_value() 函數，它為公用的 public(符號 +)，回傳值是整數；get_current_value() 函數，它為公用的 public，回傳值是 int 整數；toString() 函數，它為公用的 public，回傳值是字串 String。



這個 UML 的圖形顯示 Play_lotto 類別使用了 Lotto 類別，而虛線箭頭是表示使用（相依）的關係。Play_lotto 類別使用了 Lotto 類別。Play_lotto 類別會開始 (main() 開始執行) 丟兩個 Lotto 類別物件的骰子，然後作加總的工作。



範例

Play_lotto.java

第二行的main()函數為程式起始的地方。

第四行為sum總合的宣告，為整數int資料型態。

第五行和第六行新增Lotto類別的物件game1和game2，Lotto類別定義在Lotto.java範例中。

第七行執行game1物件的play()函數，這是丟骰子的動作，它會得到0到6的隨機數。

第九行得到game1骰子的點數和game2骰子的點數。

第十行開始為再次丟骰子，執行另外一個遊戲。

第十一行丟第一個骰子。

第十二行則設定骰子的點數為5，使用game2物件的set_current_value()函數，並且把5當參數帶入。

第十四行使用get_current_value()函數來得到骰子的點數，並且將骰子的點數相加再分配給sum變數。

第十六行開始為再次丟骰子，執行第三個遊戲。

```

1 public class Play_lotto{
2     public static void main(String[] args){
3         Lotto game1,game2;
4         int sum;
5         game1=new Lotto();
6         game2=new Lotto();
7         game1.play();
8         game2.play();
9         System.out.println(" 遊戲1:"+game1+" 遊戲2:"+game2);
10        System.out.println("-----");
11        game1.play();
12        game2.set_current_value(5);
13        System.out.println(" 遊戲1:"+game1+" 遊戲2:"+game2);
14        sum=game1.get_current_value()+game2.get_current_value();
15        System.out.println(" 遊戲總合1:"+sum);
16        System.out.println("-----");
17        sum=game1.play()+game2.play();
18        System.out.println(" 遊戲1:"+game1+" 遊戲2:"+game2);
19        System.out.println(" 遊戲總合2:"+sum);
20    }
21 }

```

這是執行的情況。

C:\Program Files\Xinox Software\Ucreator\Y3LENGE2001.exe	C:\Program Files\Xinox Software\Ucreator\Y3LENGE2001.exe
遊戲1:3 遊戲2:3	遊戲1:1 遊戲2:1
-----	-----
遊戲1:2 遊戲2:5	遊戲1:5 遊戲2:5
遊戲總合1:7	遊戲總合1:10
-----	-----
遊戲1:5 遊戲2:6	遊戲1:5 遊戲2:2
遊戲總合2:11	遊戲總合2:7

範例 Lotto.java

第一行到第二十一行是定義Lotto類別，它是一個骰子的類別，MAX則是總共有六個點數，current_value則是骰子目前的點數。

第四行為Lotto類別的建構子，它會設定骰子目前的點數為六點。

第七行到第十行是執行丟play()骰子，它會隨機得到目前骰子的點數，從1到6點，然後回傳return，回傳的資料型態為整數int。

第十一行到第十三行為設定目前的點數，current_value。

第十四行到第十六行為得到目前的點數，它會回傳目前的點數。

第十七行到第二十行則將current_value由整數型態轉成字串型態，再分配給字串convert，然後回傳return。

```
1 public class Lotto{
2     private final int MAX=6;
3     private int current_value;
4     public Lotto(){
5         current_value=6;
6     }
7     public int play(){
8         current_value=(int)(Math.random()*MAX)+1;
9         return current_value;
10    }
11    public void set_current_value(int value){
12        current_value=value;
13    }
14    public int get_current_value(){
15        return current_value;
16    }
17    public String toString(){
18        String convert=Integer.toString(current_value);
19        return convert;
20    }
21 }
```

6-2 建構子

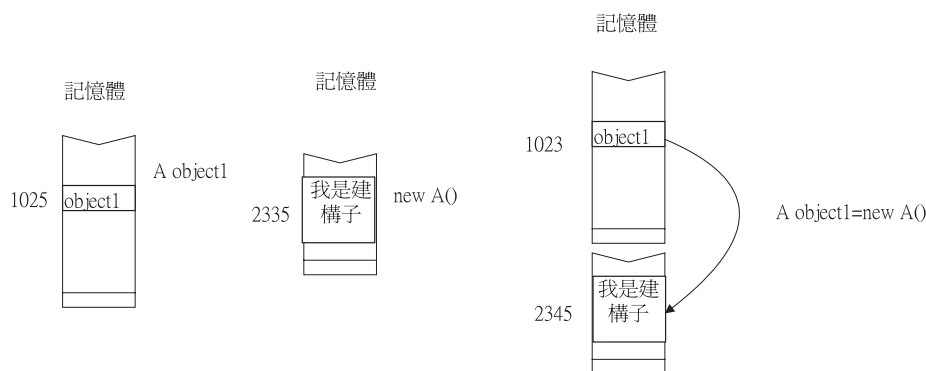
當我們建立物件時，我們常要初始化物件的實體變數。初始化實體變數，是最經常看到的初始化。建構子是當我們使用 `new` 來新增類別實體時，Java 會自動執行該函數，以便初始化物件。在 Java 中，當函數和他的類別名稱相同時，這就是該類別的建構子。當在類別內定義相同名稱的函數，則這函數就被稱為建構子。建構函數是沒有回傳值的，而且它不能在方法前指定型態，`void` 也不能放置到建構函數前。

我們使用 `new` 來新增物件，這時我們就使用了建構子。`new A()` 的動作就是呼叫建構子。建構子函數是函數的特別變形。

```
A object1=new A();
```

在建構子內是可以呼叫其它的函數，這是合法的因為建構子第一個動作是自動建立物件的實體變數。

建立物件第一步是新增 A 類別型態的變數 `object1`。第二步驟是 `new A()`，也就是新增 `A()` 類別的物件，在主記憶體上為它配置空間。第三步驟是將剛剛在記憶體上新增物件的位置給 `object1` 變數，這樣就完成參考 (物件位置的分配)，也就是將物件的位置給 A 類別型態的 `object1` 變數。



範例 AA.java

第一行到第六行為類別AA的定義。

第二到第四行為執行程式。

第三行新增A類別的物件object1，這時建構子就會執行，因此會顯示資料。

```
1 public class AA{
2     public static void main(String[] args){
3         A object1=new A();
4     }
5 }
```

這是顯示類別A的物件object1，它會自動執行建構A()函數，因此顯示”我是建構子”。



我是建構子

範例 A.java

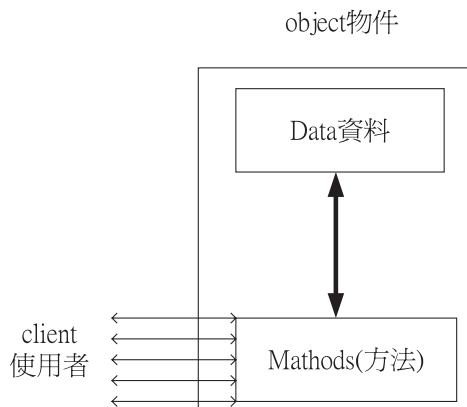
第一行到第五行是定義A類別。

第二行到第四行為類別A的建構函數A()，我們輸出字串”我是建構子”。

```
1 public class A{
2     public A(){
3         System.out.println("我是建構子");
4     }
5 }
```


6-3 封裝

物件需要封裝，來確保資料不會直接被不正當的存取。物件需要封裝於其它的系統。物件應該只開特定的方法來和程式其它部份交談，以確保程式的穩定性。這些方法定義了介面，為物件和使用它們程式間的介面。使用者端使用這物件的程式碼，為了物件本身的穩定，使用者（外部程式碼 client）不應該直接存取物件 object 內部的資料。使用者 client 應該呼叫物件 object 的方法，而再使用物件的方法來存取物件 object 內部的資料。例如在樂透程式 Play_lotto.java，在 main() 主程式，它呼叫了 game1 物件的 play() 方法，然後就會得到骰子的點數。因此 main() 主程式應該透過物件方法來存取物件資料，而不是自己直接去存取資料，這是確保物件資料的安全性。使用者不直接存取 object 物件資料，而透過方法介面去存取。



在 Java 中，我們使用存取控制修飾子來完成物件封裝的動作。存取控制修飾子就是指定程式語言建構的特殊字元。

6-3-1 存取控制修飾子

在 Java 中，我們使用存取控制修飾子來完成物件封裝的動作。存取控制修飾子就是指定程式語言建構的特殊字元。



類別中所有資料欄及方法在本類中均可存取得到，但要控制由其它的類別來存取本類別中的資料欄位及方法，或者如何由次類別來繼承本類別中的資料欄位及方法，Java 提供下列的存取控制修飾子 (access control modifier)。

public(公用者):資料欄及方法均為類別的成員。成員(資料或方法)以 **public** 宣告者，只要該類別可存取到的地方均可存取，並可由子類別繼承。為一般資料欄及方法的預設存取控制修飾子。**public** 的變數是違反 Java 封裝的原則，因為它允許外部的程式碼改變物件內部的資料。**public** 的方法提供服務給使用者端 client，外部程式碼透過這個介面來存取資料。在 UML 中，其符號是 (+)。

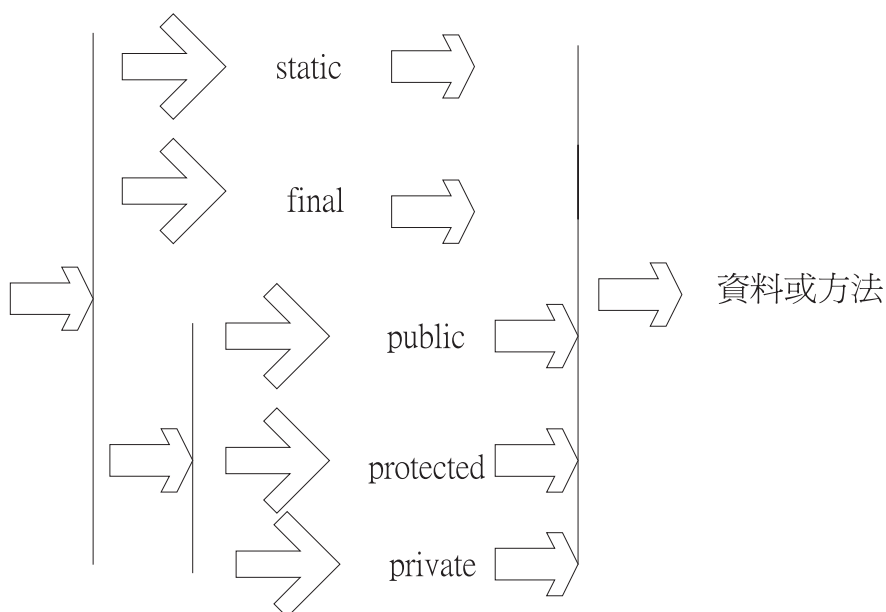
private(私用者):資料或方法成員以 **private** 宣告者，只能在本類別中存取。**Private** 方法是支援本類別其它的方法。在 UML 中，其符號是 (-)。

	public	private
variable 變數	違反封裝 原則	加強封裝
method 方法	提供服務 給使用者 端client	支援本類別 的其它方法

protected(保護者):資料或方法成員以 **protected** 宣告者，除了在本類別中可存取外，在次類別中也可存取，並可由次類別繼承，但是它存取是受到保護的，只有在相同的套件中才能被存取。

final: 加上 **final** 修飾子的為常數，因為變數加上 **final** 後，它的資料就不能被改變，因此可以在前面加上 **public**。

這是定義資料或方法的圖，**static** 和 **final** 這兩個關鍵字我們在後面再說明。我們的存取控制修飾子 **public**、**protected** 和 **private** 是放到資料或方法的前面。



當我們沒有使用明確的修飾子時，就會使用預設的修飾子。預設的修飾子有本套件內的存取範圍，但是我們不可以將保留字 `package` 當作修飾子。類別或介面有 `public` 或預設的存取範圍。只有內部的類別才有 `private` 修飾子，而且它只能被內部的類別存取。

修飾子	UML符號	類別或介面	方法或變數
預設(沒有修飾子)	~	在本套件中都可以使用。	在本套件中的任何類別都可以使用。
public	+	任何地方都可以使用。	任何地方都可以使用。
protected	#	沒有使用到。	在本套件中的任何類別都可以使用，或由繼承類別也可以使用。
private	-	只有在本類別中才可使用。	不可由其它類別使用。

public(公用者):資料欄及方法均為類別的成員。成員(資料或方法)以 `public` 宣告者，只要該類別可存取到的地方均可存取，並可由子類別繼承。

private(私用者):資料或方法成員以 `private` 宣告者，只能在本類別中存取。

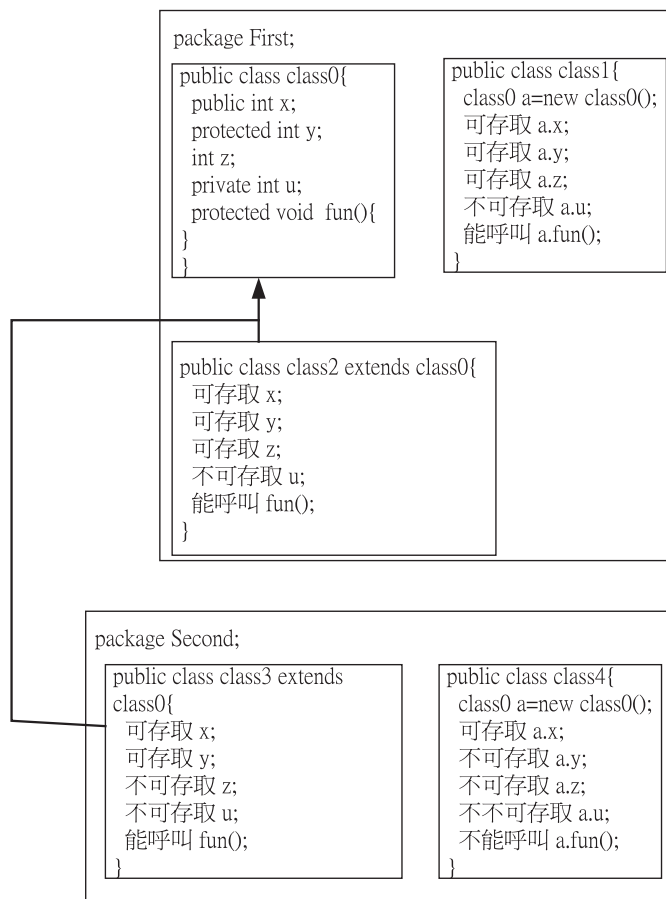


protected(保護者)：資料或方法成員以 **protected** 宣告者，除了在本類別中可存取外，在子類別中也可存取，並且在相同套件中的任何類別皆可存取，在不同套件中的子類別也能存取。

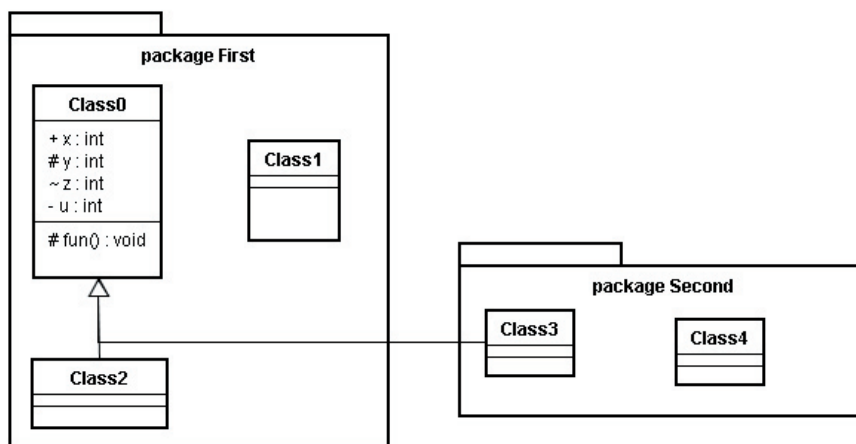
預設 (沒有修飾子) 是只可以在本套件中作存取。

這是表示 **class0** 中 **public**、**protected**、**default** 和 **private** 的資料和方法能夠被在相同套件中 **class1** 類別存取的情況；**class0** 被子類別 **class2** 存取情況；**class0** 被在不同套件中的子類別 **class3** 存取情況；**class0** 被在不同套件 **class4** 存取情況。

沒有修飾子的資料、方法或類別，不可以在不同套件外被存取，例如變數 **z**，一般沒有修飾子的資料為預設 **default**。



這是 UML 的表示方法。



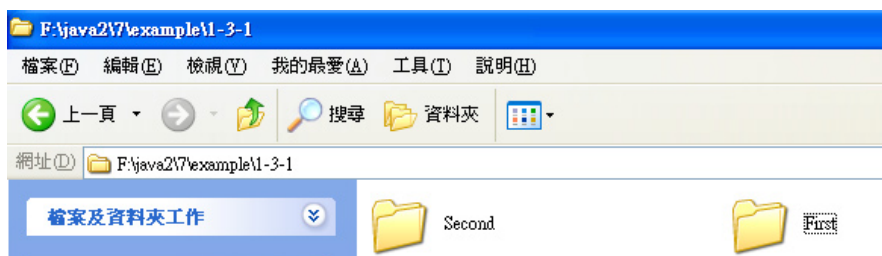
這是 Java 程式設計中，其它常用的修飾子。

修飾子	類別class	介面interface	方法method	變數variable
abstract	宣告成abstract抽象修飾子的類別不能被實體化。	所有的介面都是抽象的類別。abstract可以被選擇使用。	抽象方法是要被繼承實作的，因此抽象方法本身沒有任何作用。	沒有應用到。
final	宣告成final的類別是不能再被繼承。	沒有被應用到。	被宣告成final的類別不能被覆寫。	被宣告成final的變數為常數，它的值不能被改變。
native	沒有被應用到。	沒有被應用到。	被宣告成native的方法是要被其它語言的程式實作。	沒有被應用到。

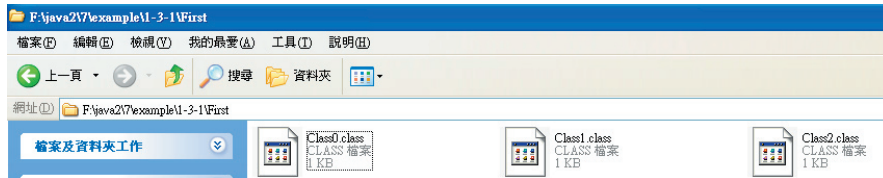
修飾子	類別class	介面interface	方法method	變數variable
static	沒有被應用到。	沒有被應用到。	被宣告成static的方法，就像是隱藏性的final。定義一個類別的方法。它不需要實體化的物件來呼叫。	定義一個類別的變數。它可以被該類別的所有實體所分享。它不需要實體化的物件來參考
synchronized	沒有被應用到。	沒有被應用到。	方法得執行是互相獨立的執行緒。	沒有被應用到。
transient	沒有被應用到。	沒有被應用到。	沒有被應用到。	變數將不會被serialized。
volatile	沒有被應用到。	沒有被應用到。	沒有被應用到。	這變數被非同步的改變。

這是我們的資料夾，裏面有 First 目錄和 Second 目錄。在這個設定前我們要設定我們類別路徑 classpath，將 classpath 設定成目前目錄所在地方。假設目前目錄是在 F:\java2\7\example\1-3-1，我們可以用下列方法來指示，這樣它就可以讀取到該類別的路徑了。 . 逗號，是代表本目錄路徑。因為有些檔案在別的套件目錄中，我們需要設定路徑，這樣 Java 才讀取得到該類別。

```
F:\>set CLASSPATH=.;F:\java2\7\example\1-3-1\;
```



First 目錄存放 Class0.class、Class1.class 和 Class2.class 三個已經編譯好的檔案。



Second 目錄存放 Class3.class 和 Class4.class 已經編譯好的檔案。



我們逐一將這些檔案給編譯，再將它放到相對應的目錄 First 和 Second 中。

```
F:\java2\7\example\1-3-1>javac Class0.java
F:\java2\7\example\1-3-1>javac Class1.java
F:\java2\7\example\1-3-1>javac Class2.java
F:\java2\7\example\1-3-1>javac Class3.java
F:\java2\7\example\1-3-1>javac Class4.java
```

範例 Class0.java

這是我們定義在First套件中的Class0類別。

第一行我們宣告Class0類別是在First套件中。

第三行宣告資料成員x為整數，而且它為public公眾的，任何地方任何人都可以存取。

第四行宣告資料成員y為整數，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。

第五行宣告資料成員z為整數，因為沒有修飾子，所以它是預設default的，只有在相同套件中才能夠存取它。

第六行宣告資料成員u為整數型態，而且它為private私有的，只有在本類別中才能夠存取它。

第七行宣告成員函數fun()，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。

```

1 package First;
2 public class Class0{
3     public int x=1;
4     protected int y=2;
5     int z=3;
6     private int u=4;
7     protected void fun(){
8         System.out.println("Class0的函數");
9     }
10 }

```

範例 Class1.java

這是我們定義在First套件中的Class0類別。

第一行我們宣告Class1類別是在First套件中。

第三行到第十行為我們的主程式main()。

第四行新增Class0的物件a。

第五行資料x為整數，而且它為public公眾的，任何地方任何人都可以存取。

第六行y為整數，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。Class0和Class1同在First套件中。

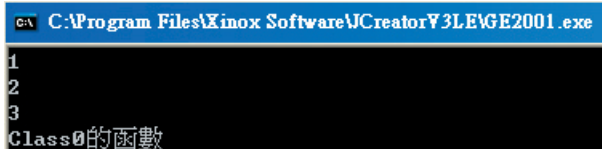
第七行z為整數，它是預設default的，只有在相同套件中才能夠存取它。

第八行u為整數型態，而且它為private私有的，只有在Class0類別中才能夠存取它，所以Class1類別的物件不能存取它。否則會發生編譯錯誤。

第九行宣告成員函數fun()，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。


```
1 package First;
2 public class Class1{
3     public static void main(String[] args){
4         Class0 a=new Class0();
5         System.out.println(a.x);
6         System.out.println(a.y);
7         System.out.println(a.z);
8         //System.out.println(a.u);
9         a.fun();
10    }
11 }
```

這是執行的情況。



範例 Class2.java

這是我們定義在First套件中的Class0類別。

第一行我們宣告Class2類別是在First套件中。

第二行Class2繼承Class0類別。

第三行到第十行為我們的主程式main()。

第四行新增Class2的物件b。

第五行資料x為整數，而且它為public公眾的，任何地方任何人都可以存取。

第六行y為整數，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。Class0和Class2同在First套件中。

第七行z為整數，它是預設default的，只有在相同套件中才能夠存取它。

第八行u為整數型態，而且它為private私有的，只有在Class0類別中才能夠存取它，所以Class2類別的物件不能存取它。否則會發生編譯錯誤。

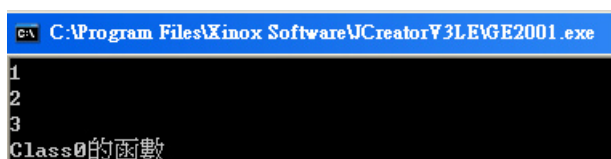
第九行宣告成員函數fun()，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。

```

1 package First;
2 public class Class2 extends Class0{
3     public static void main(String[] args){
4         Class2 b=new Class2();
5         System.out.println(b.x);
6         System.out.println(b.y);
7         System.out.println(b.z);
8         //System.out.println(b.u);
9         b.fun();
10    }
11 }

```

這是執行的情況。



```

C:\Program Files\Xinox Software\JCreator\J3LE\GE2001.exe
1
2
3
Class0的函數

```

範例 Class3.java

這是我們定義在Second套件中的Class3類別。

第一行我們宣告Class3類別是在Second套件中，和First不同套件。

第二行我們輸入了First套件的Class0類別。

第三行Class3繼承Class0類別。

第四行到第十一行為我們的主程式main()。

第五行新增Class3的物件c。

第六行資料x為整數，而且它為public公眾的，任何地方任何人都可以存取。

第七行y為整數，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。Class0和Class3不在同一個套件中，但Class3卻是Class0的子類別，所以可以存取它。

第八行z為整數，它是預設default的，只有在相同套件中才能夠存取它。否則會發生編譯錯誤。

第九行u為整數型態，而且它為private私有的，只有在Class0類別中才能夠存取它，所以Class2類別的物件不能存取它。否則會發生編譯錯誤。

第十行宣告成員函數fun()，而且它為protected受保護的，只有在相同套件

和子類別才能夠存取它。所以Class3的物件c可以存取它。

```

1 package Second;
2 import First.Class0;
3 public class Class3 extends Class0{
4     public static void main(String[] args){
5         Class3 c=new Class3();
6         System.out.println(c.x);
7         System.out.println(c.y);
8         //System.out.println(c.z);
9         //System.out.println(c.u);
10        c.fun();
11    }
12 }

```

這是執行的情況。



範例 Class4.java

這是我們定義在Second套件中的Class4類別。

第一行我們宣告Class4類別是在Second套件中，和First不同套件。

第二行我們輸入了First套件的Class0類別。

第四行到第十一行為我們的主程式main()。

第五行新增Class0的物件d。

第六行資料x為整數，而且它為public公眾的，任何地方任何人都可以存取。

第七行y為整數，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。Class0和Class4不在同一個套件中，Class4也不是Class0的子類別，所以不可以存取它，否則會發生編譯錯誤。

第八行z為整數，它是預設default的，只有在相同套件中才能夠存取它。否則會發生編譯錯誤。

第九行u為整數型態，而且它為private私有的，只有在Class0類別中才能夠存取它，所以Class2類別的物件不能存取它。否則會發生編譯錯誤。

第十行宣告成員函數fun()，而且它為protected受保護的，只有在相同套件和子類別才能夠存取它。所以Class4的物件d不可以存取它，否則會發生編譯錯誤。

```
1 package Second;
2 import First.Class0;
3 public class Class4 {
4     public static void main(String[] args){
5         Class0 d=new Class0();
6         System.out.println(d.x);
7         //System.out.println(d.y);
8         //System.out.println(d.z);
9         //System.out.println(d.u);
10        //d.fun();
11    }
12 }
```

這是執行的情況。



6-4 過載Overloading

我們也可以在同一個類別來宣告多個相同名稱的函數，這些都叫 Overloading 過載。相同名稱的函數，但是他們所帶入的參數不相同，Java 編譯器將以函數的簽名 (函數名稱和參數) 來決定使用哪個函數，這就是方法的過載 method overloading。

假如我們呼叫範例 Overloading_max.java 的 max 函數並且代入的參數是浮點數 double，則這個 max(double a,double b) 的函數將自動被啟動。

```
public static double max(double a,double b){
    if(a>b)
        return a;
    else
        return b;
}
```

假如我們呼叫範例 Overloading_max.java 的 max 函數並且代入的參數是整數 int，則這個 max(int a,int b) 的函數將自動被啟動。

```
public static int max(int a,int b){
    if(a>b)
        return a;
    else
        return b;
}
```

當我們過載 overload 一個方法名稱，每一個定義在這類別的方法，其簽名 signature 必須不同。

一個方法的簽名是由方法的名稱和參數型態的串列所組成。方法的回傳型態不屬於方法簽名的一部份。

例如：public static int max(int a,intb)
它的簽名signature是
max(int,int)

過載 overloading 和自動型別轉換的先後順序。Java 總是先尋找適合函數呼叫的簽名 signature，然後再試著自動作型別轉換。Java 會先找尋符合參數型別的函數，如果它能符合。當它都找不到時，它會試著自動型別轉換，來搜尋符合呼叫函數型別的函數定義。

範例 Overloading.java

第七行和第十三行的max()簽名不同，簽名是由函數名稱和參數串列組合而成，相同的參數名稱，但不相同的參數，這樣函數就不一樣了，這就是過載 Overloading。

第七行為max()函數，它的名稱為max，它的參數為double a和double b，浮點數a和浮點數b。

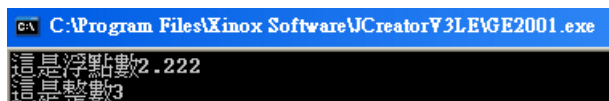
第十三行為max()函數，它的名稱為max,但是它的參數為int a和int b，整數a和整數b。

```

1 public class Overloading{
2     public static void main(String[] args){
3         Overloading ov=new Overloading();
4         ov.max(1.5,2.222);
5         ov.max(3,2);
6     }
7     public static void max(double a,double b){
8         if(a>b)
9             System.out.println("這是浮點數"+a);
10        else
11            System.out.println("這是浮點數"+b);
12    }
13    public static void max(int a,int b){
14        if(a>b)
15            System.out.println("這是整數"+a);
16        else
17            System.out.println("這是整數"+b);
18    }
19 }

```

第四行和第五行同樣呼叫max()函數，但帶入的參數不同，一個為雙精度浮點數，一個為整數，它們的結果也不同。



```

C:\Program Files\Xinox Software\JCreator\J3\J3E\GE2001.exe
這是浮點數2.222
這是整數3

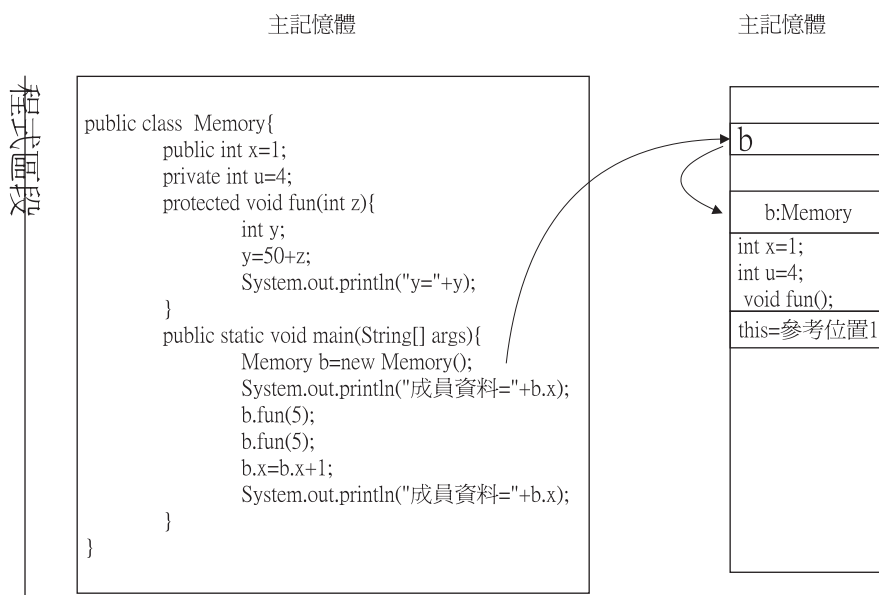
```

6-5 成員變數與區域變數

類別物件有成員變數，當新增類別物件時，就會分配給成員變數記憶體空間。當物件被 Garbage Collector 回收時，類別物件的成員變數就會結束。區域變數則是在函數呼叫時，新增的區域變數，當函數結束時，區域變數也就結束了。

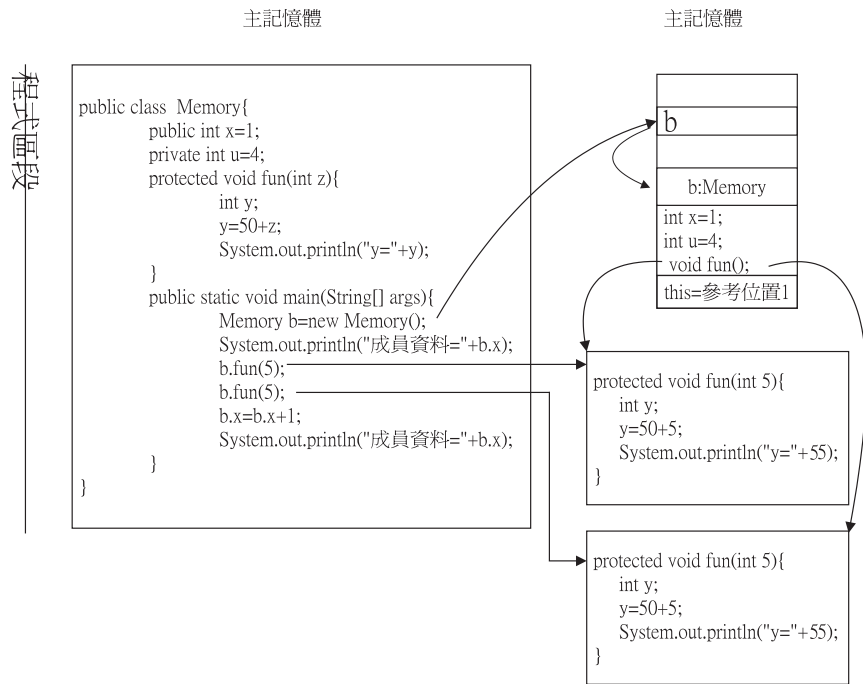
類別變數則是指被宣告成 static 的類別變數，當類別被編譯成 class 檔時，該靜態變數就存在了。

我們使用 javac 來編譯 Memory.java 成為位元組檔 bytecode。這時我們使用 java 來直譯 Memory.class 的檔案，它就會將 Memory 類別載入到我們的主記憶體，等待 CPU 來執行。而執行是由 main() 函數開始。這時在記憶體會新增 Memory 物件，並且分配給 b 變數參考。這時，型態為 Memory 的變數參考就會參考到 Memory 類別物件。b 變數參考的 Memory 類別物件就有初始化的整數型態 x 為 1，初始化的整數型態 u 為 4，還有 fun() 函數。

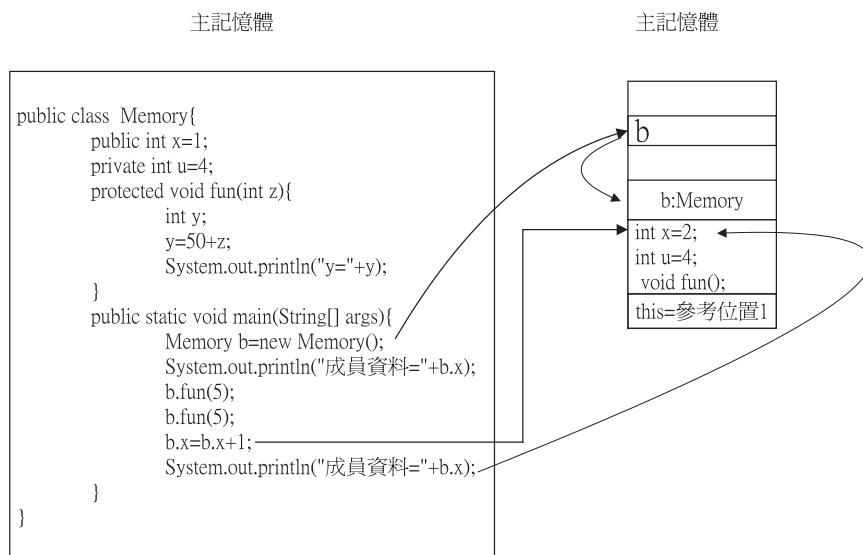




執行完 Memory 類別物件的 b.fun() 函數後，區域變數 z 和區域變數 y 就會消失。



這是修改類別的物件成員變數 x，由 x 加 1 成爲 2。當 main() 函數結束時，這個物件被 Java 的 Garbage Collector 回收時，變數 x 才會消失。



範例 Memory.java

第一行到第十七行為Memory類別。

第二行定義Memory類別的成員變數為整數型態，並且分配初始值1給它。

第四行到第八行為fun()函數。

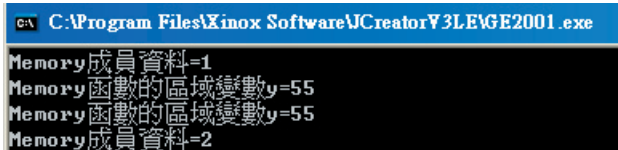
第四行的z變數和第五行的y變數都為區域變數，當呼叫完fun()函數時，該變數就會消失。

第十二行呼叫b物件的fun(5)函數，並且將5帶入。

第十四行將b物件的成員變數x加1，再分配給b物件的x欄位屬性。

```
1 public class Memory{
2     public int x=1;
3     private int u=4;
4     protected void fun(int z){
5         int y;
6         y=50+z;
7         System.out.println("Memory函數的區域變數y="+y);
8     }
9     public static void main(String[] args){
10        Memory b=new Memory();
11        System.out.println("Memory成員資料="+b.x);
12        b.fun(5);
13        b.fun(5);
14        b.x=b.x+1;
15        System.out.println("Memory成員資料="+b.x);
16    }
17 }
```

區域變數的值在該區域結束後就會消失。物件欄位屬性的值，要在物件消失時才會消失。



```
C:\Program Files\Xinox Software\JCreator\Y3LE\GE2001.exe
Memory成員資料=1
Memory函數的區域變數y=55
Memory函數的區域變數y=55
Memory成員資料=2
```



習題

1. 請說明物件導向的特色？

【答案】

物件導向程式設計(object oriented programming簡稱OOP)有幾個特色，分別為類別class、繼承inheritance、多型polymorphism。

2. 請以UML的圖型畫出一個類別？
3. 請簡述建構子？
4. 何謂封裝？
5. 請說明存取控制修飾子？
6. 請說明過載Overloading？

【答案】

我們也可以在同一個類別來宣告多個相同名稱的函數，這些都叫Overloading過載。相同名稱的函數，但是他們所帶入的參數不相同，Java編譯器將以函數的簽名(函數名稱和參數)來決定使用哪個函數，這就是方法的過載method overloading。

7. 請說明成員變數與區域變數？
8. 請畫圖說明成員變數與區域變數生成的過程？