

# 深入記憶體

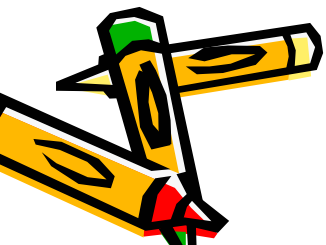
第十章

Java2全方位學習



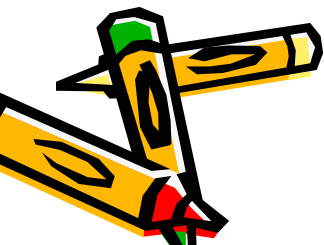
# 大綱

- 宣告
- 陣列
- 變數的存取範圍
- 參數的傳遞
- 記憶體回收



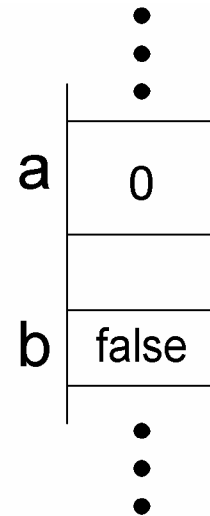
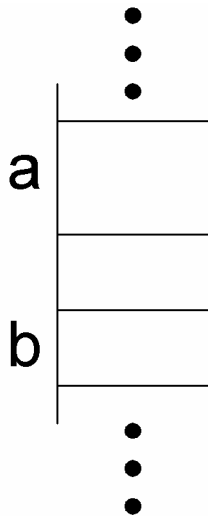
# 宣告

- 基本資料型態變數
- 物件變數



# 基本資料型態變數

- 初始值設定
  - 成員變數有預設值
  - 非成員變數需指定
    - variable i might not have been initialized

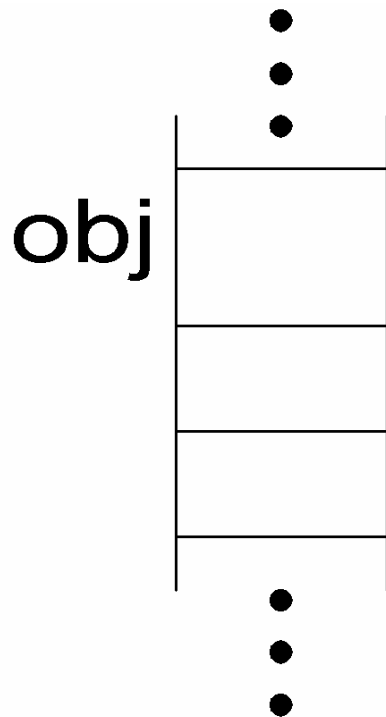


# 物件變數

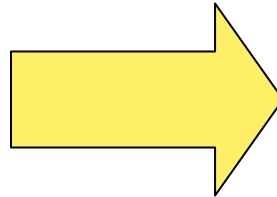
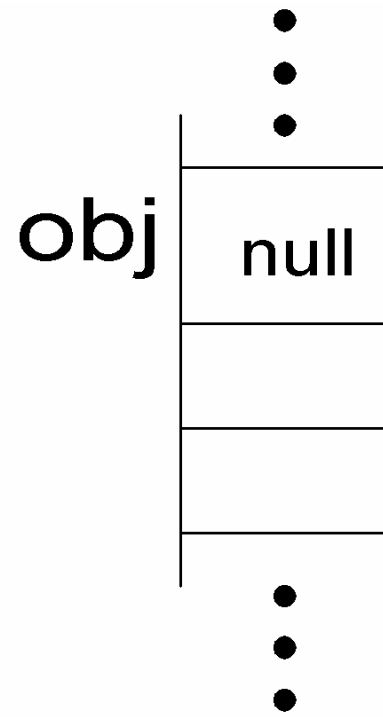
- 宣告和產生
  - `MyObject obj;`
    - 物件的參考
  - `obj = new  
MyObject(5);`
- 初始值
  - `null`
  - `NullPointerException`

# 物件變數宣告

MyObject obj;



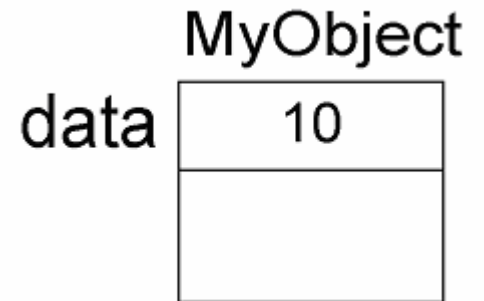
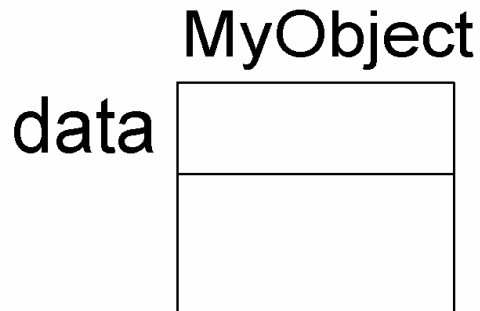
初始值設定



# 物件實體產生

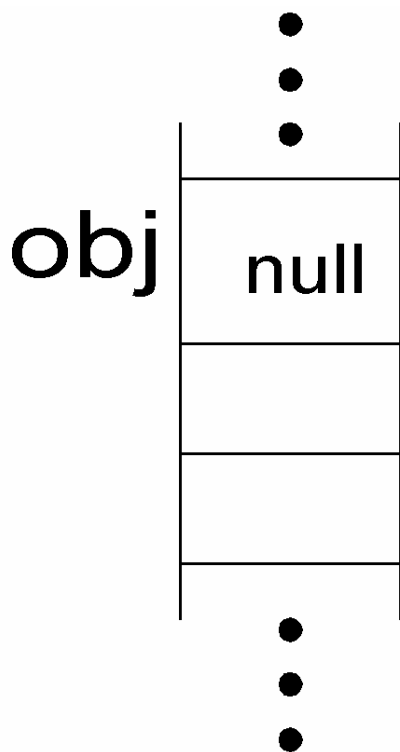
```
new MyObject(10);
```

初始值設定

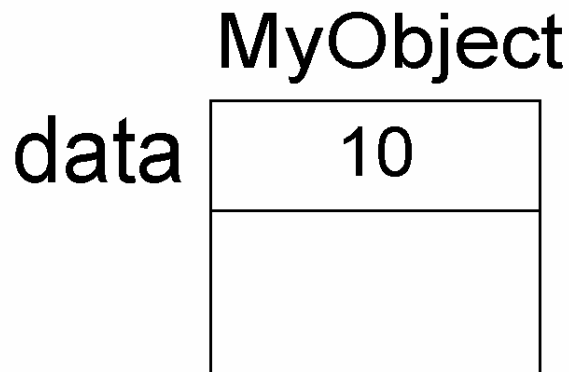


# 物件的參考與實體

物件的參考

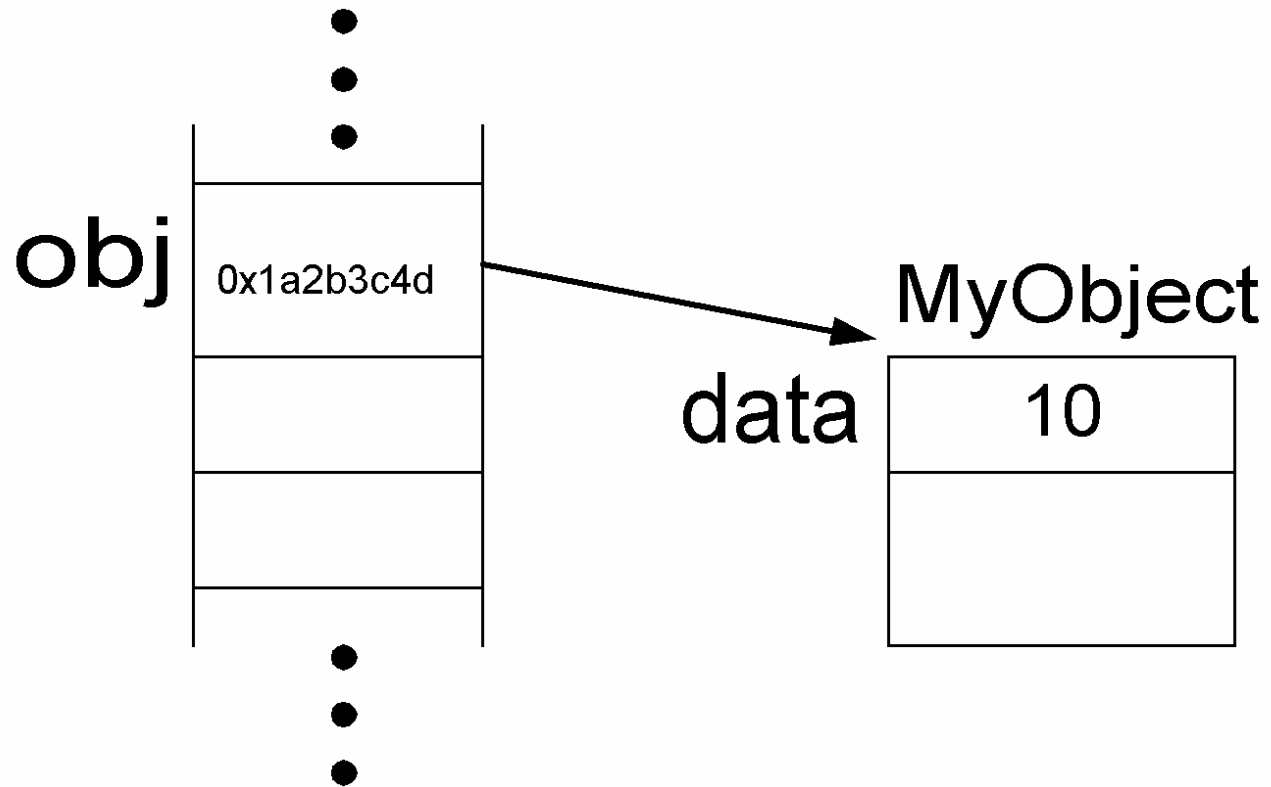


物件的實體



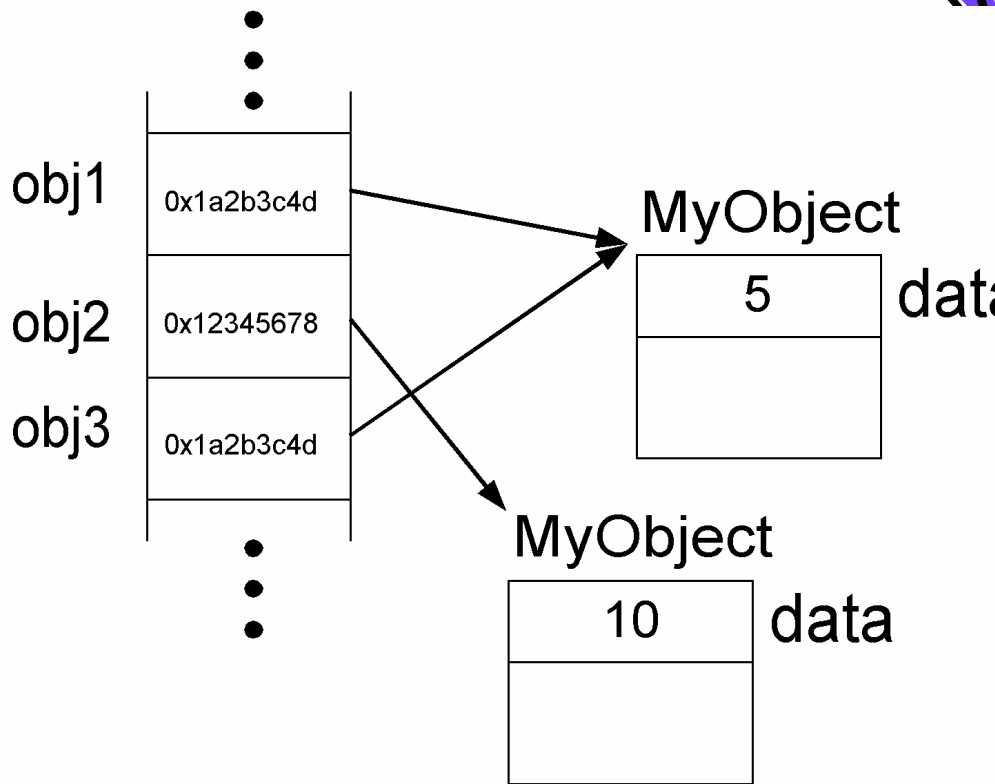


# 指向實體的參考



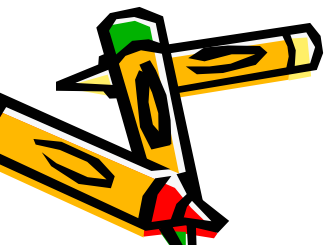
# 物件變數的宣告、產生與指定

```
MyObject obj1;  
MyObject obj2;  
MyObject obj3;  
obj1 = new MyObject(5);  
obj2 = new MyObject(10);  
obj3 = obj1;
```



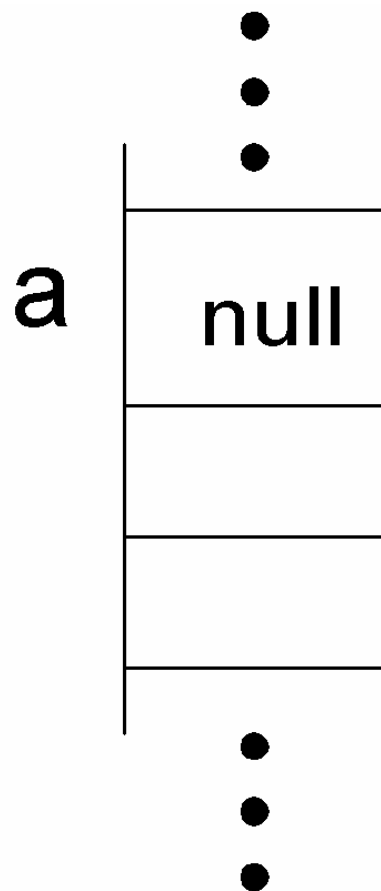
# 陣列

- 一組相同資料型態的資料集合
  - 基本資料型態
  - 物件型態
- 陣列本身是個物件



# 陣列的宣告

- <資料型態> 陣列名稱[];
  - int a[ ];
- <資料型態>[] 陣列名稱;
  - int[ ] a;
- 初始值
  - null



# 陣列實體產生

- new陳述式

- int a[ ] = new int[5];

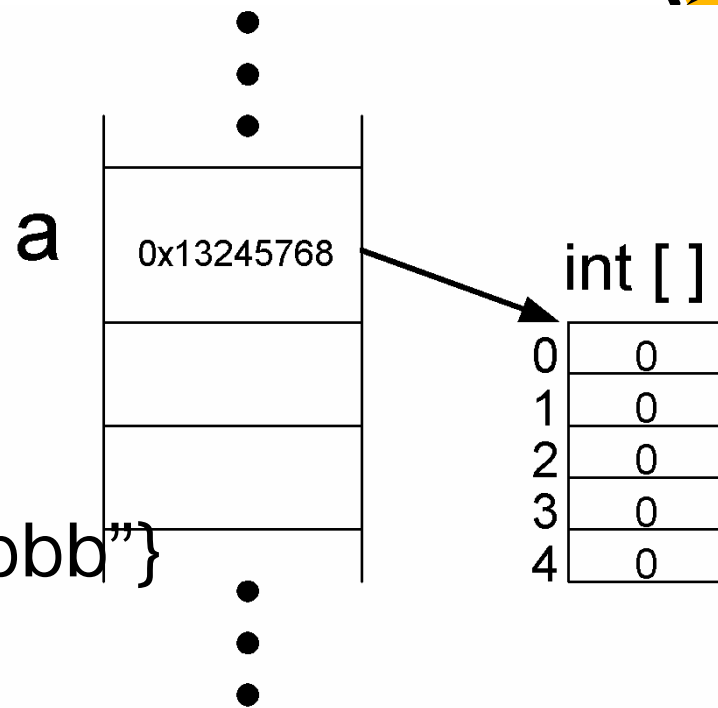
- 直接初始值設定

- int a[ ] = {1, 2, 3, 4, 5};

- String s[ ] = {"AAA", "bbb"}

- length屬性

- 取得陣列的大小



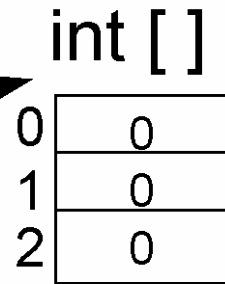
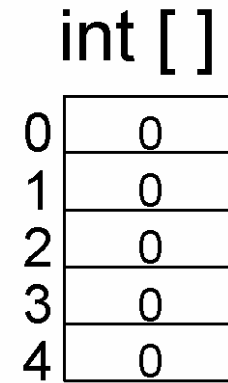
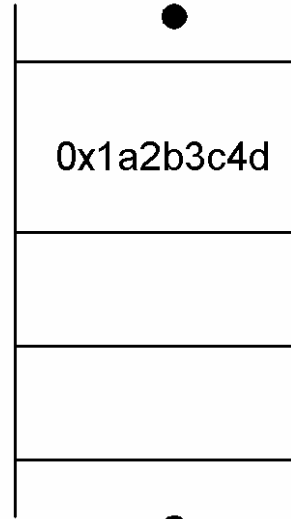
# 陣列資料的存取

- 索引值(index)
  - 從0開始
  - `ArrayIndexOutOfBoundsException`
- 指定
  - `a[3] = 10;`
- 使用
  - `b = a[2]+a[3];`

# 改變陣列的大小

```
int a[] = new int[5];  
a = new int[3];
```

**a**



# 陣列的複製

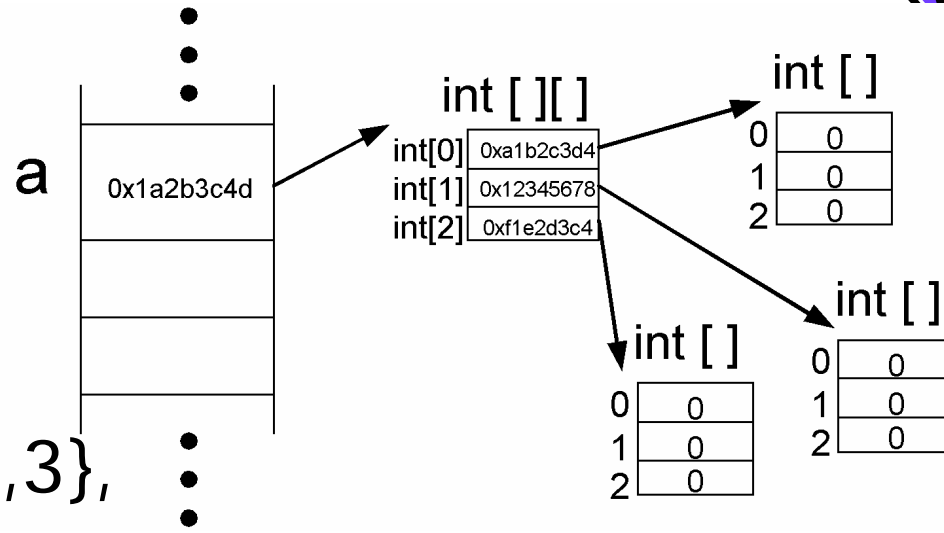
- 迴圈
- System.arraycopy方法
  - 來源陣列、來源起點、目的陣列、目的起點、複製資料長度
  - 產生例外
    - NullPointerException
    - ArrayIndexOutOfBoundsException
    - ArrayStoreException



# 多維陣列

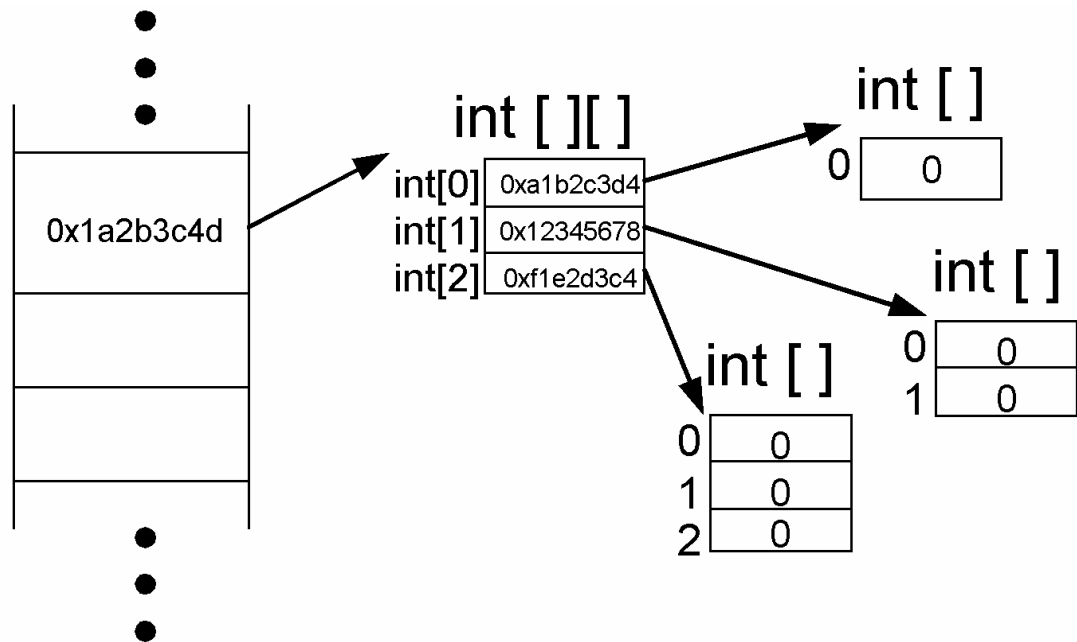
- 多個索引值
- 二維陣列

```
- int a[ ][ ] = new  
  int[3][3];  
- int a[ ][ ] = { {1, 2, 3},  
                  {2, 4, 6},  
                  {1, 3, 5}};
```



# 不規則多維陣列

```
int a[][] = new  
    int[3][];  
a[0] = new int[1];  
a[1] = new int[2];  
a[2] = new int[3];
```



# 物件陣列

- 陣列宣告完後還需產生物件的實體

```
MyObject obj[ ] = new MyObject[3];
```

```
obj[0] = new MyObject(1);
```

```
obj[1] = new MyObject(5);
```

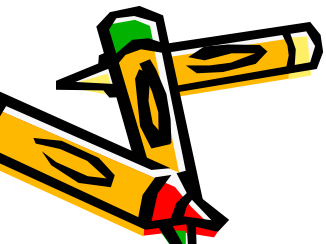
```
obj[2] = new MyObject(10);
```

# 變數的存取範圍(scope)

- 類別等級
- 物件等級
- 方法等級
- 區塊等級

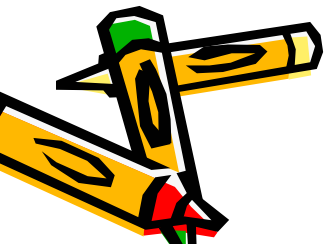
# 類別等級

- 類別成員
  - 直接透過類別名稱來取
- static
- 在同一個類別中，屬於全域變數
  - 類別中所有的方法都可以直接存取



# 物件等級

- 物件成員
  - 透過物件來存取
- 在同一個物件中，屬於全域變數
  - 物件中所有的方法都可以直接存取



# 方法等級

- 在方法中所宣告的變數
- 區域變數
  - 只在這個方法中使用，別的方法無法存取
- 用this關鍵字存取物件成員

# 區塊等級

- 在區塊中宣告的變數
  - for、if、switch...
  - {...}



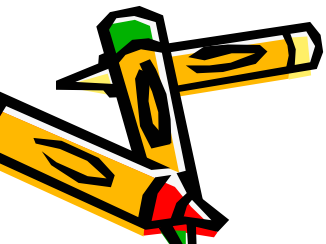
# 使用優先權

- 同名變數
- 優先使用自己範圍內的變數



# 參數的傳遞

- 傳值(call by value)
- 值
  - 基本資料型態
    - 變數的內容
  - 物件型態
    - 物件實體的參考(記憶體位址)



# 記憶體回收

- Garbage Collection機制
  - Garbage Collector(GC)
- JVM負責管理記憶體的使用
  - 配置
  - 回收不用的記憶體

# GC的啟動

- `System.gc()`
- `Runtime.getRuntime().gc()`
- 只是建議JVM去啟動GC，做記憶體的回收，但系統不一定會馬上處理這項要求
- 回收物件的順序依其產生的次序

# finalize方法

- 當物件要被回收之前，GC會呼叫該物件的finalize方法
- 釋放所佔用的資源

# Q&A

