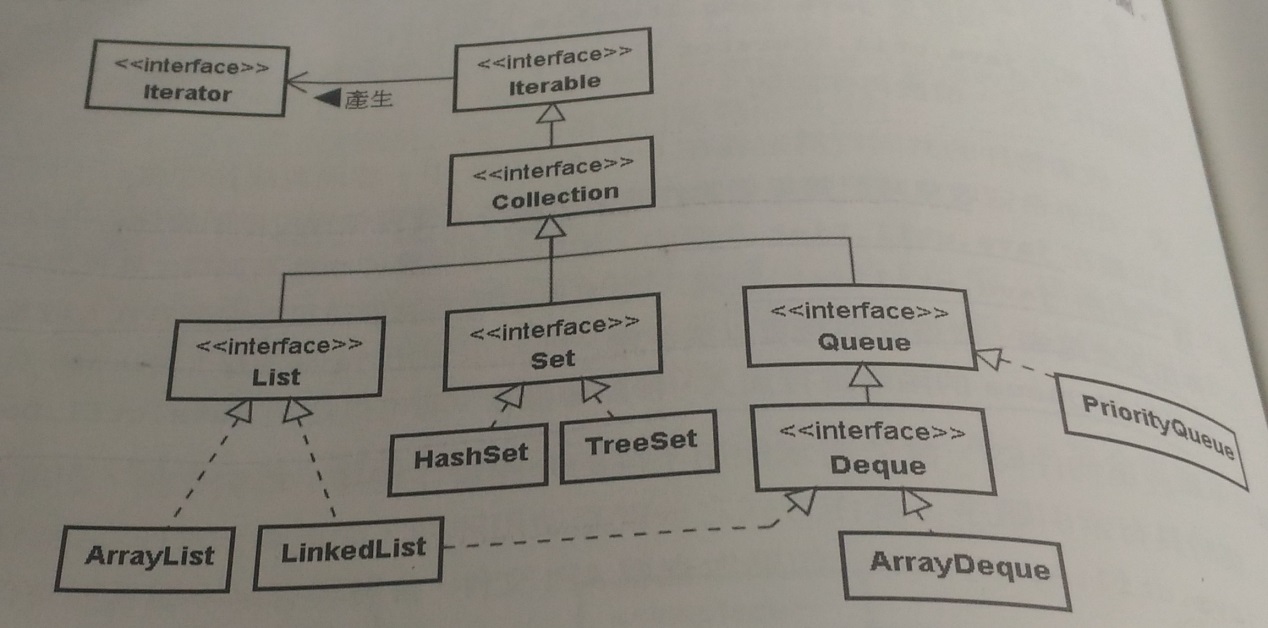
**Collection與Map**

程式中有收集物件需求，可以收集物件方法有使用Object陣列。然而java提供了數種收集物件的類別。像是新增物件add()、移除物件remove()方法都定義在java.util.Collection。



**具有索引的List**

以索引方式保留收集的物件順序，List介面定義了add()、remove()、set()等依索引操作的方法，有ArrayList、LinkList實作它。

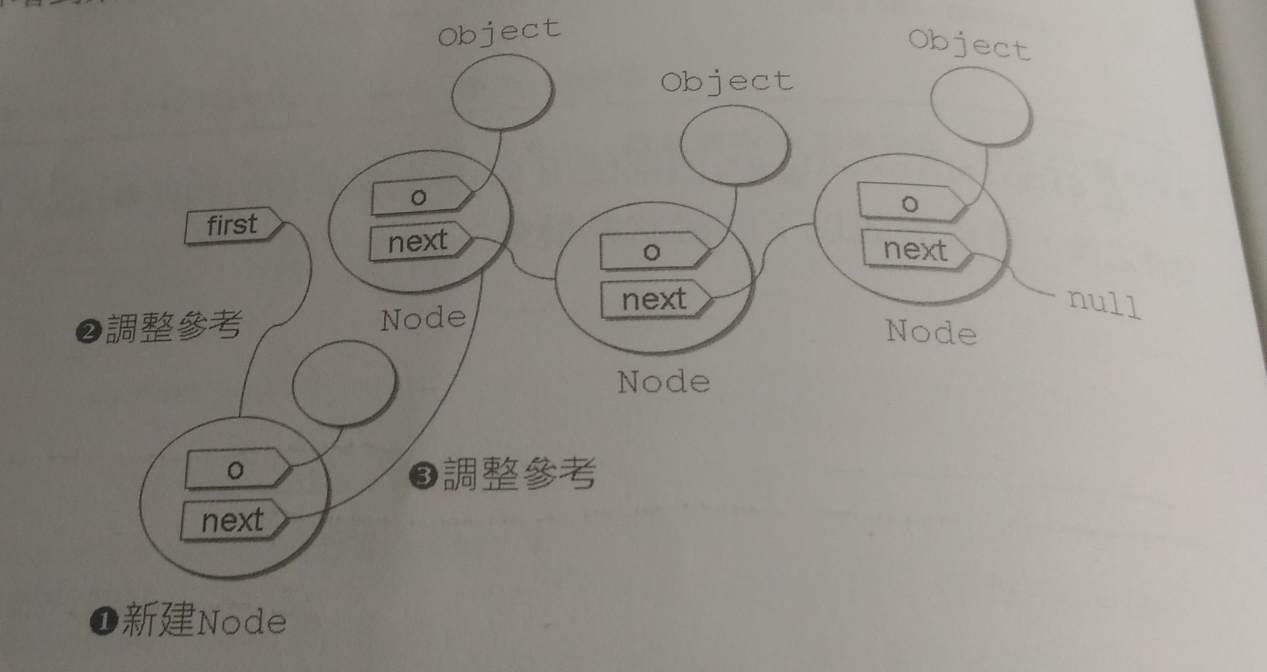
**ArrayList特性**

內部使用object陣列保存物件，根據索引隨機存取速度快，但調整索引順序時則效率較差(將物件加至索引0，原本的物件都得後調整到1、2…..)。

ArrayList內部陣列長度不夠時，會建立新陣列將舊陣列參考指定給新陣列，所以耗時和記憶體。ArrayList可指定容量的建構式。節省耗費的成本。

**LinkedList特性**

採用**鏈結**結構， 每次add物件時會建立新的node保存物件，有利調整 索引順序，但要指定隨機索引物件效率較差(從第一個索引開始找)。



**內容不重複的Set**

收集物件時，若有相同物件則不重負收集物件，其HashSet實作概念原理為，在記憶體中開設雜湊桶的空間，如果物件要加入HashSet，會呼叫物件的hashCode()取得雜湊碼，並嘗試放入對應的號碼的雜湊桶內，裡面沒物件直接放入，有物件則呼叫物件的equals()進行比較。

**Java在判斷物件是否重複都會呼叫hashcode()，equals方法，兩方法必須實作。**

**支援佇列操作的Queue**

收集物件時可以佇列方式，結構為先進後出。

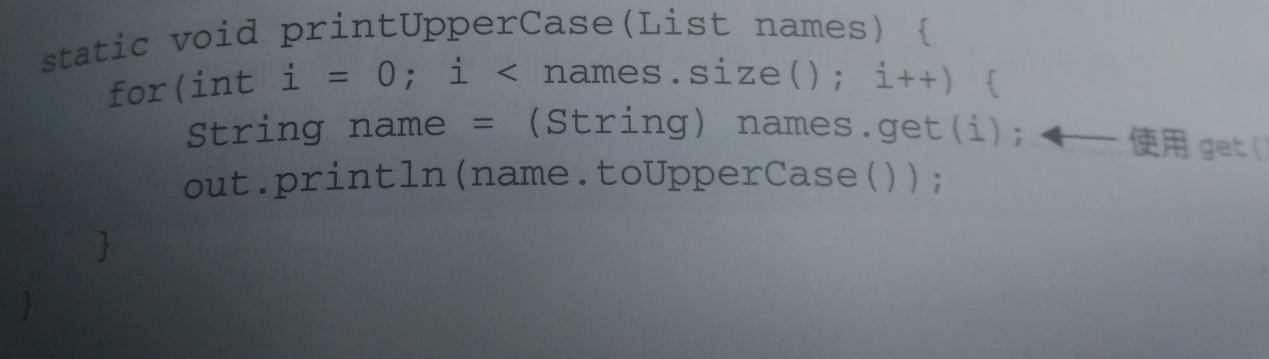
**泛型**

Collection在收集物件時，內部實作採用object參考收集之物件(因為考慮收集各種物件)，所以取回物件時必須自行記住物件真正的型態，在語法上告訴編譯器讓物件重新扮演自己的型態。

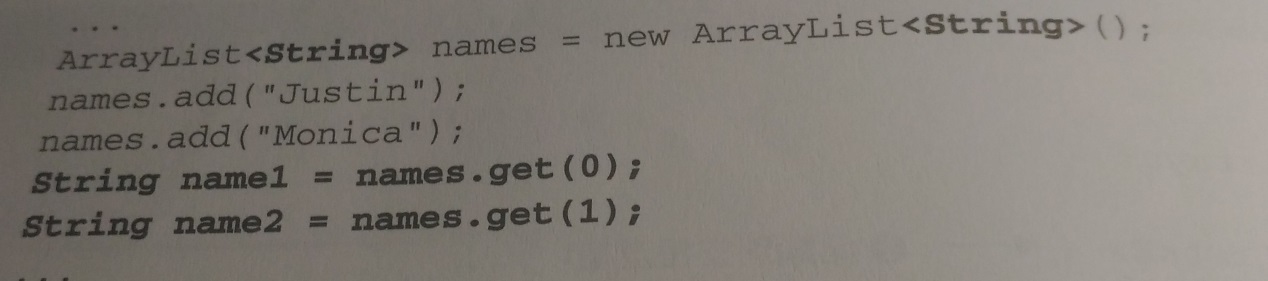
泛型，可以在宣告與建立物件時用角括號告知編譯器型態參數的實際型態，

泛型可用來指定類別與方法。

下圖為取回值還須自行記得物件型態。並讓物件扮演原本的型態



下圖為使用泛型後



**Iterable與Iterator**

無論是List、Set、Queue都有Iterator()方法，jdk5前定義在collection介面。Jdk5後提升至java.lang.Iterable父介面。可使用Iterator的hasnext()和next()查看有無物件和取得物件。

**Comparable與Comparator**

Collection提供sort()方法做排序，但其要求排序的物件必須實作comparable介面，或是建立一個實作Comparator的物件告知sort排序方法。

**鍵值對應的Map**

根據鍵值來取得相對應的值，map也支援泛型，因為判斷鍵是否重複根據hashCode()與equals()所以鍵的物件必須實作hashCode()與equals()。

HashMap建立鍵值對應後，鍵是無順序。

TreeMap鍵的部分將會排序，但作為鍵的物件必須是實作Comparable，或建構TreeMap時指定實作Comparator的物件。

**使用Properties**

繼承HashTable，所以也實作map介面，擁有map行為，一樣可以用setProperties指定字串型態的鍵值，getProperties指定字串型態的鍵，取回字串型態的值。

可以用load()從檔案中讀取屬性。

在使用java指令時可以使用-D指定系統屬性。

**走遍map鍵值**

Map沒有繼承Iterable，可以用map的keySet()傳回的Set物件與values()傳回的collection物件去取得所有鍵與值。