**5/12　認識Collection家族**

**sheep programming class**

**資訊學院資傳學習家族**

1. **查詢Java API**

**進入Java API後，先選擇你要查詢的Package**

**（例如：要查LinkedList，就找java.util）**

**然後再選擇要找class或是interface**

**再來就可以開始查詢你要查的，像是建構值、函數、常數……**

**［２］類別與介面**

**Java不允許多重繼承**

**（Ｏ） （Ｘ）**

**class A**

**class B**

**class C**

**class A**

**class B**

**class C**

**但是Java的Interface提供多重繼承**

**繼承使用方式有extends和implements**

**只需要注意是類別繼承介面還是介面繼承介面即可，虛線表示繼承interface**

**< interface >**

**A**

**class B**

**implements**

**extends**

**< interface >**

**A**

**< interface >**

**B**

**［３］Auto Boxing與Auto Unboxing**

**sheep programming class**

**資訊學院資傳學習家族**

**Java提供Compiler Sugar，會自動將變數裝箱與拆箱。**

**自動裝箱範例：**

**Stack<Integer> st = new Stack<Integer> ();**

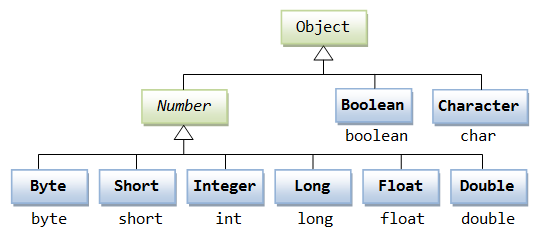
**st.add(new Integer(12)); //正規的寫法**

**st.add(12); //自動裝箱後的寫法**

**自動拆箱範例：**

**LinkedList list = new LinkedList (Arrays.asList(0,1,2,3));**

**list.get(2); //自動拆箱的寫法**

**以下的類別都是變數的物件，皆可自動拆箱與自動裝箱**

**［４］認識演算法類別**

**sheep programming class**

**資訊學院資傳學習家族**

**Java提供兩個實用的演算法類別，分別是Arrays與Collections**

**Arrays：與陣列有關的演算法**

**例如fill可以讓陣列初始化的時候更方便，還有Arrays.sort()方便排序**

**int[] n = {0,1,4,2,3};**

**Arrays.sort(n);**

**Collections：與Java Collections Framework相關的**

**例如Collections.swap()可以交換２個元素，Collections.sort()方便排序**

**ArrayList<Integer> list = new ArrayList (Arrays.asList(0,1,2,3));**

**Collections.sort(list);**

**［５］複習資料結構**

**Array**

**支援RandomAccess，所以存取資料的速度很快**

**但是宣告完陣列後，分配的空間是固定不可改變的**

**List <E>**

**一般來說，List的實用性會比Array還好用，並且有自帶很多方法可用**

**但是串列的優缺點，視子類別的特性，性能與實用性皆會不同**

**ArrayList： ArrayList支援陣列RandomAccess的特性**

**sheep programming class**

**資訊學院資傳學習家族**

**儲存資料採同步性質，當空間不夠時，會動態增加50%**

**LinkedList： 鏈結串列在add和remove元素的時候，性能表現較好**

**但在insert（java是set）和get的時候效能較差**

**Vector： 支援同步而安全，但是由於多了一些東西**

**所以對於效能上並沒有非常快，於是一般都是用List較多**

**Stack <E>**

**堆疊繼承了List的特性，也繼承的vector的同步性質**

**特色是像堆東西一樣，每次只能動用最上層的元素**

**先進後出（First In Last Out） 後進先出（Last In First Out）**

**Queue <E>**

**像排隊一樣，進去只能從後面出去，出去只能從前面出去**

**Java並沒有實現Queue，但是底下的子類別提供更好的類別可用**

**像是Double End Queue（雙向佇列）可彈性動用前後的元素**

**先進先出（First In First Out） 後進後出（Last In Last Out）**

**PriorityQueue： 優先佇列可以讓排隊中的元素，依照他們的權重優先出隊**

**比較方式可以使用自訂的Comparator修改**

**Double End Queue：雙向佇列在Java可以使用ArrayDeque**

**隊伍的頭跟尾皆可以任意對調**

**Set <E>**

**sheep programming class**

**資訊學院資傳學習家族**

**中文稱為集，他的資料結構是讓所有元素不重複，獨一無二的**

**HashSet ： 雜湊集可以快速的以雜湊資料結構儲存元素**

**因為重點在於放唯一元素，所以不需要用到高級的資料結構存**

**TreeSet： 樹狀集儲存元素的方式是以紅黑樹的樹狀資料結構儲存元素**

**每次放元素的同時，都會將資料排序（可自寫Comparator）**

**Map <K , V>**

**中文稱為映射，每一個Key都會對應一個Value，Key都是獨一無二的**

**K是Key的縮寫，表示鍵**

**V是Value的縮寫，表示映射值**

**HashMap： 雜湊映射可以快速的以雜湊資料結構儲存元素**

**主要是鍵需要不同，映射值多個一樣是不影響的**

**TreeMap： 樹狀映射儲存的方式是以紅黑樹的樹狀資料結構儲存元素**

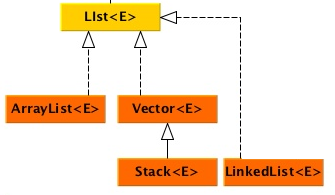
**每次放元素的同時，都會根據鍵來將資料排序**

**（可自寫Comparator）**

**［５］List與Stack的繼承關係**

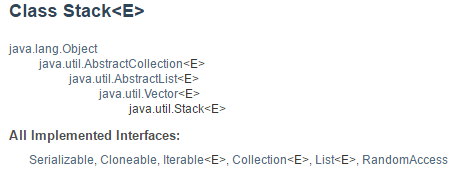
**sheep programming class**

**資訊學院資傳學習家族**

****

**ArrayList、LinkedList與Stack都繼承了List的特性**

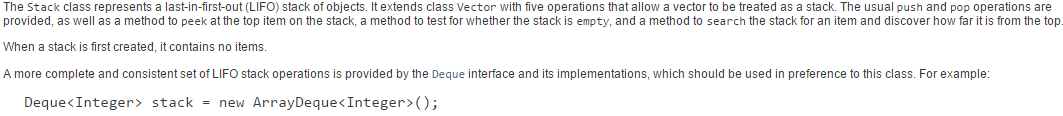
**下面以Stack為例，可以在API中查詢到所有繼承關係：**

****

**另外也可以參考裡面的補充，可以發現Deque也提供堆疊的方法可用：**

**Deque<Integer> stack = new ArrayDeque<Integer>();**

**也就是說，Deque不僅是雙向佇列，也提供push與pop的方法可提供使用**

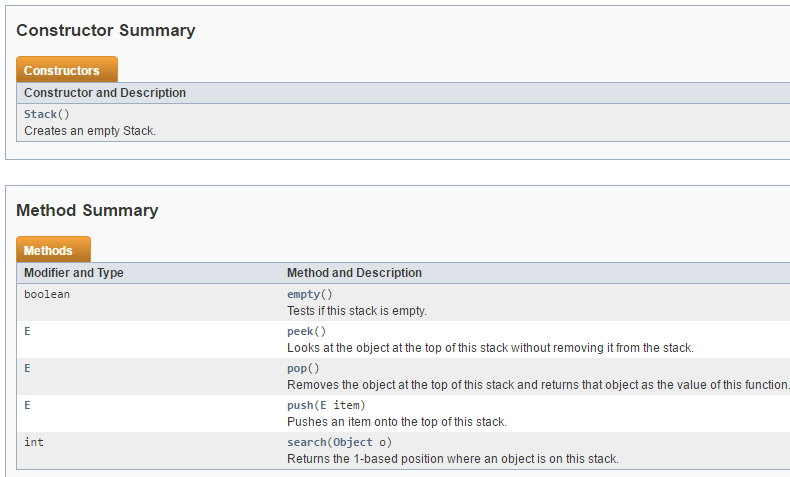
**非常的方便！！**

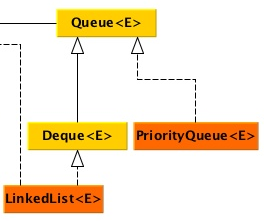
**接下來也可以查詢有哪些建構值可以使用**

**sheep programming class**

**資訊學院資傳學習家族**

**還有提供那些函數提供使用**

**［６］Queue與Priority Queue的繼承關係**

****

**ArrayDeque<E>**

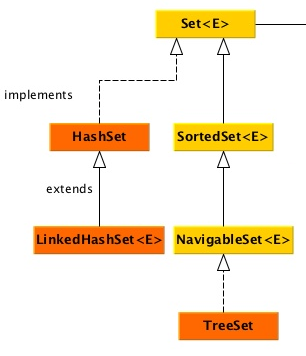
**LinkedList和ArrayDeque都繼承了Deque**

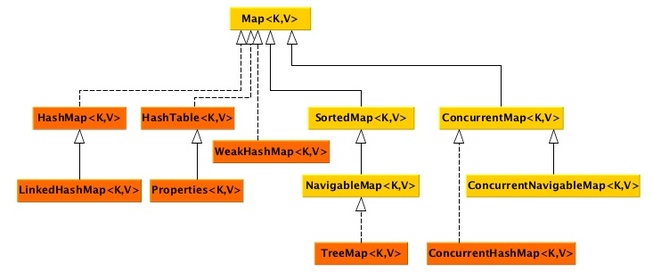
**PriorityQueue與上面兩個類別皆繼承了Queue**

**［７］Set的繼承關係**

**sheep programming class**

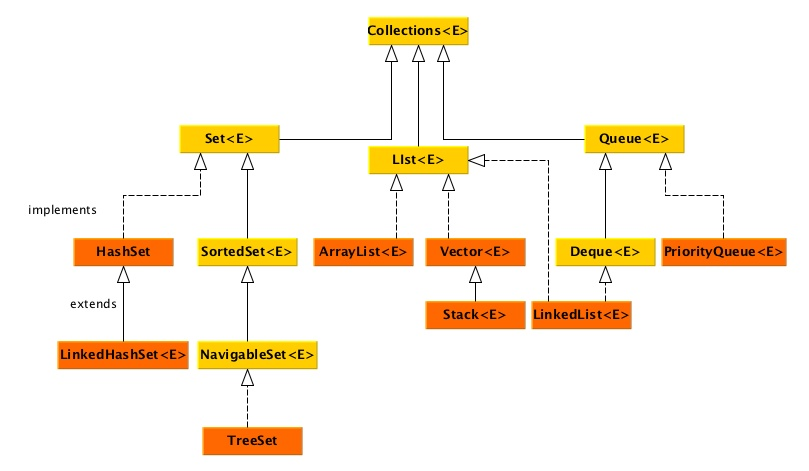
**資訊學院資傳學習家族**

****

**［８］Map與HashTable的繼承關係**

**Map並不是Collection的成員，比較常用到的是HashMap與TreeMap**

**HashTable與HashMap用法差不多，資料結構有些不同，且只能儲存字符**

**［９］總結Java Collections Framework的繼承關係**

**sheep programming class**

**資訊學院資傳學習家族**

**ArrayDeque<E>**

**［１０］如何應用資料結構？**

**以Stack來說，可以用來判斷一個算式的括弧是否合乎規定**

**校內程式檢定參考題目： ＜簡單＞　SP9 計算左/右小括號的個數**

**＜稍難＞　SP56 圓括號對應**

**Stack也能計算後序運算式**

**例如： 12+ = 3**