

工具類別(2)

TYIC 桃高資訊社

列舉

列舉(**enumerate**)，顧名思義，就是把東西列出來

在 **Java** 中，**列舉**就是一個特殊的**類別**：

1. **列舉類別**為**不可繼承類別**，也不可以**繼承類別**，但可**實作介面**
2. 其中的**常數**就是**公開靜態不可變欄位**，為該**列舉類別**的**實例**
3. **列舉類別**的**建構子**為 **private**，外界不可實例化

```
修飾子 enum 列舉類別名稱 { 常數1, 常數2, ..., 常數n}
```

java

```
修飾子 enum 列舉類別名稱 {  
    常數1(引數...), 常數2(引數...), ..., 常數n(引數...);
```

欄位...

方法...

建構子...

類別...

```
}
```

java

列舉

列舉若實作介面
可以等到創建實例
才覆寫方法

列舉類別的
公開靜態方法

Role values()

可以返回該

列舉類別的常數

組成的陣列

```
public class Main {
    public static void main(String[] args) {
        for (Role role : Role.values()) {
            new Person(role).printInfo().eat();
        }
    }
}

enum Role implements Eat {
    WORKER("上班族") {
        @Override
        public void eat() {
            System.out.println("吃土");
        }
    }, BABY("嬰兒") {
        @Override
        public void eat() {
            System.out.println("喝奶");
        }
    }, STUDENT("學生") {
        @Override
        public void eat() {
            System.out.println("叫外送");
        }
    };

    final String description;

    Role(String description) {
        this.description = description;
    }
}
```

```
interface Eat {
    void eat();
}
```

```
class Person {
    Role role;
```

```
    Person(Role role) {
        this.role = role;
    }
```

```
    Person printInfo() {
        System.out.println(
            role.description +
            " : " + switch (role) {
                case WORKER -> "上班";
                case BABY -> "哭";
                case STUDENT -> "上課";
            });
        return this;
    }
```

```
    void eat() {
        role.eat();
    }
}
```



上班族：上班
吃土
嬰兒：哭
喝奶
學生：上課
叫外送

output

java

Collection

工具類別常常大量的使用泛型，因為其可以讓使用者減少型別檢查、轉型的動作
最常見的泛型類別就是是集合框架中的所有類別，集合框架主要分為兩個介面：

`java.util.Collection<E>` 和 `java.util.Map<K, V>`

`Collection<E>` 代表一組相同型別的物件放在一起，下方為部分動態方法：

`int size()`、`boolean isEmpty()`、`void clear()`

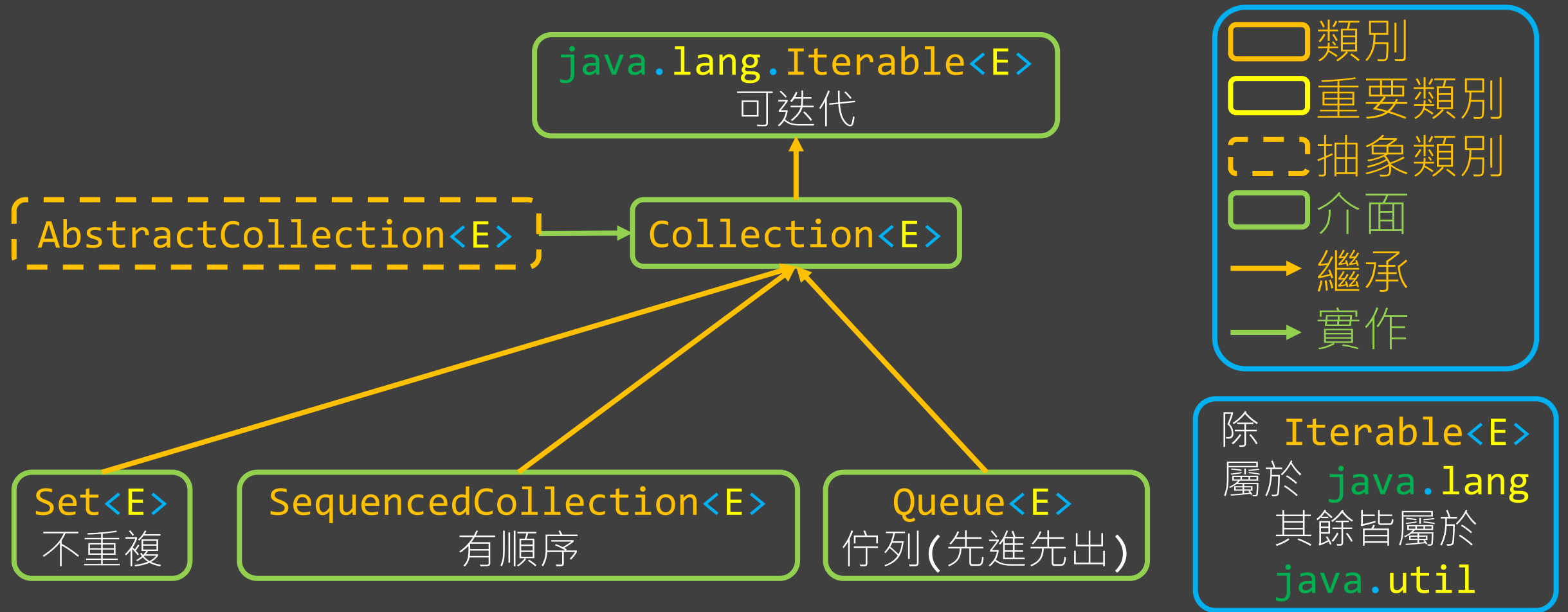
`boolean add(E e)`、`boolean addAll(Collection<? extends E> c)`

`boolean remove(Object o)`、`boolean removeAll(Collection<?> c)`

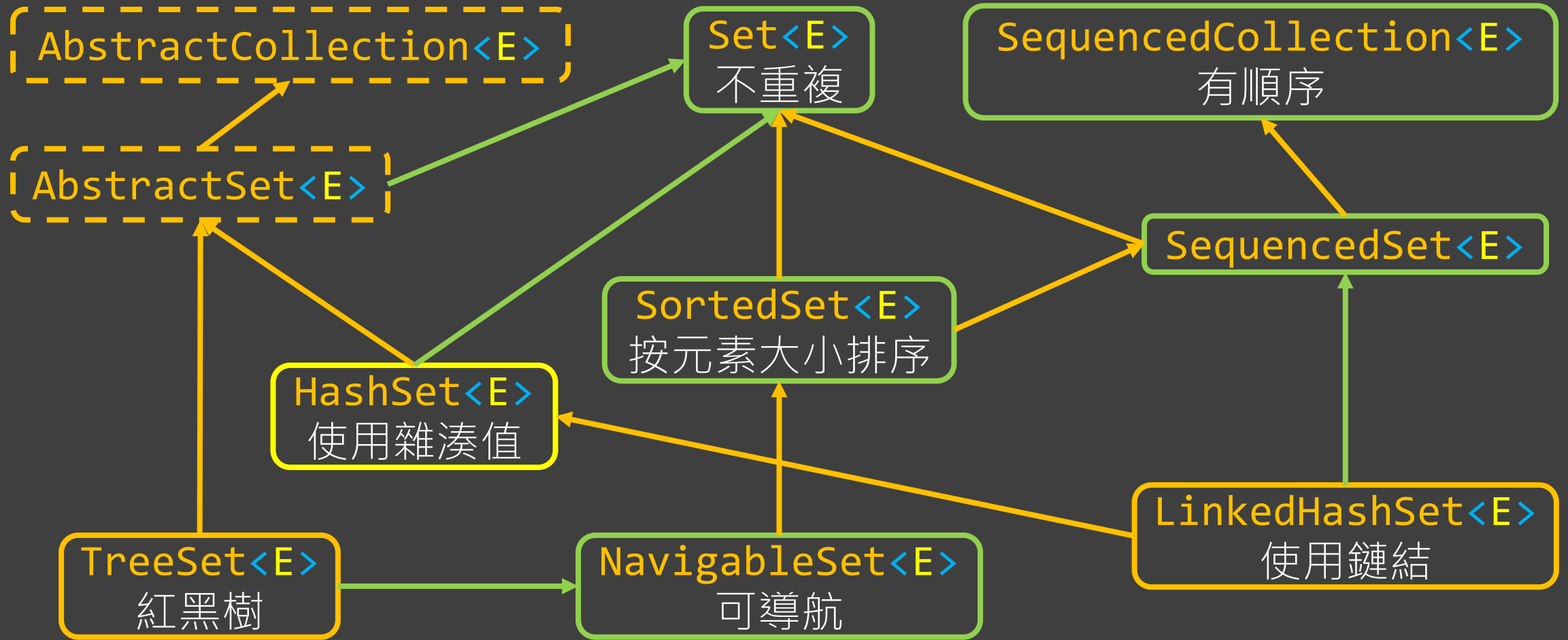
`boolean contains(Object o)`、`boolean containsAll(Collection<?> c)`

`<T> T[] toArray(T[] a)`、`Iterator<E> iterator()`

Collection



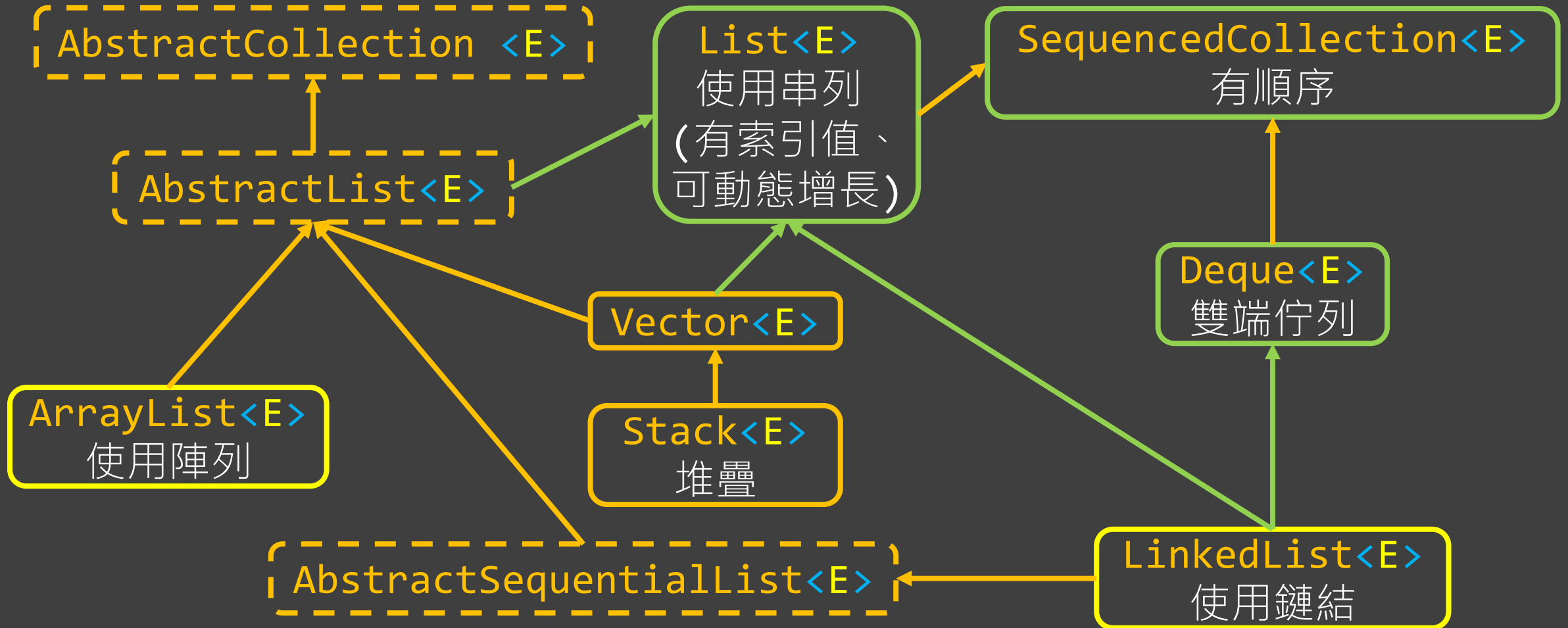
Set



HashSet

Set<E> 的實作類別儲存的元素不可重複，即數學上的「集合」
其中最重要的實作類別就是 **HashSet<E>**

List



Queue

