

套件 & 修飾字

Package & Modifier



Java Fundamental



Outline

- ◆ 套件
- ◆ 修飾字



Outline

◆ 套件

◆ 修飾字



Package

- ◆ 套件(Package)其實就是檔案系統中的目錄，如果要使用套件，首先會建立專案套件，然後在專案套件內依功能建立子套件，最後在各個子套件內建立Java檔，編譯之後會產生對應的類別檔。



- ◆ 使用套件功能時，需要使用package關鍵字再加上該類別檔所在的「內部路徑」，所謂內部路徑就是專案套件至該類別檔所在目錄的路徑。

Book.java

```
1. //標示第 3 行 Book 類別在 books 專案套件內
2. package books;
3. public class Book{
4.     public void show(){
5.         System.out.println("Book 物件資訊");
6.     }
7. }
```



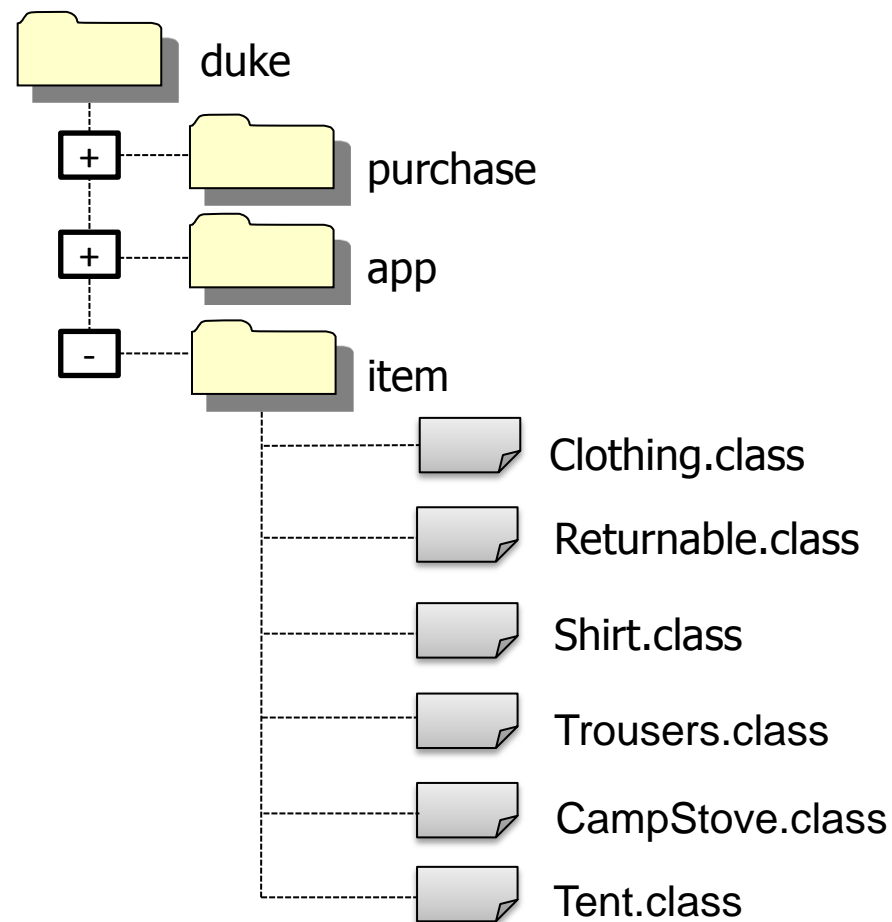
套件(package)的使用

◆ 目的

- 將許多不同的類別檔案實際分門別類放在一起
- 使類別名稱不相衝突
- 提供package(default)的存取保護

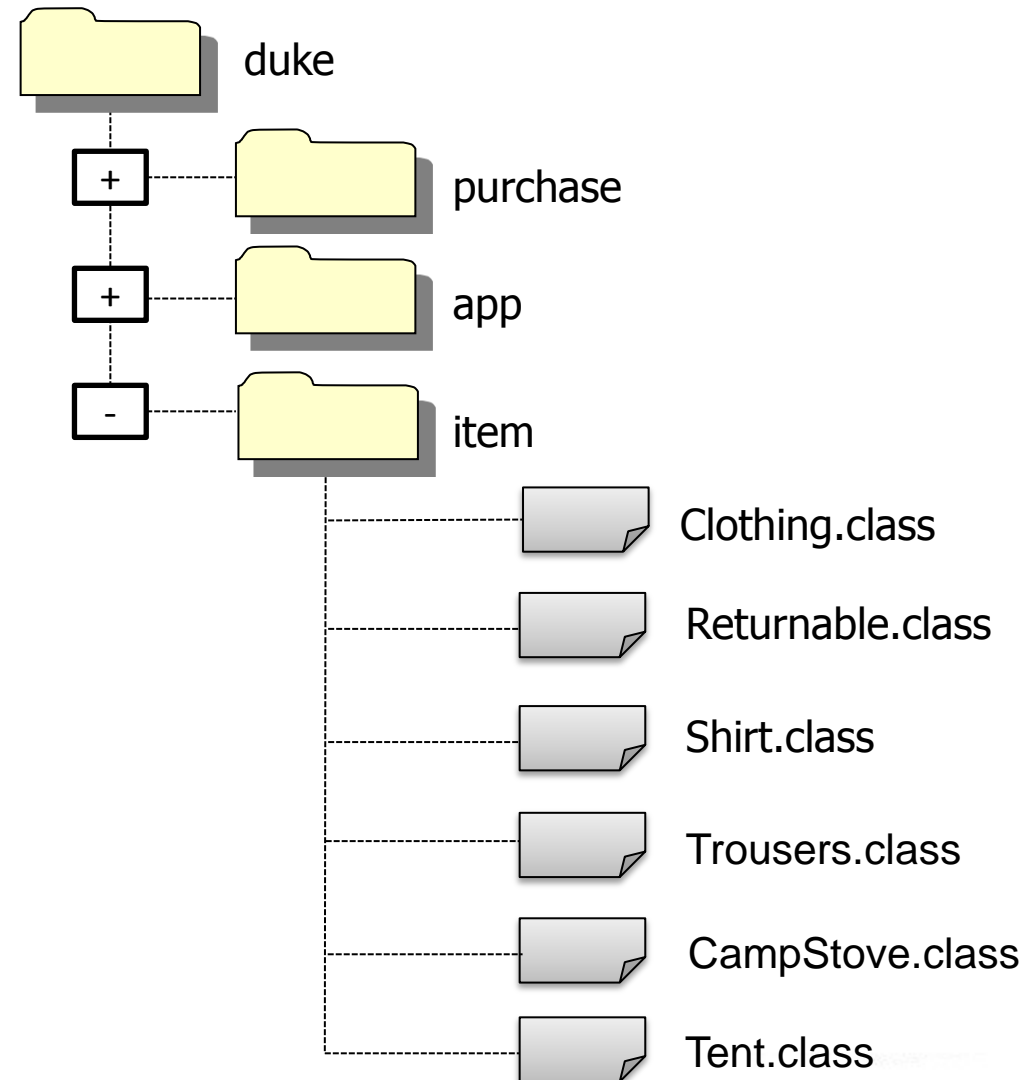
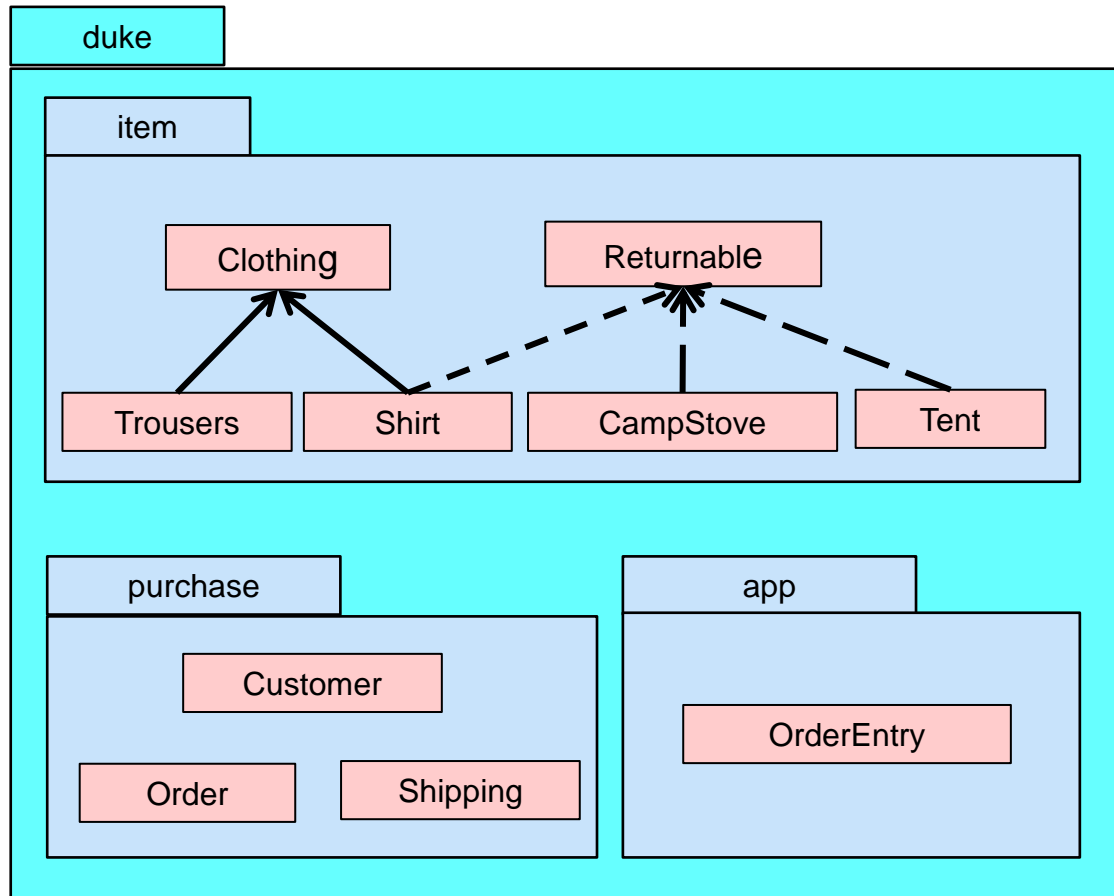
◆ 實體型態

- 同電腦之檔案目錄系統





Package 套件設計





Package 陳述式

◆ Package 宣告

- `package <package名稱>[.<subPackage名稱>]*`
- 一個 Java 原始檔中只能有一個套件的宣告
- 套件宣告一定要在第一行
- 沒有套件宣告的類別則屬於預設套件 (default package) , 就是目前目錄

◆ Package 下編譯與執行

- 切換至application根目錄
- 編譯 `javac <package 路徑>\類別名稱.java`
- 執行 `java <package 路徑>.<類別名稱>`



Package 類別存取

◆ Package 宣告

- 同package之下的類別：可直接用短名存取 (類別名稱本身)
- 使用其他package下的類別：以類別的完整長名存取 (套件名稱~... . 類別名稱)

◆ 引入(import)package

- `import packageName.className;`
- `import packageName.*;`

◆ 預設自動import

- `java.lang.*`
- `default package`(目前工作目錄)



類別的完整長名存取

- ◆ Order 類別存取 duke.purchase.Order及duke.item.Shirt 類別

```
01 package duke.purchase;  
02 public class Order {  
03     .....  
04     .....  
05 }
```

```
01 package duke.item;  
02 public class Shirt {  
03     .....  
04     .....  
05 }
```

```
01 package duke.app;  
02 public class OrderEntry1 {  
03     public static void main(String[] args) {  
04         duke.purchase.Order order = new duke.purchase.Order();  
05         duke.item.Shirt shirt = new duke.item.Shirt();  
06         order.addShirt(shirt);  
07         .....  
08     }  
09 }
```



引入(import) package

◆ import 敘述告知JVM：所使用class之可能全名

```
01 package duke.purchase;  
02 public class Order {  
03     .....  
04     .....  
05 }
```

```
01 package duke.item;  
02 public class Shirt {  
03     .....  
04     .....  
05 }
```

```
01 package duke.app;  
02 import duke.purchase.*;  
03 import duke.item.Shirt;  
04 public class OrderEntry2 {  
05     public static void main(String[] args) {  
06         Order order = new Order();  
07         Shirt shirt = new Shirt();  
08         order.addShirt(shirt);  
09         .....  
10     }  
}
```



引入(import) package

◆ import 關鍵字的使用

- 一個類別使用到其他類別內容，如果它們都在不同的套件內，就要使用import關鍵字來標示被匯入類別的內部路徑與該類別名稱；
- 如果在相同套件內，就import語法則可省略。

Execute.java

```
1. package books.execute;
2. //因為第 7、9 行使用到 Book、ComputerBook 類別，所以必須使用 import 來匯入
3. import books.Book; //也可改成「import books.*;」，代表 books 內任一類別
4. import books.computer.ComputerBook;
5. public class Execute {
6.     public static void main(String[] args){
7.         Book book = new Book();
8.         book.show();
9.         ComputerBook cBook = new ComputerBook();
10.        cBook.show();
11.    }
12. }
```



import 注意事項

◆ package、import、class 出現順序

```
01 package duke.purchase;  
02 import java.util.ArrayList;  
03 import java.io.*;  
04 public class Order {  
05     .....  
06 }
```

◆ 避免名稱混淆，同一個短名,在classpath中可找到兩個不同的長名定義時,仍需用類別長名來存取

```
01 import java.util.*;  
02 import java.sql.*;  
03 public class TestAmbiguous {  
04     public static void main(String[] args){  
05         Date d = new Date();  
06         java.util.Date d = new java.util.Date();  
07     }  
08 }
```

```
系統管理員: 命令提示字元  
c:\JavaClass>javac TestAmbiguous.java  
TestAmbiguous.java:5: error: reference to Date is ambiguous, both class java.sql  
.Date in java.sql and class java.util.Date in java.util match  
    Date d = new Date();  
    ^  
TestAmbiguous.java:5: error: reference to Date is ambiguous, both class java.sql  
.Date in java.sql and class java.util.Date in java.util match  
    Date d = new Date();  
    ^  
2 errors  
c:\JavaClass>
```



Outline

- ◆ 套件
- ◆ 修飾字



存取權限修飾字

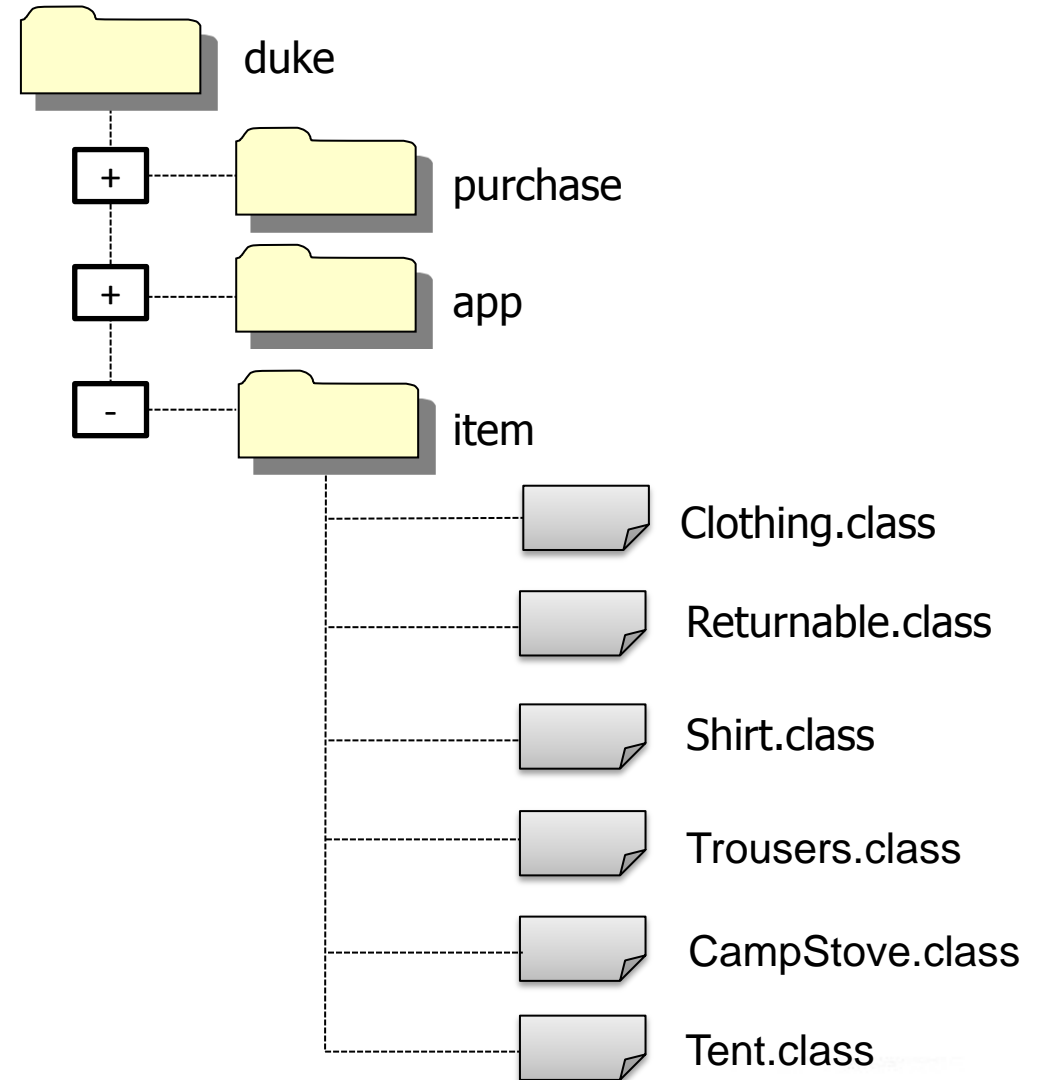
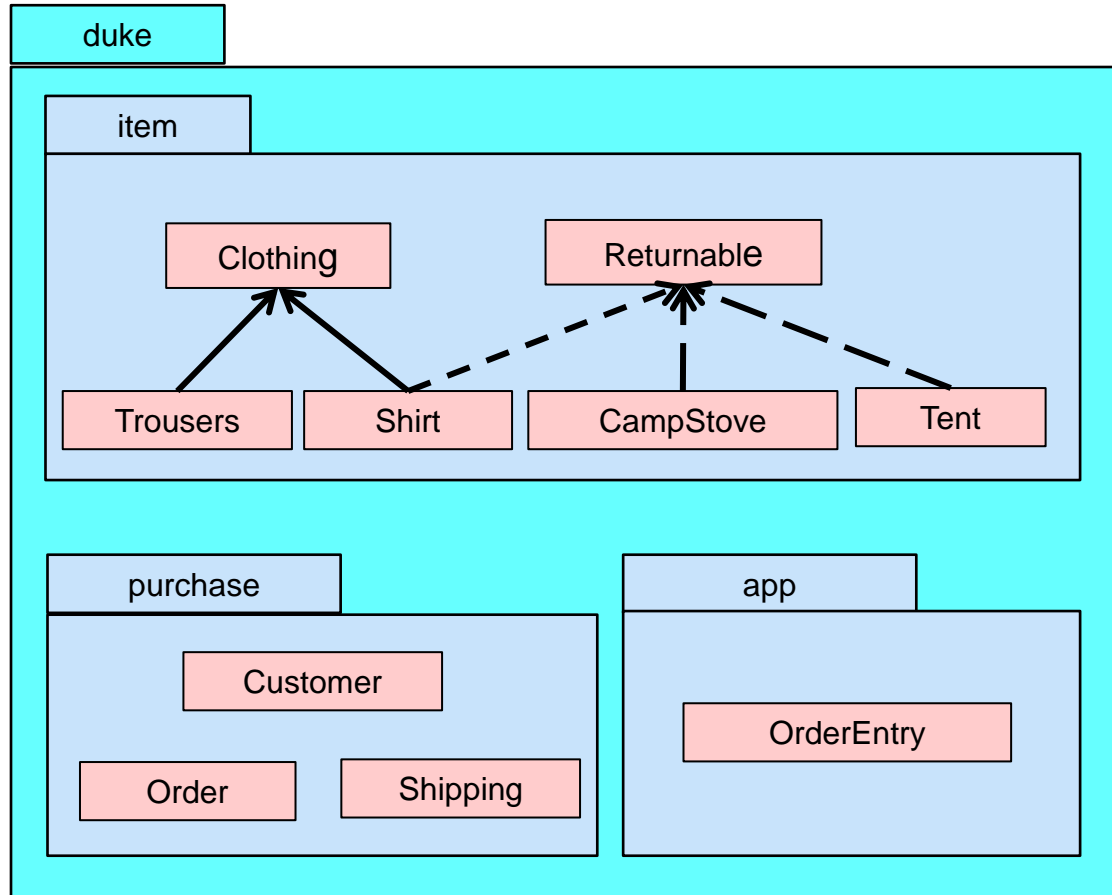
◆ 存取權限修飾字(Modifier)是用來宣告類別、屬性與方法(含建構子)可被存取的權限,分成四個等級：

- 屬性與方法：4種修飾字皆可以使用。
- 類別(class)：只能使用public與default(無修飾字)(內部類別除外)

存取權限修飾字	權限說明
private	同一個 class 才可存取
default(無修飾字)	同一個 package 的class 才可存取
protected	同一個 package 的class才可存取 不同package 的要有繼承關係才可存取
public	皆可存取



存取權限修飾字





存取權限修飾字

◆ 存取權限修飾字的可視範圍：

存取權限修飾字	同一Class中	同一Package中	子類別	全域 (不同Package)
private	Yes	-	-	-
Default (無修飾字)	Yes	Yes	-	-
protected	Yes	Yes	Yes	-
public	Yes	Yes	Yes	Yes

Q & A