CH14 練習

鄭安翔
ansel_cheng@hotmail.com

練習1 汽車集合

請依據下列描述，試撰寫出程式碼：

1. 請輸入車商收購的汽車品牌，以Set 類別儲存車商擁有的品牌。以List儲存買入車子，依購買順序存放0~N車庫位置， 車商收購三個品牌後不再收購。
2. 輸入完畢後，印出車商擁有的品牌及車子儲存位置資料。
3. 顧客尋找一指定汽車品牌，顯示是否有此品牌車輛及第一台該品牌車輛車庫位置。
4. 顧客購買該車後於List中刪除指定車輛，若此品牌車已無庫存，於Set中刪除指定品牌。
5. 輸入Quit 結束購買
6. 印出目前車商擁有的品牌。
7. 印出目前車庫位置中所有車輛。

練習1 汽車集合

```
Console
<terminated> CarTest [Java Application] C:\Program Files
輸入汽車品牌:Audi
新增品牌:Audi
輸入汽車品牌:BMW
新增品牌:BMW
輸入汽車品牌:Audi
現有品牌:Audi
輸入汽車品牌:BMW
現有品牌:BMW
輸入汽車品牌:Toyota
新增品牌:Toyota
銷售品牌:[Audi, BMW, Toyota]
現有車輛:[Audi, BMW, Audi, BMW, Toyota]
輸入欲購買品牌:Mazda
未銷售Mazda
輸入欲購買品牌:BMW
請至1號車庫賞車
BMW已銷售
輸入欲購買品牌:Toyota
請至3號車庫賞車
Toyota已銷售
輸入欲購買品牌:Quit
銷售品牌:[Audi, BMW]
現有車輛:[Audi, Audi, BMW]
```

CarTest 類別

CarTest.java

```java
package com.car;

import java.util.*;

public class CarTest {
    static Set brands = new TreeSet();
    static List garage = new LinkedList();

    private static void printData() {
        System.out.println("銷售品牌:"+brands);
        System.out.println("現有車輛:"+garage);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while(brands.size()<3) {
            System.out.print("輸入汽車品牌:");
            String car = sc.nextLine();
            garage.add(car);
            boolean newBrand = brands.add(car);
            if(newBrand)
                System.out.println("新增品牌:"+car);
            else
                System.out.println("現有品牌:"+car);
        }
        printData();

```

CarTest 類別

```
28          System.out.print("輸入欲購買品牌:");
29          String carWanted = sc.nextLine();
30          while (!carWanted.equalsIgnoreCase("Quit") || garage.isEmpty()) {
31              if(brands.contains(carWanted)) {
32                  int idx = garage.indexOf(carWanted);
33                  System.out.println("請至"+idx+"號車庫賞車");
34                  garage.remove(idx);
35                  if(!garage.contains(carWanted))
36                      brands.remove(carWanted);
37                  System.out.println(carWanted+"已銷售");
38              } else {
39                  System.out.println("未銷售"+carWanted);
40              }
41              System.out.print("輸入欲購買品牌:");
42              carWanted = sc.nextLine();
43          }
44          printData();
45      }
46
47  }
48
```

練習2 使用HashMaps 統計Part Numbers出現次數

1. 開啟 GenericsPractice專案
2. 檢視ProductCounter類別的
   - main()方法兩個區域變數
     - String[] parts

     以part number紀錄之產品銷售資訊,
     - Map<String, String> productNames

     產品描述與編號的對應關係如右圖
3. ProductCounter中宣告2個屬性
   - Map<String, String> products
     - 與main()中區域變數相同
   - Map<String, Integer> counts
     - parts中產品銷售次數,以產品編號做鍵值
     - 使用 HashMap<>() 建構

Description	Part Number
Blue Polo Shirt	1S01
Black Polo Shirt	1S02
Red Ball Cap	1H01
Duke Mug	1M02

ProductCounter.java

```java
package com.example.generics;

import java.util.HashMap;

public class ProductCounter {
    // Create a Counting Map
    // Create a Name Mapping Map

    public static void main(String[] args) {

        // List of part data
        String[] parts = new String[]{"1S01", "1S01", "1S01",

        // Create Product Name Part Number map
        Map<String, String> productNames = new TreeMap<>();
        productNames.put("Blue Polo Shirt", "1S01");
        productNames.put("Black Polo Shirt", "1S02");
        productNames.put("Red Ball Cap", "1H01");
        productNames.put("Duke Mug   ", "1M02");

        // Create Product Counter Object and process data
    }
```

```java
25
26      public ProductCounter(Map productNames) {
27          // Your code here
28      }
29
30      public void processList(String[] list) {
31          // your code here
32      }
33
34      public void printReport() {
35          // Your code here
36      }
37  }
```

4. 建立傳入Map的建構式
   - 設定productNames屬性
5. 建立processList(String[] list)方法
   - 傳入parts產品銷售資訊
   - 以產品編號做鍵值, 銷售數量為值之鍵值對置於counts中
     - 若產品編號為Map中已存在之鍵值, 取得其值+1後將鍵值對放回Map中
     - 產品編號不存在Map中, Map中新增一個以產品編號為鍵,值為1的鍵值對
6. 建立printReport()方法
   - 列印產品名稱及銷售數量
7. 測試、執行

```
Console
<terminated> ProductCounter [Java Applic
============銷售報告============
Red Ball Cap        銷售   5個
Duke Mug            銷售   3個
Blue Polo Shirt     銷售   7個
Black Polo Shirt    銷售   6個
```

ProductCounter類別

ProductCounter.java

```java
package com.example.generics;

import java.util.HashMap;

public class ProductCounter {
    // Create a Counting Map
    // Create a Name Mapping Map

    public static void main(String[] args) {

        // List of part data
        String[] parts = new String[]{"1S01", "1S01", "1S01",

        // Create Product Name Pa
        Map<String, String> produ
        productNames.put("Blue Po
        productNames.put("Black P
        productNames.put("Red Bal
        productNames.put("Duke Mu

        // Create Product Counter
    }

```

```java
public class ProductCounter {
    private Map<String, String> products;
    private Map<String, Integer> counts;

    public static void main(String[] args) {
        // List of part data
        String[] parts = new String[]{"1S01", "1S01", "1S01", "1S01",

        // Create Product Name Part Number map
        Map<String, String> productNames = new HashMap<>();
        productNames.put("Blue Polo Shirt", "1S01");
        productNames.put("Black Polo Shirt", "1S02");
        productNames.put("Red Ball Cap", "1H01");
        productNames.put("Duke Mug   ", "1M02");

        ProductCounter pc = new ProductCounter(productNames);
        pc.processList(parts);
        pc.printReport();
    }

```

ProductCounter類別

```java
25
26      public ProductCounter(Map productNames) {
27          // Your code here
28      }
29
30      public void processList(String[] list) {
31          // your code here
32      }
33
34      public void printReport() {
35          // Your code here
36      }
37  }
```

```java
27      public ProductCounter(Map productNames) {
28          this.products = productNames;
29          counts = new HashMap<>();
30      }
31
32      public void processList(String[] list) {
33          for(String item:list) {
34              if(counts.containsKey(item)) {
35                  int count = counts.get(item);
36                  counts.put(item, ++count);
37              } else
38                  counts.put(item, 1);
39          }
40      }
41
42      public void printReport() {
43          System.out.println("------------銷售報告------------");
44          for(String name: products.keySet()) {
45              String key = products.get(name);
46              int count = counts.get(key);
47              System.out.printf("%-20s銷售%3d個%n", name, count);
48          }
49      }
50  }
51
```

練習3 EmployeeDAO

1. 修改 EmployeeDAO專案
2. 新增com.example.dao.EmployeeDAOMapImpl類別
   - 實作com.example.dao.EmployeeDAO介面
   - 用 SortedMap<Integer, Employee>來儲存員工資料
   - 修改介面實作方法
     - add(emp : Employee) 方法
     - update(emp : Employee) 方法
     - delete(id : int) 方法
     - findById(id : int) : Employee 方法
     - getAllEmployees(): Employee[ ] 方法
     - close()方法

練習3 EmployeeDAO

3. 修改EmployeeDAOFactory
   - 修改 createEmployeeDAO() : EmployeeDAO 方法
     - 建立EmployeeDAOMapImpl 物件傳回
4. 刪除EmployeeDAOMemoryImpl.java
5. 測試、執行
   - 員工數量可動態增加, 沒有大小限制
   - 建立員工時, 可以使用大於10的員工編號

EmployeeDAOCollectionImpl 類別

EmployeeDAOMapImpl.java

```java
package com.example.dao;

import java.util.*;
import com.example.model.Employee;

public class EmployeeDAOMapImpl implements EmployeeDAO {
    private SortedMap<Integer, Employee> employees = new TreeMap<>();

    @Override
    public void add(Employee emp) throws DAOException {
        int id = emp.getId();
        if(employees.containsKey(id))
            throw new DAOException(id+"號員工已存在,新增失敗!");
        employees.put(id, emp);
    }

    @Override
    public void update(Employee emp) throws DAOException {
        int id = emp.getId();
        if(!employees.containsKey(id))
            throw new DAOException(id+"號員工不存在,修改失敗!");
        employees.put(id, emp);
    }

```

EmployeeDAOCollectionImpl 類別

```
25      @Override
26      public void delete(int id) throws DAOException {
27          if(!employees.containsKey(id))
28              throw new DAOException(id+"號員工不存在,刪除失敗!");
29          employees.remove(id);
30      }
31
32      @Override
33      public Employee findById(int id) throws DAOException {
34          Employee emp = employees.get(id);
35          if(emp==null)
36              throw new DAOException(id+"號員工不存在!");
37          return emp;
38      }
39
40      @Override
41      public Employee[] getAllEmployees() throws DAOException {
42          return employees.values().toArray(new Employee[0]);
43      }
44
45      @Override
46      public void close() throws Exception {
47          System.out.println("關閉資源......");
48      }
49
50  }
51
```

EmployeeDAOFactory 類別

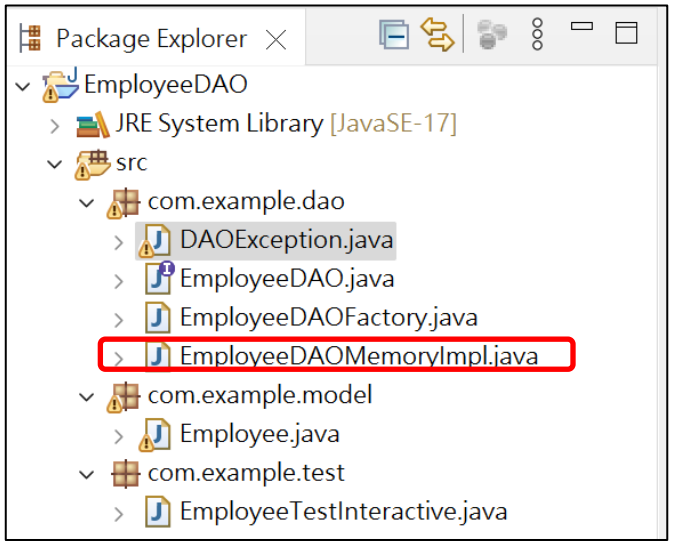
EmployeeDAOFactory.java

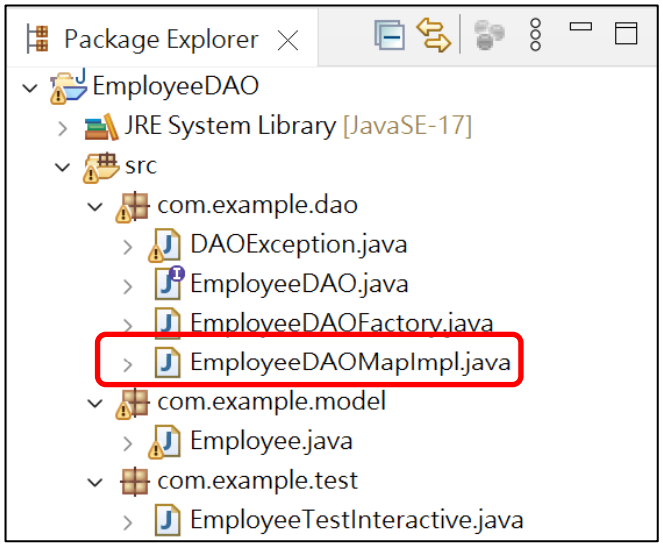
```java
package com.example.dao;

public class EmployeeDAOFactory {

    public EmployeeDAO createEmployeeDAO() {

        return new EmployeeDAOMemoryImpl();

    }

}
```

EmployeeDAOFactory.java

```java
package com.example.dao;

public class EmployeeDAOFactory {

    public EmployeeDAO createEmployeeDAO() {

        return new EmployeeDAOMapImpl();

    }

}
```

删除EmplyeeDAOMemoryImpl.java





測試、執行

```
Console
<terminated> EmployeeTestInteractive [Java Application] C:\Program Files\Java\jdk-17.0.4

[C]reate | [R]ead | [U]pdate | [D]elete | [L]ist | [Q]uit:
C
Enter int value for employee id:
101
Enter value for employee first name :
Sean
Enter value for employee last name :
Cheng
Enter value for employee birth date (MMM d, yyyy) :
Mar 21, 1974
Enter float value for employee salary :
50000
Successfully added Employee Record: 101


Created Employee ID:   101
Employee Name: Sean Cheng
Birth Date:    3月 21, 1974
Salary:        $50,000.00
```

```
[C]reate | [R]ead | [U]pdate | [D]elete | [L]ist | [Q]uit:
R
Enter int value for employee id:
101
Employee ID:   101
Employee Name: Sean Cheng
Birth Date:    3月 21, 1974
Salary:        $50,000.00



[C]reate | [R]ead | [U]pdate | [D]elete | [L]ist | [Q]uit:
U
Enter int value for employee id:
101
Modify the fields of Employee record: 101. Press return to accept current value.
Enter value for employee first name [Sean] :

Enter value for employee last name [Cheng] :

Enter value for employee birth date (MMM d, yyyy) [Mar 21, 1974] :

Enter float value for employee salary [$50,000.00] :
60000
Successfully updated Employee Record: 101
```

```
[C]reate | [R]ead | [U]pdate | [D]elete | [L]ist | [Q]uit:
L
Employee ID:   101
Employee Name: Sean Cheng
Birth Date:    3月 21, 1974
Salary:        $60,000.00



[C]reate | [R]ead | [U]pdate | [D]elete | [L]ist | [Q]uit:
D
Enter int value for employee id:
101
Deleted Employee 101


[C]reate | [R]ead | [U]pdate | [D]elete | [L]ist | [Q]uit:
L


[C]reate | [R]ead | [U]pdate | [D]elete | [L]ist | [Q]uit:
Q
```