

Ch9上課練習

鄭安翔

ansel_cheng@hotmail.com

練習一

1. 修改 **EmployeePractice** 專案
2. **EmployeeTest** 類別
 - ❑ 將各類型員工多型為 **Employee**
 - ❑ 檢視使用多型對專案會造成的影響
 - ❑ 各類別員工的資料顯示是否正常？
 - ❑ **Engineer**物件是否可以增加技能？

EmployeeTest 類別

```
EmployeeTest.java ×
1 package com.example;
2
3 import com.example.domain.*;
4
5 public class EmployeeTest {
6     public static void main(String[] args) {
7         Employee emp1 = new Employee("Sean", "A123456789", 50000);
8         Employee emp2 = new Admin("Amy", "B210987654", 70000);
9         Employee emp3 = new Engineer("David", "C109876543", 80000);
10        Employee emp4 = new Manager("Louis", "D124680135", 100000, "TW Sales");
11        Employee emp5 = new Director("Nicole", "R202468135", 120000, "Global Sales", 1000000);
12
13        // System.out.println("David 學會了Java, Android");
14        // emp3.addSkill("Java");
15        // emp3.addSkill("Android");
16
17        emp1.displayInformation();
18        emp2.displayInformation();
19        emp3.displayInformation();
20        emp4.displayInformation();
21        emp5.displayInformation();
22
23    }
24 }
25
26
```

測試

Console ×

<terminated> EmployeeTest [

=====員工資料=====

編號: 101

姓名: Sean

SSN: A123456789

薪水: \$50,000.00元

=====員工資料=====

編號: 102

姓名: Amy

SSN: B210987654

薪水: \$70,000.00元

=====員工資料=====

編號: 103

姓名: David

SSN: C109876543

薪水: \$80,000.00元

=====員工資料=====

編號: 104

姓名: Louis

SSN: D124680135

薪水: \$100,000.00元

管理部門: TW Sales

=====員工資料=====

編號: 105

姓名: Nicole

SSN: R202468135

薪水: \$120,000.00元

管理部門: Global Sales

管理預算: \$1,000,000.00

練習一之二

1. 修改 EmployeePractice 專案

2. 修改 Manager 類別設計如右

❑ addEmployee(e : Employee) : boolean

- 檢查Employee是否已存在manager所管理的ArrayList 中，已存在傳回 false
- Employee不存在，將Employee e 加入ArrayList，加入成功傳回 true

❑ removeEmployee(e : Employee) : boolean

- 檢查Employee是否已存在manager所管理的ArrayList中，不存在傳回 false
- Employee存在，將 Employee e由ArrayList中移除，移除成功傳回 true

❑ printStaffDetails()

- 列印 manager 所管理的所有 Employee 的 name 及 empID 如右程式片段

3. EmployeeTest 類別

- ❑ 使用員工陣列儲存各員工，用迴圈列印資訊
- ❑ 將Admin及Engineer加入Manager管理的員工
- ❑ 將Manager加入Director管理的員工

Manager
- deptName : String - employees : ArrayList
+ Manager(name: String, ssn String, salary double , dept: String)
+ getDeptName () : String + displayInformation() + addEmployee(e : Employee) : boolean + removeEmployee(e: Employee) : boolean + printStaffDetails()

```
for(Object obj : employees){  
    if(obj instanceof Employee){  
        Employee e = (Employee)obj;  
        .....  
    }  
}
```

Manager 类

```
36- public void printStaffDetails() {
37     if(employees.size()>0) {
38         System.out.print(getName()+"管理員工:");
39         for(Object obj: employees) {
40             if(obj instanceof Employee) {
41                 Employee e = (Employee)obj;
42                 System.out.print(String.format(" %s(%d)", e.getName(),e.getEmpId()));
43             }
44         }
45         System.out.println();
46     }
47 }
48
49- @Override
50 public void displayInformation() {
51     super.displayInformation();
52     System.out.println("管理部門: "+deptName);
53     this.printStaffDetails();
54 }
55
56 }
57
```

```
Manager.java ×
1 package com.example.domain;
2
3 import java.util.ArrayList;
4
5 public class Manager extends Employee {
6     private String deptName;
7     private ArrayList employees;
8
9
10 public Manager(String name, String ssn, double salary, String deptName) {
11     super(name, ssn, salary);
12     this.deptName = deptName;
13     employees = new ArrayList();
14 }
15
16 public String getDeptName() {
17     return deptName;
18 }
19
20 public boolean addEmployee(Employee emp) {
21     if(employees.contains(emp))
22         return false;
23     else {
24         employees.add(emp);
25         return true;
26     }
27 }
28
29 public boolean removeEmployee(Employee emp) {
30     if(employees.contains(emp)) {
31         employees.remove(emp);
32         return true;
33     }
34     return false;
35 }
```

EmployeeTest 類別

```
EmployeeTest.java ×
1 package com.example;
2
3 import com.example.domain.*;
4
5 public class EmployeeTest {
6     public static void main(String[] args) {
7         Employee[] emps = new Employee[5];
8         emps[0] = new Employee("Sean", "A123456789", 50000);
9         emps[1] = new Admin("Amy", "B210987654", 70000);
10        emps[2] = new Engineer("David", "C109876543", 80000);
11        emps[3] = new Manager("Louis", "D124680135", 100000, "TW Sales");
12        emps[4] = new Director("Nicole", "R202468135", 120000, "Global Sales", 1000000);
13
14        for(int i=0; i<emps.length; i++)
15            emps[i].displayInformation();
16
17        System.out.println("David 學會了Java, Android");
18        if(emps[2] instanceof Engineer) {
19            Engineer eng = (Engineer) emps[2];
20            eng.addSkill("Java");
21            eng.addSkill("Android");
22        }
23
24        System.out.println("部門分配.....");
25        if(emps[3] instanceof Manager) {
26            Manager m1 = (Manager) emps[3];
27            m1.addEmployee(emps[0]);
28            m1.addEmployee(emps[1]);
29            m1.addEmployee(emps[2]);
30        }
31
32        ((Manager) emps[4]).addEmployee(emps[3]);
33
34        for(int i=0; i<emps.length; i++)
35            emps[i].displayInformation();
36
37    }
38 }
39
```

測試

Console X

<terminated> EmployeeTest [Java

=====員工資料=====

編號: 101

姓名: Sean

SSN: A123456789

薪水: \$50,000.00元

=====員工資料=====

編號: 102

姓名: Amy

SSN: B210987654

薪水: \$70,000.00元

=====員工資料=====

編號: 103

姓名: David

SSN: C109876543

薪水: \$80,000.00元

=====員工資料=====

編號: 104

姓名: Louis

SSN: D124680135

薪水: \$100,000.00元

管理部門: TW Sales

=====員工資料=====

編號: 105

姓名: Nicole

SSN: R202468135

薪水: \$120,000.00元

管理部門: Global Sales

管理預算: \$1,000,000.00

David 學會了Java, Android

部門分配.....

=====員工資料=====

編號: 101

姓名: Sean

SSN: A123456789

薪水: \$50,000.00元

=====員工資料=====

編號: 102

姓名: Amy

SSN: B210987654

薪水: \$70,000.00元

=====員工資料=====

編號: 103

姓名: David

SSN: C109876543

薪水: \$80,000.00元

技能: Java Android

=====員工資料=====

編號: 104

姓名: Louis

SSN: D124680135

薪水: \$100,000.00元

管理部門: TW Sales

Louis管理員工: Sean(101) Amy(102) David(103)

=====員工資料=====

編號: 105

姓名: Nicole

SSN: R202468135

薪水: \$120,000.00元

管理部門: Global Sales

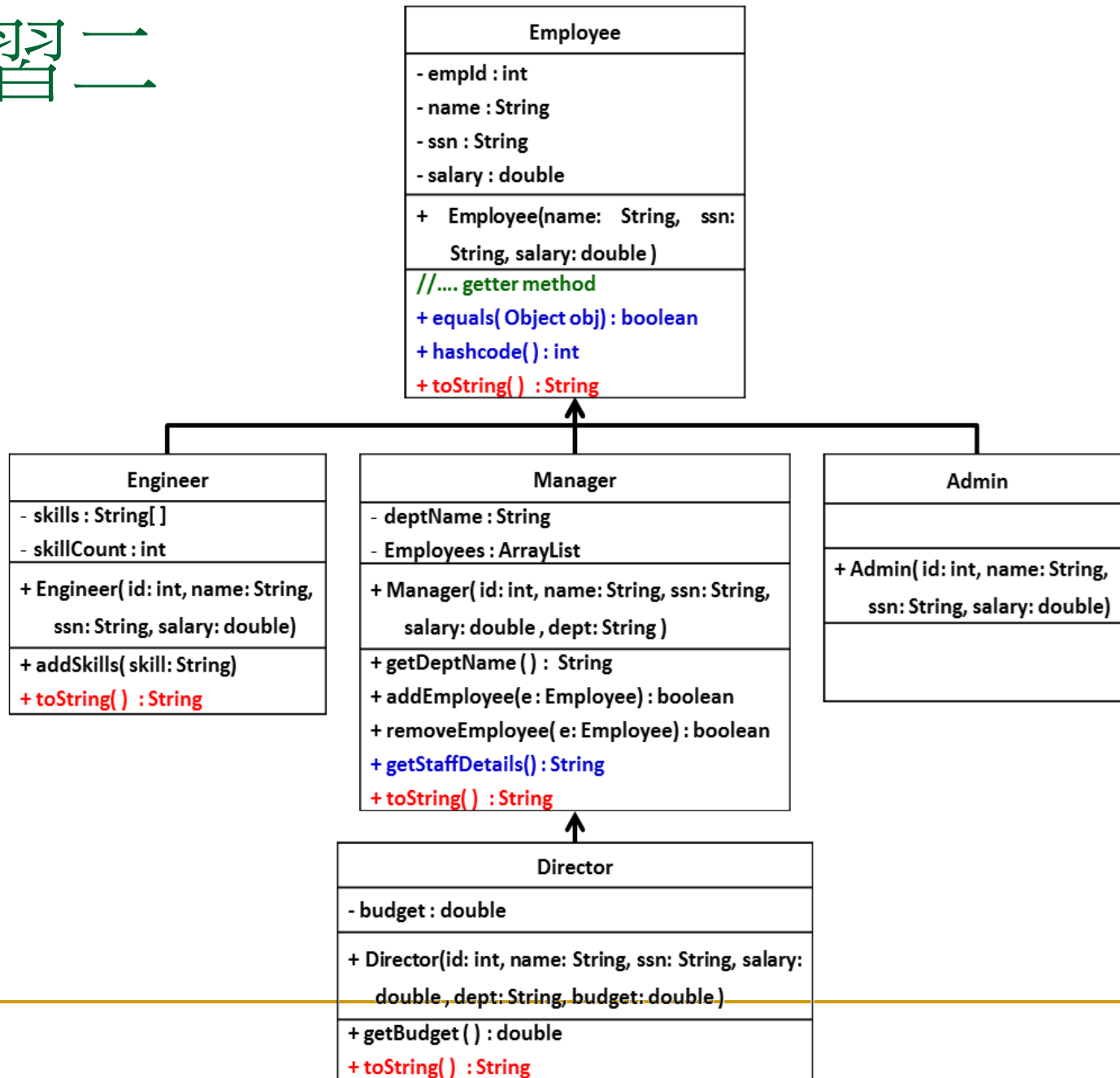
Nicole管理員工: Louis(104)

管理預算: \$1,000,000.00

練習二

1. 修改 EmployeePractice 專案
2. 修改 Employee 類別如右
 - ❑ 覆寫 equals() 及 hashCode() 方法
 - empld及ssn 相等
 - ❑ 覆寫 toString() 方法取代 displayInformation() 方法
 - 傳回empld、name、ssn、salary的字串
3. 修改 Manager 類別
 - ❑ 改寫 getStaffDetails() 方法傳回字串
 - ❑ 覆寫 toString() 方法加上 getStaffDetails() 字串
4. 修改 Engineer、Director 類別
 - ❑ 覆寫 toString() 方法加上其他相關屬性字串
5. 修改 EmployeeTest 類別
 - ❑ 將員工物件直接以 System.out.println() 列印

練習二



Ex2 Employee 類別

Employee.java ×

```
1 package com.example.domain;
2
3 import java.text.NumberFormat;
4
5 public class Employee {
6     public static int nextId = 101;
7     private int empId;
8     private String name = "John";
9     private String ssn = "A123456789";
10    private double salary = 26400;
11    protected NumberFormat formatter = NumberFormat.getCurrencyInstance();
12
13    public Employee(String name, String ssn, double salary) {
14
15    }
16
17    public int getEmpId() {
18
19    }
20
21    public String getName() {
22
23    }
24
25    public void setName(String name) {
26
27    }
28
29    public String getSsn() {
30
31    }
32
33    public double getSalary() {
34
35    }
36
37    public void raiseSalary(double increase) {
38
39    }
40
41    public void displayInformation() {
42        System.out.println("=====員工資料=====");
43        System.out.println("編號: "+this.empId);
44        System.out.println("姓名: "+this.name);
45        System.out.println("SSN: "+this.ssn);
46        System.out.println("薪水: "+formatter.format(salary)+"元");
47    }
48
49 }
```

```
53 @Override
54 public String toString() {
55     return "=====員工資料=====+"\n"+
56         "編號: "+this.empId+"\n"+
57         "姓名: "+this.name+"\n"+
58         "SSN: "+this.ssn+"\n"+
59         "薪水: "+formatter.format(salary)+"元"+"\\n";
60 }
61
```



Employee 類別

```
62 ② @Override
63  ▲ public int hashCode() {
64      final int prime = 31;
65      int result = 1;
66      result = prime * result + empId;
67      result = prime * result + ((ssn == null) ? 0 : ssn.hashCode());
68      return result;
69  }
70
71  ② @Override
72  ▲ public boolean equals(Object obj) {
73      if (this == obj) {
74          return true;
75      }
76      if (!(obj instanceof Employee)) {
77          return false;
78      }
79      Employee other = (Employee) obj;
80      if (empId != other.empId) {
81          return false;
82      }
83      if (ssn == null) {
84          if (other.ssn != null) {
85              return false;
86          }
87      } else if (!ssn.equals(other.ssn)) {
88          return false;
89      }
90      return true;
91  }
92
93 }
```

Engineer 類別

```
Engineer.java ×
1 package com.example.domain;
2
3 public class Engineer extends Employee {
4     private String[] skills;
5     private int skillCount;
6
7     public Engineer(String name, String ssn, double salary) {...}
12
13     public void addSkill(String skill) {...}
19
20     @Override
21     public String toString() {
22         StringBuilder sb = new StringBuilder(super.toString());
23         if(skillCount>0) {
24             sb.append("技能:");
25             for(int i=0; i<skillCount; i++)
26                 sb.append(" "+skills[i]);
27             sb.append("\n");
28         }
29         return sb.toString();
30     }
31
32 }
33
```

Manager 類別

```
Manager.java ×
1 package com.example.domain;
2
3 import java.util.ArrayList;
4
5 public class Manager extends Employee {
6     private String deptName;
7     private ArrayList employees;
8
9     public Manager(String name, String ssn, double salary, String deptName) {}
10
11     public String getDeptName() {}
12
13     public boolean addEmployee(Employee emp) {}
14
15     public boolean removeEmployee(Employee emp) {}
16
17     public String getStaffDetails() {
18         StringBuilder sb = new StringBuilder();
19         if(employees.size()>0) {
20             sb.append(getName()+"管理員工:");
21             for(Object obj: employees) {
22                 if(obj instanceof Employee) {
23                     Employee e = (Employee)obj;
24                     sb.append(String.format(" %s(%d)", e.getName(),e.getEmpId()));
25                 }
26             }
27             sb.append("\n");
28         }
29         return sb.toString();
30     }
31
32     @Override
33     public String toString() {
34         return super.toString() +
35             "管理部門: "+deptName + "\n" +
36             getStaffDetails();
37     }
38 }
39
40 }
```

Director 類別

```
Director.java ×
1 package com.example.domain;
2
3 public class Director extends Manager {
4     private double budget;
5
6     public Director(String name, String ssn, double salary, String deptName, double budget) {
7         super(name, ssn, salary, deptName);
8         this.budget = budget;
9     }
10
11     public double getBudget() {
12         return budget;
13     }
14
15     @Override
16     public String toString() {
17         return super.toString() +
18             "管理預算: " + formatter.format(budget) + "\n";
19     }
20
21 }
22
```

EmployeeTest 類別

```
EmployeeTest.java ×
1 package com.example;
2
3+ import com.example.domain.Admin;
4
5
6
7
8
9 public class EmployeeTest {
10
11
12     public static void main(String[] args) {
13         Employee[] emps = new Employee[5];
14         emps[0] = new Employee("Sean", "A123456789", 50000);
15         emps[1] = new Admin("Amy", "B210987654", 70000);
16         emps[2] = new Engineer("David", "C109876543", 80000);
17         emps[3] = new Manager("Louis", "D124680135", 100000, "TW Sales");
18         emps[4] = new Director("Nicole", "R202468135", 120000, "Global Sales", 1000000);
19
20         for(int i=0; i<emps.length; i++)
21             System.out.println(emps[i]);
22
23         System.out.println("David 學會了Java, Android");
24         if(emps[2] instanceof Engineer) {
25             Engineer eng = (Engineer) emps[2];
26             eng.addSkill("Java");
27             eng.addSkill("Android");
28         }
29
30         System.out.println("部門分配.....");
31         if(emps[3] instanceof Manager) {
32             Manager m1 = (Manager) emps[3];
33             m1.addEmployee(emps[0]);
34             m1.addEmployee(emps[1]);
35             m1.addEmployee(emps[2]);
36         }
37
38         ((Manager) emps[4]).addEmployee(emps[3]);
39
40         for(int i=0; i<emps.length; i++)
41             System.out.println(emps[i]);
42     }
43 }
44
```


測試

Console ×

<terminated> EmployeeTest [

=====員工資料=====

編號: 101

姓名: Sean

SSN: A123456789

薪水: \$50,000.00元

=====員工資料=====

編號: 102

姓名: Amy

SSN: B210987654

薪水: \$70,000.00元

=====員工資料=====

編號: 103

姓名: David

SSN: C109876543

薪水: \$80,000.00元

=====員工資料=====

編號: 104

姓名: Louis

SSN: D124680135

薪水: \$100,000.00元

管理部門: TW Sales

=====員工資料=====

編號: 105

姓名: Nicole

SSN: R202468135

薪水: \$120,000.00元

管理部門: Global Sales

管理預算: \$1,000,000.00

David 學會了Java, Android

部門分配.....

=====員工資料=====

編號: 101

姓名: Sean

SSN: A123456789

薪水: \$50,000.00元

=====員工資料=====

編號: 102

姓名: Amy

SSN: B210987654

薪水: \$70,000.00元

=====員工資料=====

編號: 103

姓名: David

SSN: C109876543

薪水: \$80,000.00元

技能: Java Android

=====員工資料=====

編號: 104

姓名: Louis

SSN: D124680135

薪水: \$100,000.00元

管理部門: TW Sales

Louis管理員工: Sean(101) Amy(102) David(103)

=====員工資料=====

編號: 105

姓名: Nicole

SSN: R202468135

薪水: \$120,000.00元

管理部門: Global Sales

Nicole管理員工: Louis(104)

管理預算: \$1,000,000.00