

# Ch8 課堂練習

鄭安翔

ansel\_cheng@hotmail.com

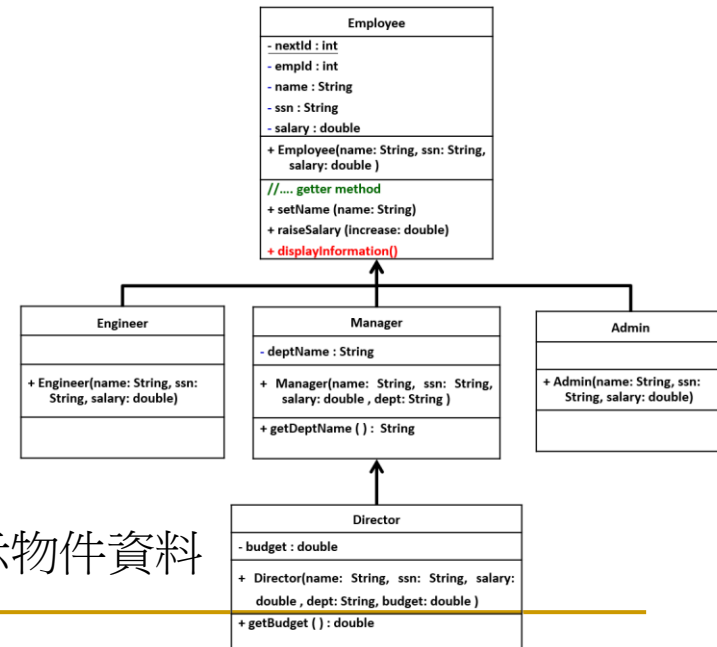
# 練習一

1. 修改 EmployeePractice 專案
2. 修改 Employee 類別 displayInformation() 方法
  - NumberFormat 物件的 format() 方法格式化顯示金額
    - ✓ java.text.NumberFormat.getCurrencyInstance() 取得物件
3. 建立 Employee 子類別 Engineer, Admin, Manager 及 Director

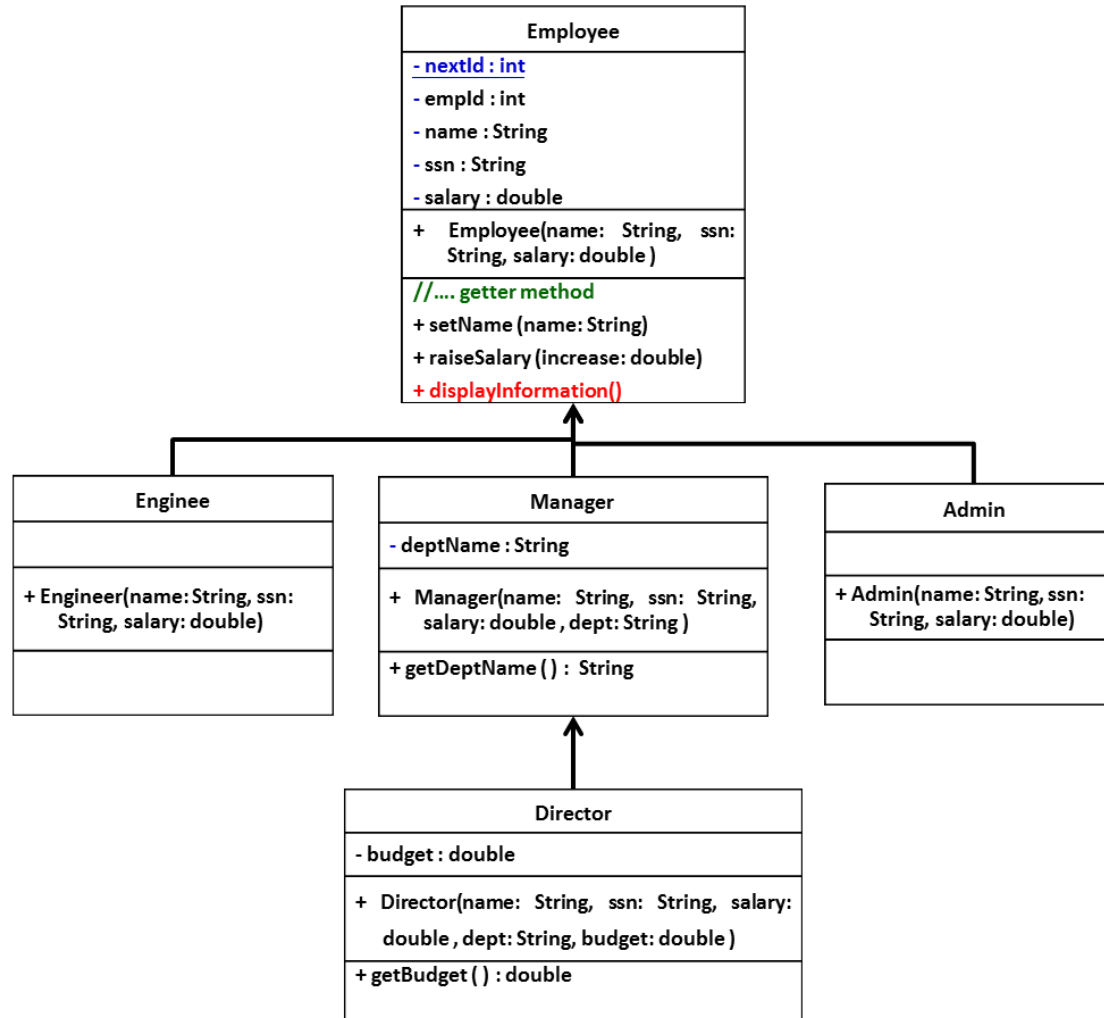
- Engineer, Admin 未新增屬性
- Manager 新增 deptName 屬性及 getter
- Director 新增 budget 屬性及 getter
- 建構子不被繼承

## 4. EmployeeTest 類別

- 使用建構子建立多個不同型態員工
- 呼叫物件的 displayInformation() 方法，顯示物件資料



# 練習一



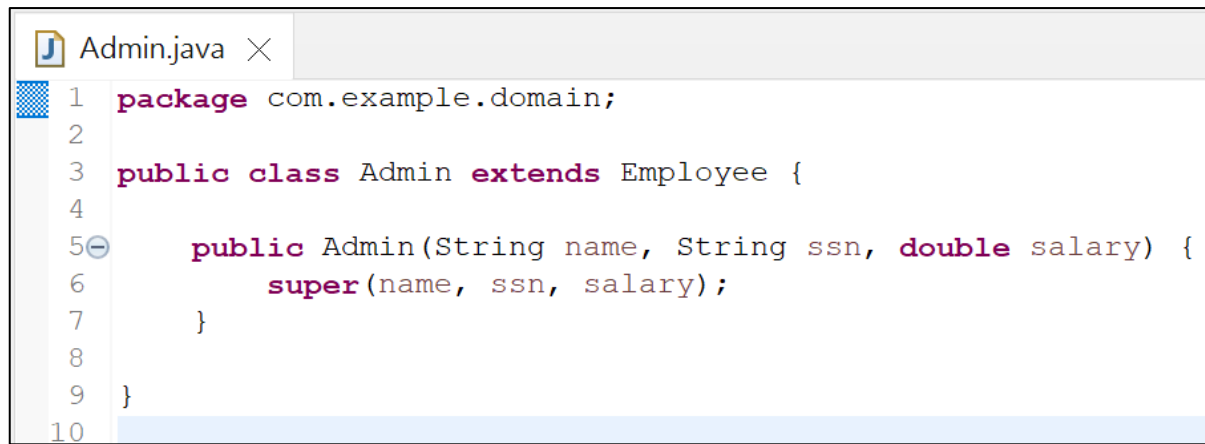
# Employee類別

\*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
12    public Employee(String name, String ssn, double salary) {
13        this.empId = nextId++;
14        if(name!=null && name.length() !=0)
15            this.name = name;
16        if(ssn!=null && ssn.length() !=0)
17            this.ssn = ssn;
18        if(salary>26400)
19            this.salary = salary;
20    }
21
22    public int getEmpId() {
23        return empId;
24    }
25
26    public String getName() {
27        return name;
28    }
29
30    public String getSsn() {
31        return ssn;
32    }
33
34    public double getSalary() {
35        return salary;
36    }
37
```

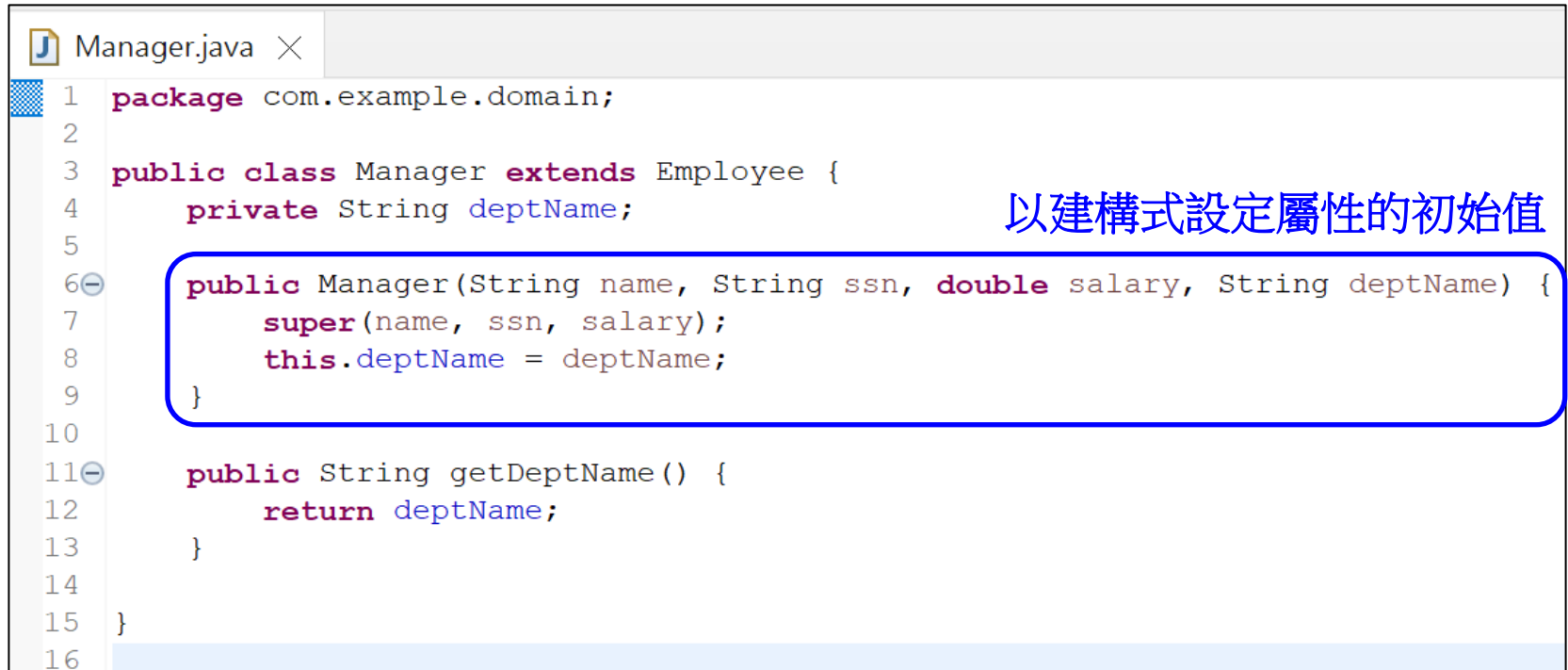
```
38    public void setName(String name) {
39        if(name!=null && name.length() !=0)
40            this.name = name;
41        else
42            System.out.println("參數為空值改名失敗!");
43    }
44
45    public void raiseSalary(double increase) {
46        if(increase>0)
47            this.salary += increase;
48        else
49            System.out.println("參數錯誤加薪失敗!");
50    }
51
52    public void displayInformation() {
53        NumberFormat formatter = NumberFormat.getCurrencyInstance();
54        System.out.println("=====員工資料=====");
55        System.out.println("編號: "+this.empId);
56        System.out.println("姓名: "+this.name);
57        System.out.println("SSN: "+this.ssn);
58        System.out.println("薪水: "+formatter.format(salary)+"元");
59    }
60
61
62 }
```

# Admin類別



```
Admin.java ×
1 package com.example.domain;
2
3 public class Admin extends Employee {
4
5     public Admin(String name, String ssn, double salary) {
6         super(name, ssn, salary);
7     }
8
9 }
10
```

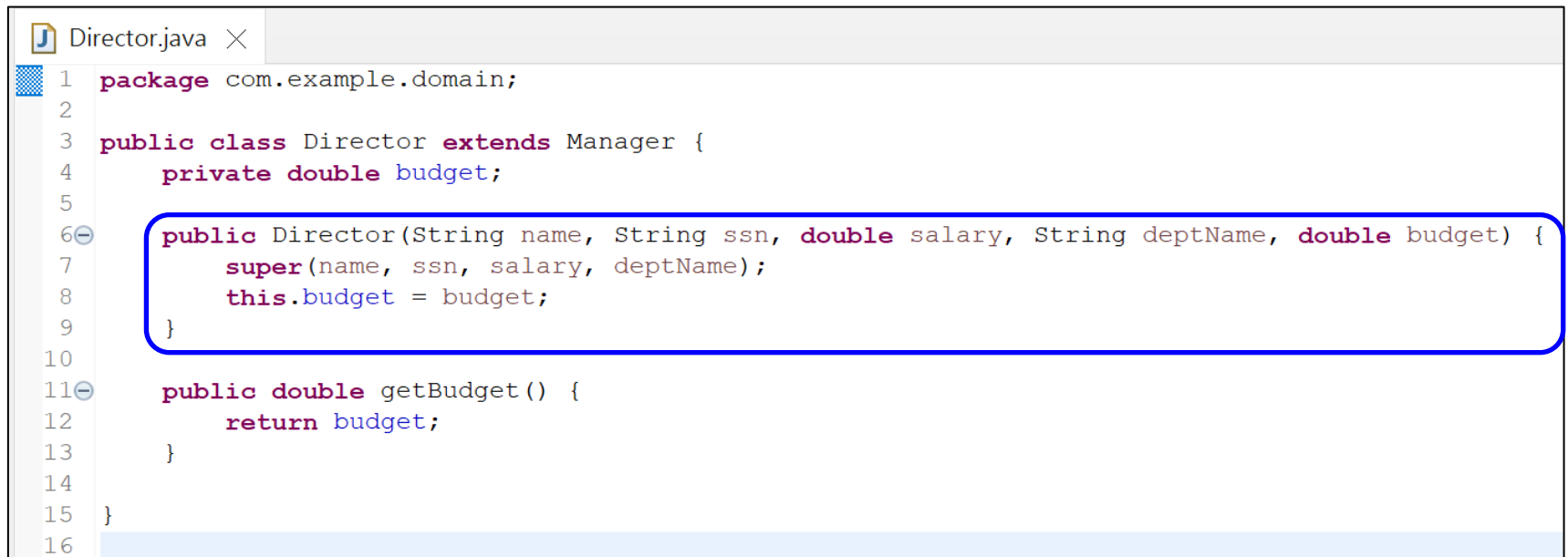
# Manager類別



```
1 package com.example.domain;
2
3 public class Manager extends Employee {
4     private String deptName;
5
6     public Manager(String name, String ssn, double salary, String deptName) {
7         super(name, ssn, salary);
8         this.deptName = deptName;
9     }
10
11     public String getDeptName() {
12         return deptName;
13     }
14
15 }
16
```

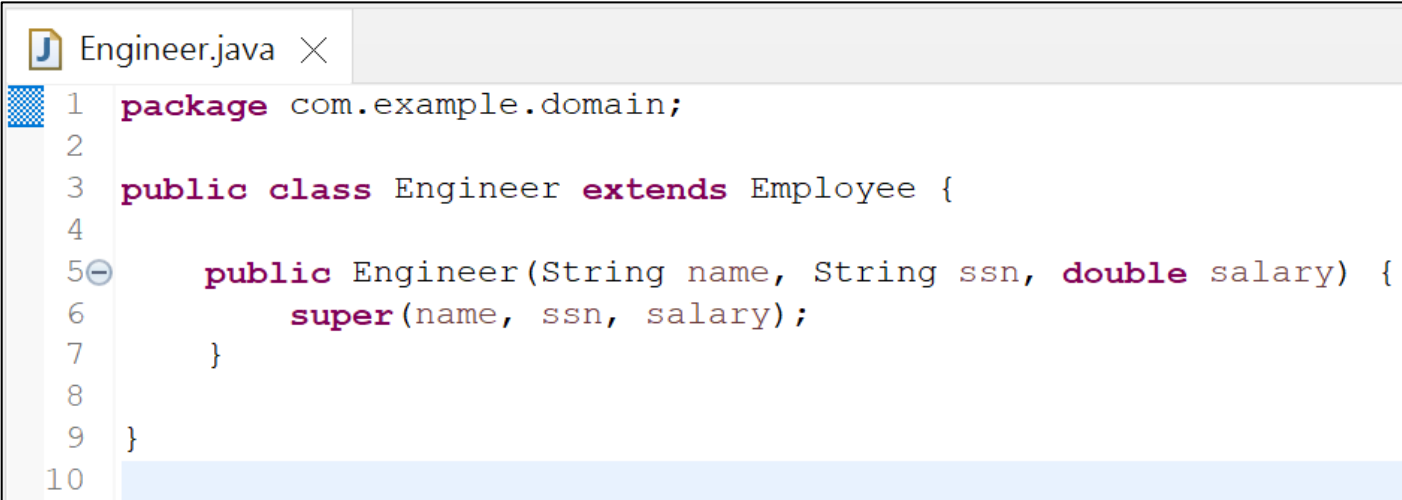
以建構式設定屬性的初始值

# 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 }
16
```

# Engineer 類別



```
Engineer.java ×
1 package com.example.domain;
2
3 public class Engineer extends Employee {
4
5     public Engineer(String name, String ssn, double salary) {
6         super(name, ssn, salary);
7     }
8
9 }
10
```



# EmployeeTest

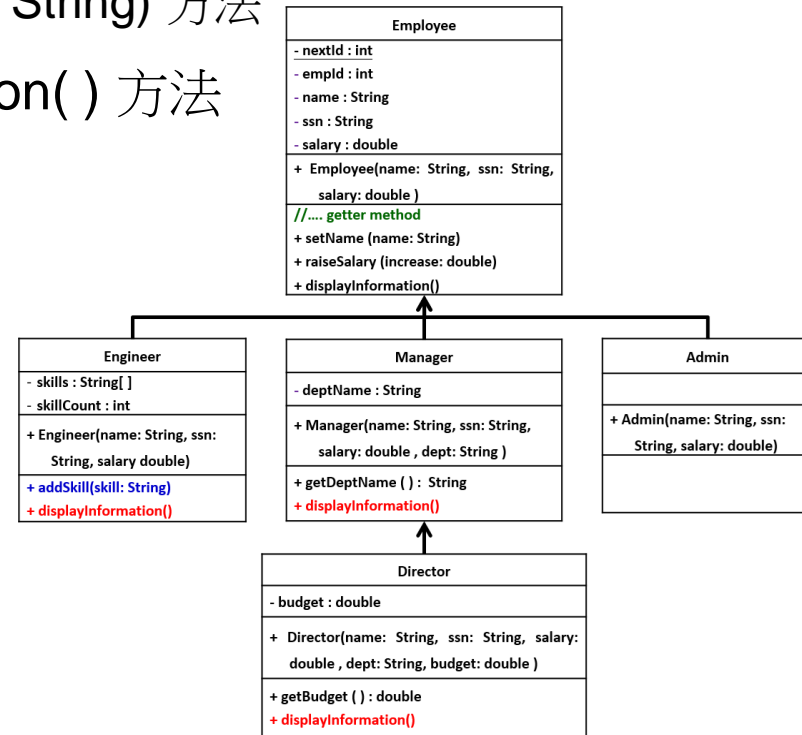
```
EmployeeTest.java ×
1 package com.example;
2
3 import com.example.domain.*;
4
5 public class EmployeeTest {
6     public static void main(String[] args) {
7         Employee emp1 = new Employee("Sean", "A123456789", 50000);
8         Admin emp2 = new Admin("Amy", "B210987654", 70000);
9         Engineer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

# 測試

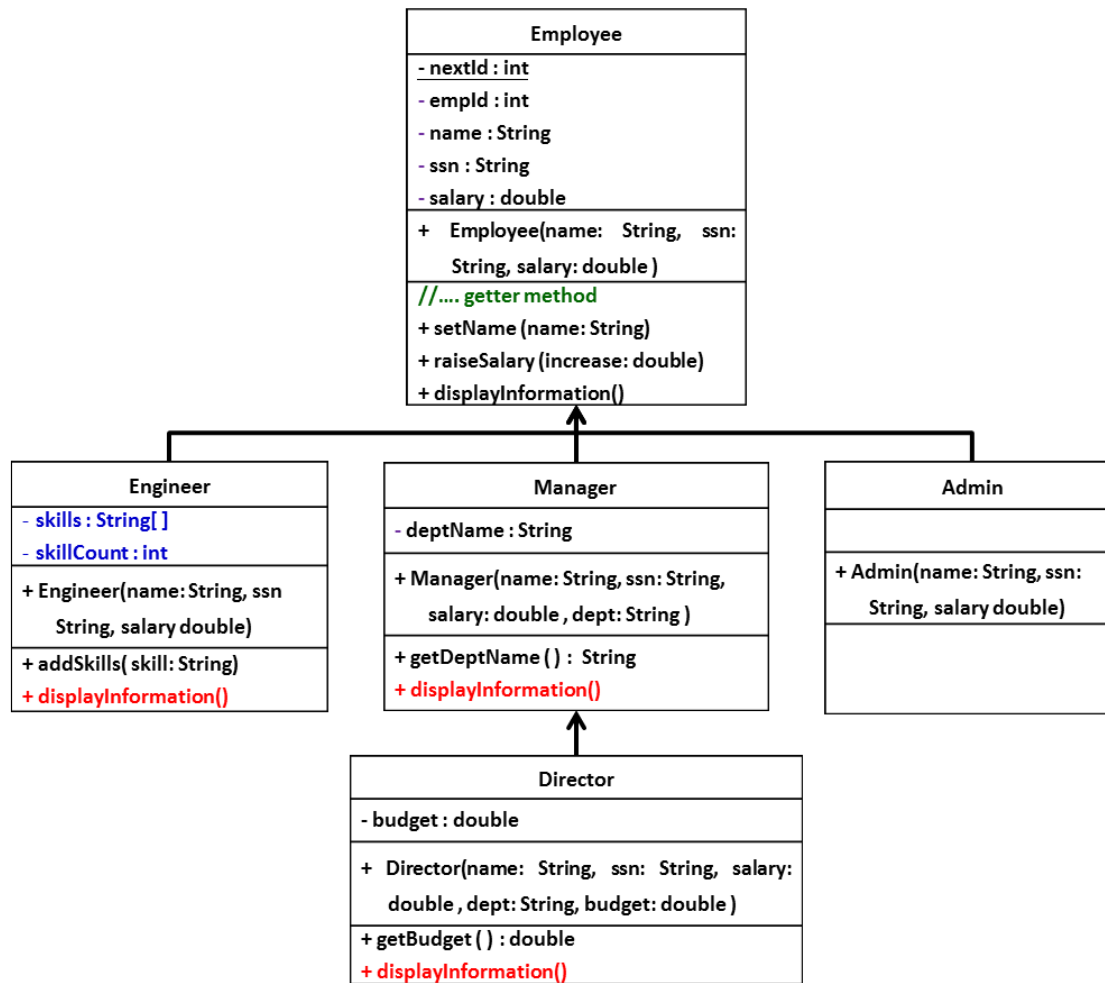
```
Console X
<terminated> EmployeeTest
=====員工資料=====
編號: 101
姓名: Sean
SSN: A123456789
薪水: $50,000.00元
=====員工資料=====
編號: 102
姓名: Amy
SSN: B210987654
薪水: $70,000.00元
=====員工資料=====
編號: 103
姓名: David
SSN: C109876543
薪水: $80,000.00元
=====員工資料=====
編號: 104
姓名: Louis
SSN: D124680135
薪水: $100,000.00元
=====員工資料=====
編號: 105
姓名: Nicole
SSN: R202468135
薪水: $120,000.00元
```

# 練習二

1. 修改 EmployeePractice 專案
2. 修改 Engineer 類別
  - 新增 `String[]` 屬性 `skills` 及 `addSkill(skill: String)` 方法
3. Employee 子類別覆寫 `displayInformation()` 方法
  - Admin 不需覆寫
  - Engineer 額外顯示 Skills 資料
  - Manager 額外顯示 `deptName` 資料
  - Director 額外顯示 `budget` 資料
4. EmployeeTest 類別
  - 使用建構子建立多個不同型態員工
  - Engineer 員工加入技能 `Java, Android`
  - 呼叫物件的 `displayInformation()` 方法，顯示物件資料



# 練習二



# Manager類別

```
Manager.java ×
1 package com.example.domain;
2
3 public class Manager extends Employee {
4     private String deptName;
5
6     public Manager(String name, String ssn, double salary, String deptName) {
7         super(name, ssn, salary);
8         this.deptName = deptName;
9     }
10
11     public String getDeptName() {
12         return deptName;
13     }
14
15     @Override
16     public void displayInformation() {
17         super.displayInformation();
18         System.out.println("管理部門: "+deptName);
19     }
20
21 }
22
```

覆寫displayInformation()方法

# 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 void displayInformation() {
17         super.displayInformation();
18         System.out.println("管理預算: "+formatter.format(budget));
19     }
20
21 }
22
```

覆寫displayInformation()方法

# 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) {
8         super(name, ssn, salary);
9         skills = new String[5];
10        skillCount = 0;
11    }
12
13    public void addSkill(String skill) {
14        if(skillCount<5)
15            skills[skillCount++] = skill;
16        else
17            System.out.println("最多註冊五種技能,新增失敗!");
18    }
19
20    @Override
21    public void displayInformation() {
22        super.displayInformation();
23        if(skillCount>0) {
24            System.out.print("技能:");
25            for(int i=0; i<skillCount; i++)
26                System.out.print(" "+skills[i]);
27            System.out.println();
28        }
29    }
30
31 }
32
33
```

覆寫displayInformation()方法

# EmployeeTest

```
EmployeeTest.java X
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         Admin emp2 = new Admin("Amy", "B210987654", 70000);
9         Engineer emp3 = new Engineer("David", "C109876543", 80000);
10        Manager emp4 = new Manager("Louis", "D124680135", 100000, "TW Sales");
11        Director emp5 = new Director("Nicole", "R202468135", 120000, "Global Sales", 1000000);
12
13        emp1.displayInformation();
14        emp2.displayInformation();
15        emp3.displayInformation();
16        emp4.displayInformation();
17        emp5.displayInformation();
18
19        System.out.println("David 學會了Java, Android");
20        emp3.addSkill("Java");
21        emp3.addSkill("Android");
22
23        emp1.displayInformation();
24        emp2.displayInformation();
25        emp3.displayInformation();
26        emp4.displayInformation();
27        emp5.displayInformation();
28
29    }
30 }
31
```



# 測試

Console ×

<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

=====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

=====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