

12/22當天

1. 設計一父類別 Device，擁有 brand, size, price 等屬性，且具有開機、關機功能，設計二個子類別 Laptop, 以及 SmartPhone 繼承 Device 類別，且設計其他屬性與功能
2. 設計一個 Employee 類別，具備 name, id, 以及 salary 屬性，且提供 getSalary(), getBonus() 等功能查詢薪資與獎金。設計二個子類別 Manager, Developer 繼承 Employee 類別，且 override getSalary(), getBonus() 成員函式，並具備自己獨有的功能設計
3. 以下程式碼，
 - * 當主程式的 (1) 執行時，(a), (b), (c), (d), (e) 哪幾行會被執行？先後順序關係為何？
 - * 當主程式的 (2) 執行時，(a), (b), (c), (d), (e) 哪幾行會被執行？先後順序關係為何？

```
1 public class staticEx {
2     static int a=2; //(a)
3
4     { System.out.println("non-static block a="+a); a=3;} //(b)
5
6     static {
7         System.out.println("static block: a="+a); a=4; //(c)
8     }
9     staticEx() { System.out.println("constructor: a="+a); a=5; } //(d)
10    static void fun() { System.out.println("static fun(): a="+a); a=6; } //(e)
11    public static void main(String[] args) {
12        staticEx fun(); //(1)
13    }
```

第一題:

```
/* 12/22 楊育哲
 * 實作第一題：父類別繼承練習
 */
public class p1_1222 {
    public static void main(String args[]){
        Laptop A = new Laptop("HP", 10000, 11);
        Laptop C = new Laptop("A", 20000, 13);
        A.onBot(); C.onBot();
        A.info(); C.info();

        SmartPhone B = new SmartPhone("123546789", "apple", 1
        System.out.print("call test: "); B.call("987654321");
        B.offBot();
```

```

        System.out.print("call test: "); B.call("987654321");
    }
}
class Device{
    private String brand;
    private int size, price;
    private boolean state;
    Device(String name, int s, int p){state=false;brand=name;
    public void onBot(){state=true;}
    public void offBot(){state=false;}
    public boolean Bot(){return state;}
    public void info(){
        System.out.println("brand: "+brand);
        System.out.println("size: "+size);
        System.out.println("price: "+price);
    }
}
class Laptop extends Device{
    private static int power, voice;
    Laptop(String name, int s, int p){super(name, s, p); power=100; voice=0;
    public void setVoice(int v){if(v>=0&&v<=100)voice=v;}
    public void checkPower(){System.out.println("power :"+power);
}
class SmartPhone extends Device{
    private static String number;
    SmartPhone(String no, String name, int s, int p){super(name, s, p); number=no;
    public void call(String no){if(Bot())System.out.println("call test: call "+no+" from "+number);
}
}

```

brand: HP

size: 10000

price: 11

brand: A

size: 20000

price: 13

call test: call 987654321 from 123546789

call test: phone off.

第二題:

```

/* 12/22 楊育哲
 * 實作第二題：繼承練習2
 */
public class p2_1222 {
    public static void main(String args[]){
        Manager andy = new Manager("andy", 1, 10000);
        Developer cammy = new Developer("cammy", 2, 10000);
        System.out.println("developer's salary: "+cammy.getSa
        andy.adjustSalary(15000);
        System.out.println("after salary adjusted, developer'
        System.out.println("before cammy work overtime, she's
        cammy.overtime();
        System.out.println("after cammy work overtime once, s
    }
}
class Employee{
    private int id, salary;
    static int adjustedSalary;
    private String name;
    Employee(String n, int i, int s){name=n; id=i; salary=s;a
    Employee(){}
    static void setSalary(int s){adjustedSalary=s;}
    public int getSalary(){ salary=adjustedSalary; return sal
    public int getBouns(){return salary*3;}
}
class Manager extends Employee{
    Manager(String n, int i, int s){super(n, i, s);}
    @Override
    public int getSalary(){ return super.getSalary()+10000; }
    @Override
    public int getBouns(){ return super.getBouns()*3; }
    public void adjustSalary(int s){setSalary(s);}
}
class Developer extends Employee{
    private float rate=2;
    Developer(String n, int i, int s){super(n, i, s);}
    @Override
    public int getSalary(){

```

```

        Employee check = new Employee();
        if(check.getSalary()!=super.getSalary()) super.setSal
        return super.getSalary()+5000; }
@Override
public int getBouns(){ return (int)(super.getBouns()*rate
public void overtime(){ rate+=0.1; }
}

```

輸出:

developer's salary: 15000

after salary adjusted, developer's salary: 20000

before cammy work overtime, she's bouns: 90000

after cammy work overtime once, she's bouns: 94499

3. 以下程式碼，

- * 當主程式的 (1) 執行時，(a), (b), (c), (d), (e) 哪幾行會被執行？先後順序關係為何？
- * 當主程式的 (2) 執行時，(a), (b), (c), (d), (e) 哪幾行會被執行？先後順序關係為何？

```

1
2 public class staticEx {
3     static int a=2; //(a)
4
5     { System.out.println("non-static block a="+a); a=3;} //(b)
6
7     static {
8         System.out.println("static block: a="+a); a=4; //(c)
9     }
10    staticEx() { System.out.println("constructor: a="+a); a=5; } //(d)
11    static void fun() { System.out.println("static fun(): a="+a); a=6; } //(e)
12    public static void main(String[] args) {
13        staticEx.fun(); //(1)
14        System.out.println("-----");
15        (new staticEx()).fun(); //(2)
16    }
17 }
18 |

```

/* 12/22 楊育哲

- * 實作上課範例：先後順序探討

```

    */
public class StaticEx {
    static int a=2; //(a)

    { System.out.println("non-static block a="+a); a=3; } //(b)

    static{
        System.out.println("static block a="+a); a=4; //(c)
    }

    StaticEx(){ System.out.println("constructor: a="+a); a=5;

    static void fun(){ System.out.println("static fun(): a="+a); }

    public static void main(String[] args){
        StaticEx.fun(); //(1)
        System.out.println("-----");
        (new StaticEx()).fun(); //(2)
    }
}
//(1): a -> c -> e, (2) [承接(1)] -> b -> d -> e
/*
static block a=2
static fun(): a=4
-----
non-static block a=6
constructor: a=3
static fun(): a=5
*/

```

輸出如註解所示。