

物件導向程式語言認證

Java



Contents

2



1. 第一類- 基本認識

2. 第二類-條件判斷式

3. 第三類-迴圈

4. 第四類-遞迴程式設計

5. 第五類-陣列設計能力

6. 第六類-物件導向程式設計與例外處理

7. Q & A

第一類-基本認識



距離計算

4



□ 設計說明：

- (1) `Math`套件(package)內有許多數學上常用的函數。
- (2) 請使用 `Math.pow` 及 `Math.sqrt` 撰寫程式，顯示【輸入第1組的 x 和 y 座標：】，提示使用者首先輸入第一組座標($x1, y1$)， x 與 y 座標的輸入以一空格鍵分隔，其資料型態皆為正整數。
- (3) 要求輸入第2組座標($x2, y2$)。
- (4) 兩組座標輸入完畢，輸出兩組座標之間的距離，顯示如執行結果參考畫面。

距離計算

5



- 執行結果參考畫面：

 命令提示字元

```
D:\>java JPA104  
請輸入第1組的x和y座標:1 5  
請輸入第2組的x和y座標:10 22  
介於(1.00,5.00)和(10.00,22.00)之間的距離是19.24
```

距離計算

6



□ 設計說明：

```
1 import java.util.Scanner;
2
3 public class JPA104{
4     public static void main (String[] args) {
5         Scanner sc = new Scanner(System.in);
6         double x1,x2,y1,y2;
7         System.out.print("請輸入第1組的x和y座標:");
8         x1 = sc.nextDouble();
9         y1 = sc.nextDouble();
10
11         System.out.print("請輸入第2組的x和y座標:");
12         x2 = sc.nextDouble();
13         y2 = sc.nextDouble();
14         //printf的用法是顯示出一串字串，而字串中會有一些變數可帶入，例如%d 就是帶入整數，%f就是帶入浮點數，而%.2f是指全部的位數有四位（包含小數點），而小數點後面佔兩位
15         //不過如果輸出的為數大於使用者輸出的，則會忽略使用者所輸入的位數限制
16         //Math.sqrt(X) = X開根號
17         //Math.pow(2.0,3.0) = 二的三次方，其中兩個變數皆是double形式
18         System.out.printf("介於(%.2f,%.2f)和(%.2f,%.2f)之間的距離是%.2f", x1,y1,x2,y2,(Math.sqrt(Math.pow((x2-x1),2) + Math.pow((y2-y1),2))));
19     }
20 }
```

數學函數

7



□ 設計說明：

- (1) 有一數學函數 $f(x) = 3x^3 + 2x - 1$ 。
- (2) 請撰寫一函數 f 用來傳回 $f(x)$ 的值，並分別計算 $f(-3.2)$ 、 $f(-2.1)$ 、 $f(0)$ 、與 $f(2.1)$ 。
- (3) 函數 $f(x)$ 值取到小數第四位。
- (4) 依此數學函數計算，分別輸出此四個計算值，顯示如執行結果參考畫面。

數學函數

8



- 執行結果參考畫面：

 命令提示字元

```
D:\>java JPA106  
f(-3.2) = -105.7040  
f(-2.1) = -32.9830  
f(0.0) = -1.0000  
f(2.1) = 30.9830
```


數學函數

9



□ 設計說明：

```
1 public class JPA106 {  
2     public static void main (String[] args) {  
3         double x;  
4         x = -3.2;  
5         System.out.printf("f(%.1f) = %4.4f\n",x,f(x));  
6         x = -2.1;  
7         System.out.printf("f(%.1f) = %4.4f\n",x,f(x));  
8         x = 0;  
9         System.out.printf("f(%.1f) = %4.4f\n",x,f(x));  
10        x = 2.1;  
11        System.out.printf("f(%.1f) = %4.4f\n",x,f(x));  
12    }  
13    static double f(double d) {  
14        return (3*(Math.pow(d, 3))+2*d-1);  
15    }  
16 }
```

第二類-條件判斷式



比較大小

11



□ 設計說明：

- ▣ (1) 請設計一程式，使用者可以輸入兩個整數，並且可以比較兩個整數的大小。
- ▣ (2) 程式執行時，畫面顯示【Input:】，請使用者輸入兩組整數，各組整數分別輸入兩個數字，數字中間以空隔間隔。
- ▣ (3) 重複執行兩次，依輸入的兩個整數比較大小，顯示如執行結果參考畫面。

比較大小

12



- 執行結果參考畫面：

 命令提示字元

```
D:\>java JPA202
Input:
50 88
88 is larger than 50
Input:
33 45
45 is larger than 33
```

比較大小

13



□ 設計說明：

```
1 import java.util.*;
2 public class JPD02 {
3     static Scanner keyboard = new Scanner(System.in);
4     public static void main(String[] args) {
5         test();
6         test();
7     }
8
9     public static void test() {
10        System.out.println("Input:");
11        int num1 = keyboard.nextInt();
12        int num2 = keyboard.nextInt();
13        if(num1 > num2)
14            System.out.println(num1 + " is larger than " + num2);
15        else if(num1 < num2)
16            System.out.println(num2 + " is larger than " + num1);
17        else
18            System.out.println(num1 + " is equal to " + num2);
19    }
20 }
```

鍵盤字元判斷

14



□ 設計說明：

- ▣ (1) 請使用switch完成編輯區中的test()的程式。
- ▣ (2) 程式執行時，畫面顯示【Input a character:】，請使用者輸入一個英文字母。
- ▣ (3) 若輸入 a 或 b ，顯示【You entered a or b】。
- ▣ (4) 若輸入 x ，顯示【You entered x】；若輸入 y ，顯示【You entered y】。
- ▣ (5) 若皆非上述所列英文字母，則顯示【You entered something else.】。


鍵盤字元判斷

15



□ 執行結果參考畫面：

□ 程式執行時，畫面顯示[Input a character:]，要求輸入英文字母。

 命令提示字元

```
D:\>java JPA210
Input a character:
a
You entered a or b
Input a character:
b
You entered a or b
Input a character:
x
You entered x
Input a character:
y
You entered y
Input a character:
c
You entered something else.
```

鍵盤字元判斷



16

□ 設計說明：

```
1 import java.util.*;
2 public class JPA210 {
3     static Scanner keyboard = new Scanner(System.in);
4     public static void main(String[] args) {
5         test();
6         test();
7         test();
8         test();
9         test();
10    }
11
12    public static void test() {
13        System.out.println("Input a character:");
14        String input = keyboard.next();
15        switch(input) {
16            case "a":
17                System.out.println("You enter a or b:");
18                break;
19            case "b":
20                System.out.println("You enter a or b:");
21                break;
22            case "x":
23                System.out.println("You enter "+input);
24                break;
25            case "y":
26                System.out.println("You enter "+input);
27                break;
28            default:
29                System.out.println("You enter something else");
30                break;
31        }
32    }
33 }
```


第三類-迴圈



巢狀迴圈

18



□ 設計說明：

- ▣ (1) 一個外迴圈包住一個或多個內迴圈的巢狀迴圈，每次執行外迴圈時，都會進入內迴圈，重複執行。
- ▣ (2) 「假設」外迴圈重複了3次，而內迴圈重複了2次，則內迴圈內的敘述就會執行 $3 * 2$ 次。
- ▣ (3) 請完成JPD03.java檔案中的程式，讓count之值為27。
- ▣ (4) 顯示如執行結果參考畫面。

巢狀迴圈

19



- 執行結果參考畫面：

 命令提示字元

```
D:\>java JPA302  
count = 27
```

巢狀迴圈

20



□ 設計說明：

```
1 import java.util.Scanner;
2 public class JPD03 {
3     public static void main(String[] args) {
4         int i = 1, j = 1, count = 0;
5         for (i = 1; i <= 3; i++) {           //第一個迴圈，i從1到3
6             for(j=1;j<=9;j++) {
7                 count++;
8             }
9         }
10        System.out.printf("count = %d\n", count);
11    }
12 }
```

餐點費用

21



□ 設計說明：

- ▣ (1) 有五位朋友到知名美式餐廳聚餐慶生，此餐廳採點餐的計算方式。
- ▣ (2) 程式執行時，如執行結果參考畫面(1)，畫面顯示【**Please enter meal dollars or enter -1 to stop:**】，請使用者輸入第一道餐點費用，再分別依序要求輸入其他道餐點的費用。
- ▣ (3) 若要停止執行程式計算，需輸入-1後停止程式計算，-1不列入餐點計算費用內。請輸出餐點數量、所有餐點的總費用及平均負擔費用，平均值取自小數第二位。
- ▣ (4) 顯示如執行結果參考畫面。

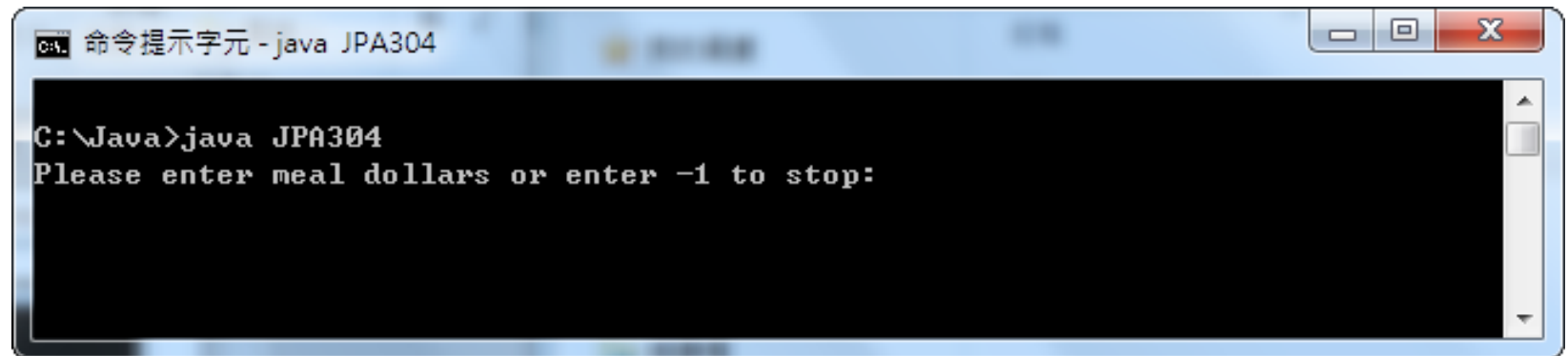
餐點費用

22



□ 執行結果參考畫面：

- ▣ (1) 程式執行時，畫面顯示「Please enter meal dollars or enter -1 to stop：」，請使用者輸入第一道餐點費用。



```
命令提示字元 - java JPA304

C:\Java>java JPA304
Please enter meal dollars or enter -1 to stop:
```

餐點費用

23



□ 執行結果參考畫面：

- ▣ (2) 分別依序要求輸入費用，最後輸入 -1，輸出所有餐點的總費用及平均負擔費用，平均值取至小數第二位。

```
C:\Java>java JPA304
Please enter meal dollars or enter -1 to stop: 180
Please enter meal dollars or enter -1 to stop: 120
Please enter meal dollars or enter -1 to stop: 99
Please enter meal dollars or enter -1 to stop: 399
Please enter meal dollars or enter -1 to stop: 150
Please enter meal dollars or enter -1 to stop: -1
餐點總費用:948.0
5 道餐點平均費用為: 189.60
C:\Java>
```

餐點費用

24



□ 設計說明：

```
1 import java.util.Scanner;
2 public class JPD03 {
3
4     public static void main(String[] args) {
5         int total = 0;
6         int s = 0;
7         int count = 0;
8         double average;
9
10        System.out.print("Please enter meal dollars or enter -1 to stop: ");
11
12        s = new Scanner(System.in).nextInt();
13
14        while(s != -1) {
15            count++;
16            total+=s;
17            System.out.print("Please enter meal dollars or enter -1 to stop: ");
18            s = new Scanner(System.in).nextInt();
19        }
20        average=(double)total/count;
21        System.out.println("餐點總費用:"+total);
22        System.out.printf(" %d 道餐點平均費用為: %.2f %n",count,average);
23    }
24 }
```


第四類-遞迴程式設計



遞迴字串反向

26



□ 設計說明：

- (1) 使用遞迴設計一個類別方法，此方法能夠將字串反向。
- (2) 程式執行時，顯示【Input a string: 】要求輸入字串。
- (3) 連續執行兩次，結果參考如下頁，將字串反向印出。

遞迴字串反向

27



- 執行結果參考畫面：

 命令提示字元

```
D:\>java JPA408
Input a string: computer
retupmoc
Input a string: skills
slliks
```

遞迴字串反向

28



□ 設計說明：

```
1 import java.util.Scanner;
2 public class JPA408 {
3     static Scanner keyboard = new Scanner(System.in);
4     public static void main(String args[]) {
5         String s, c;
6         System.out.print("Input a string: ");
7         s = keyboard.nextLine();
8         System.out.printf("%s\n", reverse(s));
9         System.out.print("Input a string: ");
10        s = keyboard.nextLine();
11        System.out.printf("%s\n", reverse(s));
12    }
13
14    static String reverse(String s )
15    {
16        if(s.equals(""))//如果傳入字串為空，則回傳s字串
17            return s;
18        else//將字串的第一個字元從後方累加上，然後傳入的字串則是透過substring讀出位置1之後的所有字串並傳入reverse
19            return reverse(s.substring(1))+s.substring(0, 1);
20    }
21 }
```

遞迴字串替換

29



□ 設計說明：

- ▣ (1) 請使用遞迴撰寫一個類別方法，此方法能夠將一個字串內的某個字元換成另一個字元。例如輸入字串【windows】，將字串中 w 值替換成 g 值，輸出字串為【gindogs】。
- ▣ (2) 程式執行時，顯示【Input a string: 】要求輸入字串，接續顯示【Input a character: 】要求輸入「被替換」的字元，最後顯示【Input another character: 】要求輸入替換字元。
- ▣ (3) 請利用 `replace()` 函數進行字串替換，顯示如執行結果參考畫面。

遞迴字串替換

30



- 執行結果參考畫面：

A screenshot of a Windows command prompt window titled "命令提示字元". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is black with white text. The text shows the execution of a Java program named JPA410. The user enters the command "java JPA410". The program prompts for "Input a string:", and the user enters "windows". It then prompts for "Input a character:", and the user enters "w". Finally, it prompts for "Input another character:", and the user enters "g". The output of the program is "gindogs". The prompt "C:\Java>" is visible at the bottom of the window.

```
C:\Java>java JPA410
Input a string: windows
Input a character: w
Input another character: g
gindogs
C:\Java>
```

遞迴字串替換

31



□ 設計說明：

```
1 import java.util.Scanner;
2 public class JPA410 {
3     static Scanner keyboard = new Scanner(System.in);
4     public static void main(String args[]){
5         String s, c1, c2;
6         System.out.print("Input a string: ");
7         s = keyboard.nextLine();
8         System.out.print("Input a character: ");
9         c1 = keyboard.nextLine();
10        System.out.print("Input another character: ");
11        c2 = keyboard.nextLine();
12        System.out.printf("%s\n", replace(s, c1, c2));
13    }
14
15    static String replace(String s, String c1, String c2){
16        if(s.equals(""))//如果傳入的字串為空的話，則回傳空字串
17            return "";
18        if(s.substring(0, 1).equals(c1))//如果讀出的字元等於要取代的字元，則將新的字元加上
19            return c2+replace(s.substring(1),c1,c2);
20        else//如果不等於的話，補回至原本的字串
21            return s.substring(0, 1)+replace(s.substring(1),c1,c2);
22    }
23 }
```

第五類-陣列設計能力



浮點數計算

33



□ 設計說明：

- ▣ (1) 請撰寫一程式，由鍵盤輸入學生的人數，根據所輸入的學生人數，動態產生一個符合大小的浮點數陣列。
- ▣ (2) 將所輸入的每位學生成績存放到陣列裡(不限制輸入的小數點位數)。
- ▣ (3) 程式執行時，顯示【請輸入學生人數：】，要求輸入學生人數。
- ▣ (4) 接續要求輸入第1個至第n個學生的成績，n是剛才所輸入的學生人數。
- ▣ (5) 計算出人數、總分及平均值(不限制小數點位數)，顯示如執行結果參考畫面 (3)。

浮點數計算

34



□ 執行結果參考畫面：

▣ (1) 程式執行時，顯示「請輸入學生人數：」，要求輸入學生人數。

```
命令提示字元 - java JPA502
C:\Java>java JPA502
請輸入學生人數：
```

浮點數計算

35



□ 執行結果參考畫面：

- ▣ (2) 接續要求輸入第 1 個至第 n 個學生的成績，n 是剛才所輸入的學生人數。

```
命令提示字元 - java JPA502

C:\Java>java JPA502
請輸入學生人數：5
第1個學生的成績：
```

浮點數計算

36



- 執行結果參考畫面：
- (3) 計算出人數、總分及平均。

```
C:\Java>java JPA502
請輸入學生人數:5
第1個學生的成績:81.24
第2個學生的成績:56.14
第3個學生的成績:92.84
第4個學生的成績:42.96
第5個學生的成績:64.37
人數:5
總分:337.55
平均:67.509995

C:\Java>
```

浮點數計算

37



□ 設計說明：

```
1 import java.util.Scanner;
2 public class JPA502 {
3     public static Scanner keyboard = new Scanner(System.in);
4     public static void main(String args[]) {
5         System.out.print("請輸入學生人數：");
6         Scanner sc = new Scanner(System.in);
7         int poe = sc.nextInt();
8         float sum = 0;
9         float[] ps = new float[poe];
10        //迴圈次數決定可輸入幾個學生成績，而迴圈次數由使用者輸入
11        for(int a =0;a<poe;a++) {
12            System.out.print("第"+(a+1)+"個學生的成績：");
13            ps[a]=sc.nextFloat();
14            sum +=ps[a];
15        }
16        System.out.println("人數："+poe);
17        System.out.println("總分："+sum);
18        System.out.println("平均："+sum/poe);
19    }
20 }
```

二分搜尋法

38



□ 設計說明：

- 程式內有已排序資料{5, 9, 13, 15, 17, 19, 25, 30, 45}，請使用二分搜尋法尋找輸入的資料。
- 程式連續執行兩次，於程式執行時，如下頁所示，顯示【請輸入要找尋的資料：】要求輸入欲尋找的資料n。
- 若沒有搜尋到相符的數值，顯示【n不在陣列中】，將欲尋找的資料代入n，如下頁所示。
- 尋找時，列出尋找區間及此區間的中間值，搜尋幾次就列出幾項，最後產出【經過 y 次的尋找】，y代入搜尋次數；若有搜尋到相符的數值，值位在陣列中的第幾個位置，如下頁所示。

二分搜尋法



39

□ 執行結果參考畫面：

- (1) 程式執行時，顯示[請輸入要找尋的資料：]，並於後方要求輸入要找尋的資料。

```
D:\>java JPA510  
請輸入要找尋的資料：
```

- (2) 請輸出二分搜尋法的搜尋過程，並顯示[經過 y 次的尋找]，若資料不在陣列裡則顯示[n不在陣列中]，若在陣列裡則顯示[您要找的資料在陣列中的第x個位子]，執行結果如範例圖。

```
請輸入要找尋的資料：2  
尋找區間: 0(5)..8(45),中間: 4(17)  
尋找區間: 0(5)..3(15),中間: 1(9)  
尋找區間: 0(5)..0(5),中間: 0(5)  
經過 3 次的尋找  
2不在陣列中  
請輸入要找尋的資料：30  
尋找區間: 0(5)..8(45),中間: 4(17)  
尋找區間: 5(19)..8(45),中間: 6(25)  
尋找區間: 7(30)..8(45),中間: 7(30)  
經過 3 次的尋找  
您要的資料在陣列中的第7個位置
```

二分搜尋法

40



□ 設計說明：

```
1 import java.util.Scanner;
2 public class JPA510 {
3     public static Scanner targetboard = new Scanner(System.in);
4     static int time = 0;
5     public static void main(String[] argv) {
6         search();
7         time = 0;
8         search();
9     }
10    //搜尋方法，主要是在做排版的工作
11    public static void search() {
12        int[] data = {5, 9, 13, 15, 17, 19, 25, 30, 45}; // 已排序資料
13
14        System.out.print("請輸入要找尋的資料：");
15
16        int target = targetboard.nextInt();
17        int ans = binary_search(data, target, data.length);
18        if(ans == -1)
19            System.out.println("經過 "+time+" 次的尋找\n"+target+"不在陣列中");
20        else
21            System.out.println("經過 "+time+" 次的尋找\n您要的資料在陣列中的第"+ans+"個位置");
22    }
```


二分搜尋法

41



□ 設計說明：

```
23 //二分法搜尋方法，真正在搜尋的地方
24 //傳入值有三個，陣列，目標值，陣列長度
25⊖ static int binary_search(int[] data,int target, int max)
26 {
27     int middle, left, right;
28     left = 0; right = max-1; // 設定起始搜尋範圍：左邊界及右邊界（右邊界由最大值減1得到）
29     while (left <= right)
30     {
31         time++;
32         middle = (left + right) / 2; // 找出中間位置
33         System.out.printf("尋找區間: %d(%s)..%d(%s),中間: %d(%s)\n",
34             left,String.valueOf(data[left]),
35             right,String.valueOf(data[right]),
36             middle,String.valueOf(data[middle]));
37         if (target == data[middle]) return middle; // 找到資料，傳回找到之位置
38         // 調整搜尋範圍
39         if (target < data[middle]) // 往左半邊找（調整右邊界）
40             right = middle - 1;
41         else // 往右半邊找（調整左邊界）
42             left = middle + 1;
43     }
44     return -1; // 沒找到資料，傳回 -1
45 }
46 }
```

第六類-物件導向程式設計與例外處理



電腦零件設計

43



□ 題目一設計說明：

- ▣ (1) 請撰寫一個LCD類別。10吋的成本是2000元，15吋的成本是2500元，17吋的成本是3000元。
- ▣ (2) 請撰寫一個CPU類別。速度1.66的成本是6000元，速度2.2的成本是8000元，速度2.4的成本是11000元。
- ▣ (3) 請撰寫一個HD類別。120G的成本是2400元，160G的成本是2800元。
- ▣ (4) 請使用以上的零件，撰寫以下的類別。
- ▣ (5) 請寫一個MiniNote類別。一個MiniNote類別的筆電有一個10吋的LCD，一個速度1.66的CPU及一個120G的HD。

電腦零件設計

44



□ 題目一設計說明：

- ▣ (6) 請寫一個Note15類別。一個Note15類別的筆電有一個15吋的LCD，一個速度2.2的CPU及一個160G的HD。
- ▣ (7) 這兩型電腦的成本是其零件成本的1.4倍。定價則是其零件成本的2倍。
- ▣ (8) 請分別製造一個MinitNote及一個Note15，呼叫其getCost方法及getPrice方法，印出其成本及定價的傳回值

電腦零件設計

45



- 執行結果參考畫面：

A screenshot of a Windows command prompt window. The title bar at the top reads "命令提示字元" (Command Prompt). The window has standard Windows window controls (minimize, maximize, close) on the right. The main area is black with white text. The text shows the command "C:\Java>java JPA06_1" being executed, followed by two lines of output: "MiniNote cost:14559.999999999998, price:20800" and "Note15 cost:18620.0, price:26600". The prompt "C:\Java>" is shown again at the bottom.

```
C:\Java>java JPA06_1
MiniNote cost:14559.999999999998, price:20800
Note15 cost:18620.0, price:26600

C:\Java>
```

電腦零件設計

46



□ 題目一設計說明：

```
1 //建立一個共有的方法，以供其他方法繼承用
2 class Unit{
3     int cost;
4     Unit(){cost = 0;}
5     public int getcost(){return cost;}
6 }
7
8 //建立LCD的方法，內有初始化cost
9 class LCD extends Unit{
10     LCD(int i){
11         if(i==10)
12             cost=2000;
13         else if(i==15)
14             cost=2500;
15         else
16             cost=3000;
17     }
18 }
19 //建立CPU的方法，內有初始化cost
20 class CPU extends Unit{
21     CPU(double d ){
22         if(d==1.66)
23             cost=6000;
24         if(d==2.2)
25             cost=8000;
26         if(d==2.4)
27             cost=11000;
28     }
29 }
```

```
30 //建立HD的方法，內有初始化cost
31 class HD extends Unit {
32     HD(String s) {
33         if(s=="120G")
34             cost = 2400;
35         else
36             cost = 2800;
37     }
38 }
39 //建立一個Note抽象類別
40 abstract class Note{
41     double cost;
42     LCD l;
43     CPU c;
44     HD dd;
45     Note(int i,double d,String s){
46         l = new LCD(i);
47         c = new CPU(d);
48         dd = new HD(s);
49     }
50     abstract public double getCost();
51     abstract public double getPrice();
52 }
53 //建立一個小筆電的方法
54 class MiniNote extends Note{
55     //建構子，將值傳遞給父親
56     MiniNote(){super(10,1.66,"120G");}
57     //計算cost值，其中使用的物件由父親繼承下來
58     cost = l.getcost()+c.getcost()+dd.getcost();
59 }
60
61 public double getCost(){return cost*1.4;}
62 public double getPrice(){return cost*2.0;}
63 }
```

電腦零件設計

47



□ 題目一設計說明：

```
64 //建立一個15吋筆電的方法
65 class Note15 extends Note{
66     Note15(){
67         super(15,2.2,"160G");
68         cost = l.getcost()+c.getcost()+dd.getcost();
69     }
70
71     public double getCost(){return cost*1.4;}
72     public double getPrice(){return cost*2.0;}
73 }
74
75 class JPA06_1 {
76     public static void main(String args[]){
77         //產生小筆電的物件
78         MiniNote mininote = new MiniNote();
79         System.out.println("MiniNote cost:"+mininote.getCost()+", price:"+(int)mininote.getPrice());
80         //產生15吋筆電的物件
81         Note15 note15 = new Note15();
82         System.out.println("Note15 cost:"+note15.getCost()+", price:"+(int)note15.getPrice());
83     }
84 }
```

電腦零件設計

48



□ 題目二設計說明：

- (1) 請寫一個PC類別，一個PC類別的桌上型電腦有一個速度2.4的CPU 及一個160G的HD。製造一個PC類別的桌上型電腦的成本是「零件成本」加500，售價則為「零件成本」的1.8倍。
- (2) 另請寫一個MultiPC類別，一個MultiPC的超級電腦，可以有多顆2.4G的CPU及多顆160G的HD。MultiPC的成本為其「零件成本」的1.2倍，售價則為「零件成本」的1.8倍。
- (3) 請製造一個PC類別的桌上型電腦，一個2顆CPU及4顆HD的MultiPC及一個4顆CPU及8顆HD的MultiPC。
- (4) 分別呼叫其getCost方法及getPrice方法，印出其傳回值。

電腦零件設計

49



- 執行結果參考畫面：

A screenshot of a Windows command prompt window titled "命令提示字元". The window has a standard Windows interface with minimize, maximize, and close buttons in the top right corner. The background is black, and the text is white. The text shows the execution of a Java program named JPA06_2, which outputs cost and price information for different PC configurations.

```
C:\Java>java JPA06_2
PC cost:14300.0, price:24840.0
MultiPC: 2CPU, 4HD, cost:39840.0, price:59760.0
MultiPC: 4CPU, 8HD, cost:79680.0, price:119520.0

C:\Java>
```

電腦零件設計

50



□ 題目二設計說明：

```
1 class Unit {
2     int cost;
3     Unit(){
4         cost = 0;
5     }
6     public int getcost(){
7         return cost;
8     }
9 }
10
11 class LCD extends Unit {
12     LCD(int i){
13         if(i==10)
14             cost=2000;
15         else if(i==15)
16             cost=2500;
17         else
18             cost=3000;
19     }
20 }
21
22 class CPU extends Unit {
23     CPU(double d) {
24         if(d==1.66)
25             cost=6000;
26         if(d==2.2)
27             cost=8000;
28         if(d==2.4)
29             cost=11000;
30     }
31 }
```

```
33 class HD extends Unit {
34     HD(String s) {
35         if(s=="120G")
36             cost = 2400;
37         else
38             cost = 2800;
39     }
40 }
41
42 //建立公用的方法，以供繼承
43 abstract class PCandMultiPC {
44     double cost;
45     CPU l;
46     HD c;
47     PCandMultiPC(){
48         l = new CPU(2.4);
49         c = new HD("160G");
50     }
51     abstract public double getCost();
52     abstract public double getPrice();
53 }
54
55
56 class PC extends PCandMultiPC {
57     public double getCost(){
58         return (l.getcost()+c.getcost()+500);
59     }
60     public double getPrice(){
61         return (l.getcost()+c.getcost())*1.8;
62     }
63 }
64 }
```

電腦零件設計

51



□ 題目二設計說明：

```
66 class MultiPC extends PCandMultiPC {
67     double toa;
68     MultiPC(int a,int b) {
69         toa = a*l.getcost()+b*c.getcost();
70     }
71
72     public double getCost() {
73         return (toa*1.2);
74     }
75     public double getPrice() {
76         return (toa*1.8);
77     }
78 }
79
80
81 class JPA06_2 {
82     public static void main(String args[]){
83         PC pc = new PC();
84         System.out.println("PC cost:"+pc.getCost()+", price:"+pc.getPrice());
85         MultiPC multipc1 = new MultiPC(2, 4);
86         System.out.println("MultiPC: 2CPU, 4HD, cost:"+multipc1.getCost()+", price:"+multipc1.getPrice());
87         MultiPC multipc2 = new MultiPC(4, 8);
88         System.out.println("MultiPC: 4CPU, 8HD, cost:"+multipc2.getCost()+", price:"+multipc2.getPrice());
89     }
90 }
```

電腦零件設計

52



□ 題目三設計說明：

- ▣ (1) 請比較一個Note15的筆電和一個PC類別的桌上型電腦何者售價較貴。
- ▣ (2) 請新增一個名為AIIPC的類別，於此類別中撰寫isExpensive方法，此方法能夠比較何者的售價較貴，並傳回一個布林值。顯示如執行結果參考畫面。

電腦零件設計

53



- 執行結果參考畫面：

A screenshot of a Windows Command Prompt window. The title bar reads "命令提示字元". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows the following text:

```
C:\Java>java JPA06_3  
Note15 is more expensive then PC  
  
C:\Java>
```

電腦零件設計

54



□ 題目三設計說明：

```
1 class Unit {
2     int cost;
3     Unit(){
4         cost = 0;
5     }
6     public int getcost(){
7         return cost;
8     }
9 }
10
11 class LCD extends Unit{
12     LCD(int i){
13         if(i==10)
14             cost=2000;
15         else if(i==15)
16             cost=2500;
17         else
18             cost=3000;
19     }
20 }
21
22 class CPU extends Unit{
23     CPU(double d) {
24         if(d==1.66)
25             cost=6000;
26         if(d==2.2)
27             cost=8000;
28         if(d==2.4)
29             cost=11000;
30     }
31 }
```

```
33 class HD extends Unit{
34     HD(String s){
35         if(s=="120G")
36             cost = 2400;
37         else
38             cost = 2800;
39     }
40 }
41
42 abstract class Note{
43     double cost;
44     LCD l;
45     CPU c;
46     HD dd;
47     Note(int i,double d,String s){
48         l = new LCD(i);
49         c = new CPU(d);
50         dd = new HD(s);
51     }
52     abstract public double getCost();
53     abstract public double getPrice();
54 }
55
56 class MiniNote extends Note{
57     MiniNote() {
58         super(10,1.66,"120G");
59         cost = l.getcost()+c.getcost()+dd.getcost();
60     }
61     public double getCost(){
62         return cost*1.4;
63     }
64     public double getPrice(){
65         return cost*2.0;
66     }
67 }
```

電腦零件設計

55



□ 題目三設計說明：

```
69 class Note15 extends Note{
70     Note15(){
71         super(15,2.2,"160G");
72         cost = l.getcost()+c.getcost()+dd.getcost();
73     }
74     public double getCost(){return cost*1.4;}
75     public double getPrice(){return cost*2.0;}
76 }
77
78 abstract class PCandMultiPC{
79     double cost;
80     CPU l;
81     HD c;
82     PCandMultiPC() {
83         l = new CPU(2.4);
84         c = new HD("160G");
85     }
86     abstract public double getCost();
87     abstract public double getPrice();
88 }
89
90 class PC extends PCandMultiPC {
91     public double getCost(){
92         return (l.getcost()+c.getcost()+500);
93     }
94     public double getPrice(){
95         return (l.getcost()+c.getcost())*1.8;
96     }
97 }
```

電腦零件設計

56



□ 題目三設計說明：

```
99 class MultiPC extends PCandMultiPC {
100     double total;
101     MultiPC(int a,int b){
102         total = a*l.getcost()+b*c.getcost();
103     }
104     public double getCost(){
105         return (total*1.2);
106     }
107     public double getPrice(){
108         return (total*1.8);
109     }
110 }
111 //建立一個AllPC的方法
112 class AllPC {
113     //將傳進來的物件，取得其成本的方法
114     double a1,a2;
115     AllPC(PCandMultiPC p,Note n) {
116         a1=p.getCost();
117         a2=n.getCost();
118     }
119     //設計一個比較方法，回傳布林值
120     public boolean isExpensive() {
121         if(a1>a2)
122             return true;
123         else
124             return false;
125     }
126 }
```


電腦零件設計

57



□ 題目三設計說明：

```
129 class JPA06_3 {  
130     public static void main(String args[]) {  
131         PC pc = new PC();  
132         Note15 note15 = new Note15();  
133         //產生一個物件，放入兩個物件去比較  
134         AllPC app = new AllPC(pc, note15);  
135         if(app.isExpensive())  
136             System.out.println("PC is more expensive then Note15 ");  
137         else  
138             System.out.println("Note15 is more expensive then PC ");  
139     }  
140 }
```

電腦零件設計

58



□ 題目四設計說明：

- (1) 請使用LinkedList為這家工廠寫一個Order類別。
- (2) 此筆訂單的資料如下：
 - MiniNote: 1台
 - Note15: 1台
 - PC: 1台
- (3) 請寫一revenue方法，輸出此訂單的總收入。

電腦零件設計

59



- 執行結果參考畫面：

A screenshot of a Windows command prompt window. The title bar at the top reads "命令提示字元" (Command Prompt). The window has a black background with white text. The text shows the command "C:\Java>java JPA06_4" being entered, followed by the output "72240.0". The prompt "C:\Java>" is shown again on the next line. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
C:\Java>java JPA06_4
72240.0

C:\Java>
```

電腦零件設計

60



□ 題目四設計說明：

```
1 import java.util.Iterator;
2 import java.util.LinkedList;
3
4 class Unit{
5     int cost;
6     Unit(){
7         cost = 0;
8     }
9     public int getcost(){
10         return cost;
11     }
12 }
13
14 class LCD extends Unit {
15     LCD(int i){
16         if(i==10)
17             cost=2000;
18         else if(i==15)
19             cost=2500;
20         else
21             cost=3000;
22     }
23 }
24
25 class CPU extends Unit {
26     CPU(double d ) {
27         if(d==1.66)
28             cost=6000;
29         if(d==2.2)
30             cost=8000;
31         if(d==2.4)
32             cost=11000;
33     }
34 }
```

```
36 class HD extends Unit {
37     HD(String s) {
38         if(s=="120G")
39             cost = 2400;
40         else
41             cost = 2800;
42     }
43 }
44
45 abstract class Note extends AllPC{
46     double cost;
47     LCD l;
48     CPU c;
49     HD dd;
50     Note(int i,double d,String s){
51         l = new LCD(i);
52         c = new CPU(d);
53         dd = new HD(s);
54     }
55     abstract public double getCost();
56     abstract public double getPrice();
57 }
58
59 class MiniNote extends Note{
60     MiniNote(){
61         super(10,1.66,"120G");
62         cost = l.getcost()+c.getcost()+dd.getcost();
63     }
64
65     public double getCost(){
66         return cost*1.4;
67     }
68     public double getPrice(){
69         return cost*2.0;
70     }
71 }
```

電腦零件設計

61



□ 題目四設計說明：

```
73 class Note15 extends Note{
74     Note15() {
75         super(15,2.2,"160G");
76         cost = l.getcost()+c.getcost()+dd.getcost();
77     }
78
79     public double getCost(){
80         return cost*1.4;
81     }
82     public double getPrice(){
83         return cost*2.0;
84     }
85 }
86
87 abstract class PCandMultiPC extends AllPC {
88     double cost;
89     CPU l;
90     HD c;
91     PCandMultiPC() {
92         l = new CPU(2.4);
93         c = new HD("160G");
94     }
95     abstract public double getCost();
96     abstract public double getPrice();
97 }
98
99 class PC extends PCandMultiPC {
100     public double getCost() {
101         return (l.getcost()+c.getcost()+500);
102     }
103     public double getPrice() {
104         return (l.getcost()+c.getcost())*1.8;
105     }
106 }
```

```
108 class MultiPC extends PCandMultiPC {
109     double total;
110     MultiPC(int a,int b){
111         total = a*l.getcost()+b*c.getcost();
112     }
113
114     public double getCost(){
115         return (total*1.2);
116     }
117     public double getPrice(){
118         return (total*1.8);
119     }
120 }
121
122 abstract class AllPC{
123     AllPC(){}
124     abstract public double getCost();
125     abstract public double getPrice();
126 }
127 }
```

電腦零件設計

62



□ 題目四設計說明：

```
129 //建立一個訂單的方法
130 class Order{
131     //使用LinkedList來儲存每筆電腦物件
132     LinkedList pcs;
133     //利用建構子來建立LinkedList物件
134     Order(){
135         pcs = new LinkedList();
136     }
137     void in(AllPC allpc){
138         pcs.add(allpc);
139     }
140     public double revenue() {
141         double d = 0.0;
142         //利用iterator來依序讀取LinkedList內的物件
143         for(Iterator iterator = pcs.iterator(); iterator.hasNext();) {
144             //將下一個物件讀入
145             AllPC allpc = (AllPC)iterator.next();
146             d=d+allpc.getPrice();
147         }
148         return d;
149     }
150 }
151 }
152
153 class JPA06_4 {
154     public static void main(String args[]){
155         //建立訂單物件
156         Order ord = new Order();
157         //寫入訂單
158         ord.in(new MiniNote());
159         ord.in(new Note15());
160         ord.in(new PC());
161         System.out.println(ord.revenue());
162     }
163 }
```

電腦零件設計

63



□ 題目五設計說明：

▣ (1) 此筆訂單資料如下：

■ MiniNote 1台

■ Note15 1台

■ PC 1台

▣ (2) 利潤 = 收入-成本(非零件總成本)

▣ (3) 如果此次累積利潤超過 20000，則印出信息 [This order exceeds 20000:xx]，將累積利潤代入xx。

電腦零件設計

64



- 執行結果參考畫面：

A screenshot of a Windows command prompt window. The title bar reads "命令提示字元" (Command Prompt). The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows the following text:

```
C:\Java>java JPA06_5  
This order exceeds 20000:24760.0  
C:\Java>
```


電腦零件設計

65



□ 題目五設計說明：

```
1 import java.util.Iterator;
2 import java.util.LinkedList;
3
4 class Unit{
5     int cost;
6     Unit(){
7         cost = 0;
8     }
9     public int getcost(){
10         return cost;
11     }
12 }
13
14 class LCD extends Unit{
15     LCD(int i){
16         if(i==10)
17             cost=2000;
18         else if(i==15)
19             cost=2500;
20         else
21             cost=3000;
22     }
23 }
24
25 class CPU extends Unit{
26     CPU(double d){
27         if(d==1.66)
28             cost=6000;
29         if(d==2.2)
30             cost=8000;
31         if(d==2.4)
32             cost=11000;
33     }
34 }
```

```
36 class HD extends Unit{
37     HD(String s){
38         if(s=="120G")
39             cost = 2400;
40         else
41             cost = 2800;
42     }
43 }
44
45 abstract class Note extends AllPC{
46     double cost;
47     LCD l;
48     CPU c;
49     HD dd;
50     Note(int i,double d,String s){
51         l = new LCD(i);
52         c = new CPU(d);
53         dd = new HD(s);
54     }
55     abstract public double getCost();
56     abstract public double getPrice();
57 }
58
59 class MiniNote extends Note{
60     MiniNote() {
61         super(10,1.66,"120G");
62         cost = l.getcost()+c.getcost()+dd.getcost();
63     }
64
65     public double getCost(){
66         return cost*1.4;
67     }
68     public double getPrice(){
69         return cost*2.0;
70     }
71 }
```

電腦零件設計

66



□ 題目五設計說明：

```
73 class Note15 extends Note{
74     Note15(){
75         super(15,2.2,"160G");
76         cost = l.getcost()+c.getcost()+dd.getcost();
77     }
78
79     public double getCost(){
80         return cost*1.4;
81     }
82     public double getPrice(){
83         return cost*2.0;
84     }
85 }
86
87 abstract class PCandMultiPC extends AllPC{
88     double cost;
89     CPU l;
90     HD c;
91     PCandMultiPC() {
92         l = new CPU(2.4);
93         c = new HD("160G");
94     }
95     abstract public double getCost();
96     abstract public double getPrice();
97 }
98
99 class PC extends PCandMultiPC{
100     public double getCost(){
101         return (l.getcost()+c.getcost()+500);
102     }
103     public double getPrice(){
104         return (l.getcost()+c.getcost())*1.8;
105     }
106 }
```

```
108 class MultiPC extends PCandMultiPC{
109     double toa;
110     MultiPC(int a,int b){
111         toa = a*l.getcost()+b*c.getcost();
112     }
113
114     public double getCost(){
115         return (toa*1.2);
116     }
117     public double getPrice(){
118         return (toa*1.8);
119     }
120 }
121
122 abstract class AllPC{
123     AllPC(){}
124     abstract public double getCost();
125     abstract public double getPrice();
126 }
```

電腦零件設計

67



□ 題目五設計說明：

```
128 class Order{
129     LinkedList pcs;
130     Order(){
131         pcs = new LinkedList();
132     }
133     void in(AllPC allpc){
134         pcs.add(allpc);
135     }
136     public double revenue(){
137         double d =0.0;
138         for(Iterator iterator = pcs.iterator();iterator.hasNext();) {
139             AllPC allpc = (AllPC)iterator.next();
140             d=d+allpc.getPrice();
141         }
142         return d;
143     }
144 }
145 }
```

電腦零件設計

68



□ 題目五設計說明：

```
147 class JPA06_5 {
148     public static void main(String args[]){
149         //產生物件
150         Order ord = new Order();
151         MiniNote m =new MiniNote();
152         Note15 n =new Note15();
153         PC p =new PC();
154         //加入訂單中
155         ord.in(m);
156         ord.in(n);
157         ord.in(p);
158         //計算總成本
159         double totalcost = m.getCost()+n.getCost()+p.getCost();
160         //價算利潤
161         double profit = ord.revenue() - totalcost;
162         if(profit>20000)
163             System.out.printf("This order exceeds 20000:%.1f",profit);
164     }
165 }
166 }
```

Q & A