

控制敘述 - 選擇

- if 選擇敘述
- if ... else 選擇敘述
- if ... else if ... else 選擇敘述
- switch 選擇敘述
- 巢狀結構
- 認證實例演練

3.1 前言

「敘述區段」是指連續多行敘述的集合，當程式執行時會由上往下逐行執行的敘述區段。下次重新執行時仍會維持相同流程，所得結果都是一樣的，此方式只能設計出簡單的程式。像這樣的程式，就稱為「循序結構」。實際上設計程式並非都如此單純，常會因應程式需求而改變程式執行的流向，此種稱為「選擇結構」。或是程式中某些程式區段需要重複執行多次，這就是「重複結構」。

「循序結構」和「選擇結構」中，必須透過各種「控制敘述」(control statements)來達成。設計程式時要能靈活運用控制敘述，就必須具備良好的邏輯觀念，才能設計出具有彈性且複雜的程式。Java 語言提供選擇敘述、重複敘述、分支敘述等三類控制敘述，將在本章和下一章陸續介紹。

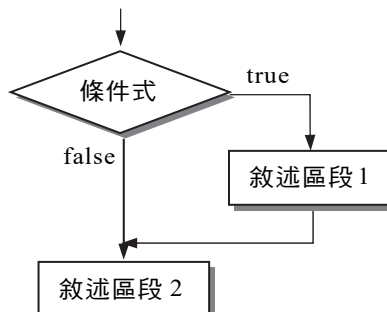
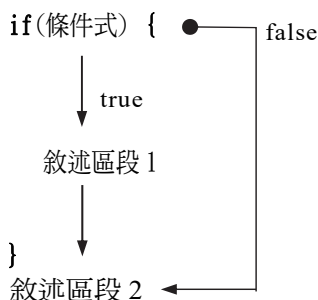
3.2 if 選擇敘述

所謂「選擇敘述」就是讓程式執行此敘述時能按照條件，來決定程式執行的流向。在 Java 中提供 if 和 switch 兩種選擇敘述。if 如同字面上所看到的，它的意思就是「如果」。當在寫程式的時候希望流程是「如果...就做...」，此時

就可以使用 if 敘述來達成這個條件。Java 中 if 選擇敘述總共分為單向、雙向、多向和巢狀等四種型式，可以依照不同的需求套用。

3.2.1 if 單向式選擇敘述的使用

單向式選擇是最簡單的敘述，只使用一個 if 做為判斷的條件，其語法如下：



說明

- 條件式是由運算式構成，若條件式含有多個關係式時，可以使用適當的邏輯運算子來連接。譬如：

① x 大於零且是偶數，條件式的寫法：

```
if ((x > 0) && (x % 2 == 0))
```

② 若希望判斷輸入的字元(ch)不是數字，其 if 條件式寫法為：

```
if (ch < '0' || ch > '9')
```

- 若條件式的結果為 true，會執行接在 if 敘述後左、右大括號 { } 內的 <敘述區段 1>，接著再執行 <敘述區段 2>；
 - 若條件式結果為 false，則會跳過 if 內的 <敘述區段 1> 不執行，直接執行 <敘述區段 2>。
- 若 <敘述區段 1> 有多行敘述時，必須用大括號 { } 括起來。若只有一行敘述，則可省略大括號。



簡例 奇偶數判斷

檔名：\ex03\src\ex03\Ex03_01.java

```
01 package ex03;
```

```
02 public class Ex03_01 {
```

```

03 public static void main(String[] args) {
04     int num = 3;
05     String even = "奇數";
06     if (num % 2 == 0) {
07         even = "偶數";
08     }
09     System.out.println(num + " 為" + even);
10 }
11 }

```



說明

1. 第 5 行：預設 `even` 字串變數值為 "奇數"。
2. 第 6~8 行：使用 `if` 單向式選擇敘述，當 `num % 2 == 0` 條件為真，就執行第 7 行的敘述，設 `even` 字串變數值為 "偶數"。
3. 第 9 行：顯示奇偶數的判斷結果，執行結果為「3 為奇數」。





實作

試設計一個購物節優惠程式，消費金額超過 1,000 元的部分打九折優惠。例如：消費 3,000 元，其中 2,000 元可打九折變成 1,800 元，實付 1,000 元 + 1,800 元 = 2,800 元。



結果

請輸入消費金額：570 
實付金額：570 元

請輸入消費金額：1570 
實付金額：1513 元



程式碼

檔名：\ex03\src\ex03\If.java

```

01 package ex03;
02 import java.util.Scanner;
03 public class If {
04     public static void main(String[] args) {
05         Scanner scn = new Scanner(System.in);
06         System.out.print("請輸入消費金額：");
07         int money = scn.nextInt(); // 取得輸入的金額並轉成 int 資料型別
08         if(money>1000) { // 如果金額大於 1000 時
09             money = 1000 + (int)((money - 1000) * 0.9); //超出部分打九折
10     }

```

```

11     System.out.printf("實付金額：%d 元", money);
12     scn.close();
13 }
14 }

```



說明

1. 第 7 行：使用 `Scanner` 類別物件 `scn` 的 `nextInt()` 方法，來接受使用者所輸入的資料並轉為 `int` 資料型別。
2. 第 8~10 行：為單向 `if` 選擇敘述，判斷的條件為「`money > 1000`」運算式，表示 `money` 大於 1000。如果運算式為 `true` 就執行第 9 行敘述；若為 `false` 就直接執行第 11 行敘述，此時因為沒有執行第 9 行敘述，所以金額將不會改變。
3. 第 9 行：實付金額等於不打折的 1000 元，加上打九折的超出部分，就是實付的金額。因為超出的部分乘以 0.9 後會自動轉型為浮點數，要用 `(int)` 強制轉型為整數。
4. 第 8~10 行：因為第 9 行是單行敘述，所以 `if` 選擇敘述的大括號 `{ }` 可以省略，可以改寫為：

```

if (money>1000)           // 如果金額大於 1000 時
    money = 1000 + (int)((money - 1000) * 0.9); // 超出部分打九折

```

5. 第 11 行：使用 `printf()` 方法，以格式化方式顯示實付的金額。

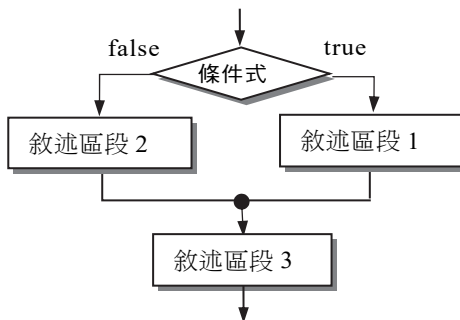
3.2.2 if 雙向式選擇敘述的使用

如果希望程式達到「如果...就進行...，否則才做...」，單向式 `if` 就無法處理，此時就需要採用雙向式的 `if...else` 語法來完成，其語法如下：

```

if (條件式) {
    敘述區段 1
    .....
} else {
    敘述區段 2
    .....
}
敘述區段 3

```



雙向式 if 選擇敘述，除了 if 外又多了一個 else 敘述，else 如同字面的意義就是「否則」。else 不能單獨使用，必須配合 if 一起使用。當 <條件式> 的結果為真 (true)，就執行 <敘述區段 1>；如果結果是假 (false)，就執行 <敘述區段 2>。請注意 <敘述區段 1> 和 <敘述區段 2> 兩個不同的敘述區段，只能有一個敘述區段被執行。這種雙向式的 if-else 比單向式的 if 更具彈性，但是不管條件是否滿足，最後都會回到接在 else 敘述區段後面的 <敘述區段 3> 敘述，然後繼續往下執行。雙向式 if 選擇敘述，如果是將結果指定給一個變數時，也可以使用條件運算子 ? 和 : 來撰寫，程式碼會比較簡潔，語法為：

```
變數 = 條件式 ? 運算式 1 : 運算式 2 ;
```



簡例 兩數中取最大值

檔名：\ex03\src\ex03\Ex03_02.java

```
int max, n1=2, n2=3;
if (n1 > n2) { // 使用 if-else 雙向條件式
    max = n1;
} else {
    max = n2;
}
max = (n1 > n2) ? n1 : n2; // 使用條件運算子的寫法
```



實作 詢問使用者是否為會員，若輸入的字串為「Y」，則顯示正確訊息，否則顯示錯誤訊息。



結果

```
請問是否為會員:(請輸入 Y 或 N) N
無法使用本系統!!
```

```
請問是否為會員:(請輸入 Y 或 N) Y
歡迎使用本系統!!
```



程式碼

檔名：\ex03\src\ex03\ElseIf.java

```
01 package ex03;
02 import java.util.Scanner;
03 public class ElseIf {
04     public static void main(String[] args) {
```

```
05     Scanner scn = new Scanner(System.in);
06     String member;
07     System.out.print("請問是否為會員:(請輸入 Y 或 N) ");
08     member = scn.next();
09     if (member.equals("Y")) {
10         System.out.println("歡迎使用本系統!!");
11     } else {
12         System.out.println("無法使用本系統!!");
13     }
14     scn.close();
15 }
16 }
```



說明

1. 第 9~13 行：是雙向式 if 選擇敘述，如果 member 等於"Y"，就執行第 10 行敘述區段；否則就執行第 12 行的敘述區段。
2. 第 9 行：字串的比較必須使用 equals() 方法，不能直接用 == 來判斷。
3. 第 10 和 12 行：在這個選擇結構中只包含單行敘述，所以其前後的 {...} 可以省略。雖然 {} 可以省略但是仍建議使用 {} 將敘述區段包括住，因為如此程式碼層次較明確，可提高可讀性也不易出錯。



NOTE

在 Java 中若要比較字串是否相同時，必須使用 String 類別的 equals() 方法。若要檢查 member 字串變數是否等於"Y"，應該使用 member.equals("Y")，不可以使用 member == "Y"，執行結果會是錯誤。

若要忽略大小寫的差異時，可以使用 equalsIngoreCase() 方法。如果使用 member.equalsIgnoreCase("Y")，則等於"Y"或"y"結果都會為真。

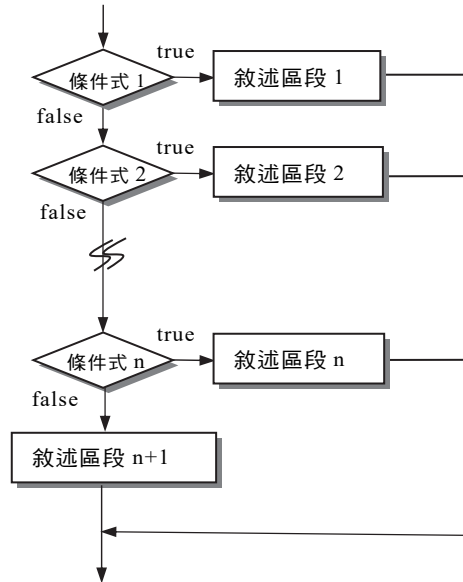
3.2.3 if 多向式選擇敘述的使用

程式設計時如果希望做到「如果...就進行...，否則就去做...，如果兩個條件都不成立，就去...」，也就當判斷條件不只兩項情形比較複雜時，就需使用 if...else if...else 多向式選擇敘述來處理。其語法如下：

```

if (條件式 1) {
    敘述區段 1
    .....
} else if (條件式 2) {
    敘述區段 2
    .....
    .....
} else if (條件式 n) {
    敘述區段 n
    .....
} else {
    敘述區段 n+1
    .....
}

```



if ...else if...else 多向式選擇敘述，多了 else if 敘述來增加判斷的條件，不管有多少個條件，只要多加 else if 就可以處理。但和 if...else 雙向式選擇敘述一樣，只允許執行其中某一個敘述區段。執行時會由上向下逐一檢查條件，一旦符合某條件就執行對應的敘述區段，接著跳離整個多向式選擇敘述。注意最後是 else，可以處理以上條件都不滿足時的情形。


簡例 依時間顯示適當的問候語

檔名：\ex03\src\ex03\Ex03_03.java

```

int hours = 22;
if (hours <= 10) {
    System.out.println("早安!");
} else if (hours >= 20) {
    System.out.println("晚安!");
} else {
    System.out.println("您好!");
}

```

 **實作** 設計一個水費查詢程式。當用水度數在 1~10 度，每度單價為 7.35 元；度數在 11~30 度，每度單價為 9.45 元；度數在 31~50 度，每度單價為 11.55 元；度數在 50 度以上，每度單價為 12.075 元。



說明

- 第 8~18 行：為巢狀結構，外層用 if...else 選擇結構來判斷 score 是否介於 0~100，正確時執行 9~15 行敘述其中有內層的 switch 選擇結構；不正確時執行第 17 行敘述。
- 第 10~14 行：為 switch 選擇結構根據(score / 10)運算式的值，分別執行不同的 case 值敘述區段。因為採用新語法，所以允許多個 case 值共用一個敘述區段。
- 第 9~14 行：此段敘述使用賦值方式可以改寫如下：

```
int prize = switch (score / 10) {
    case 9, 10 -> 500;
    case 8 -> 200;
    default -> 0;
};
```

3.4 認證實例演練



題目

- 某俱樂部加入的資格如下，請使用下列選項(可不用或多次使用)來完成 Java 程式。

- 年齡若超過 65 歲至少需捐助 10000 元
- 年齡若超過 21 歲至少需捐助 25000 元

(A) && (B) || (C) < (D) <= (E) > (F) >= (G) == (H) !=

```
if(ages ① 65 ② money ③ 10000 ④ )
    ages ⑤ 21 ⑥ money ⑦ 25000 ){
    System.out.println("資格符合");
}
else{
    System.out.println("資格不符合");
}
```

說明

根據題目 if 敘述寫法為：if (ages >= 65 && money >= 10000 || ages >= 21 && money >= 25000))，所以答案①(F)、②(A)、③(F)、④(B)、⑤(F)、⑥(A)、⑦(F)，程式碼請參考 Ex03_T01.java。



二. 你正在寫一個計算博物館入場費的 JAVA 程式。一般入場券訂價為 100 元，加入會員會費 400 元。此程式必須符合以下要求：

1. 會員入場券打七折。
2. 超過 65 歲的會員免費入場。
3. 非會員除可選擇購買入場券外，也可以選擇加入會員。

```
int countPrice(boolean isMem, int ages, boolean addMem) {
    int price;
    if (isMem) {
        ①
        price = 0;
        ②
        price = 70;
    }
    ③
    price = 470;
    ④
    price = 100;
}
return price;
}
```

① 填入的敘述為：

(A) if (ages > 65) (B) else if (ages > 65) (C) elseif (ages > 65) (D) else {

② 填入的敘述為：

(A) if (ages > 65) (B) else if (ages > 65) (C) elseif (ages > 65) (D) else {

③ 填入的敘述為：

(A) if (addMem) (B) else if (addMem) (C) elseif (addMem) (D) else {

④ 填入的敘述為：

(A) if (addMem) (B) else if (addMem) (C) elseif (addMem) (D) else {

說明

根據題意答案為①(A)、②(D)、③(B)、④(D)，程式碼請參考 Ex03_T02.java。



三. 你正在面試 JAVA 開發人員的工作，需呈現你對選擇敘述的能力。下列程式碼片段是否能夠正確區分成三種狀態，請填「是」或「否」。

與 ChatGPT 協作開發

- 認識 ChatGPT
- ChatGPT 聊天初體驗
- ChatGPT 協作開發 Java 程式

17.1 認識 ChatGPT

「ChatGPT」AI 聊天機器人於 2022 年 11 月 30 日推出後，由於能夠快速自動處理文字生成、問題解答、摘要…等多重任務，所以瞬間爆紅僅上線兩個月就擁有上億使用者。由於 ChatGPT 的崛起，而且還不斷擴充其學習能力，人類因此產生可能被 AI (人工智慧) 取代的焦慮。

17.1.1 ChatGPT 是什麼

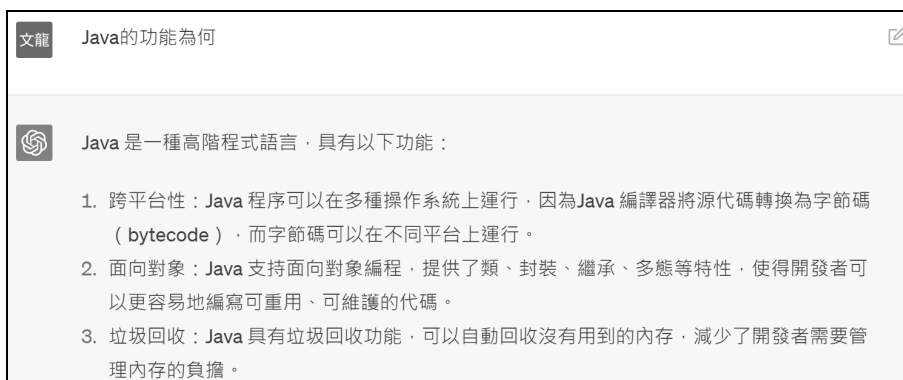
ChatGPT (Chat Generative Pre-trained Transformer，聊天生成型預訓練變換模型)，是由 OpenAI 人工智慧研究實驗室所開發的人工智慧聊天機器人程式。2022 年 11 月所推出的版本，是根據 GPT-3.5 架構的模型，以監督、強化、深度…等學習所訓練而成，目前 GPT-4.0 版已經推出 (版本會隨時更新)。



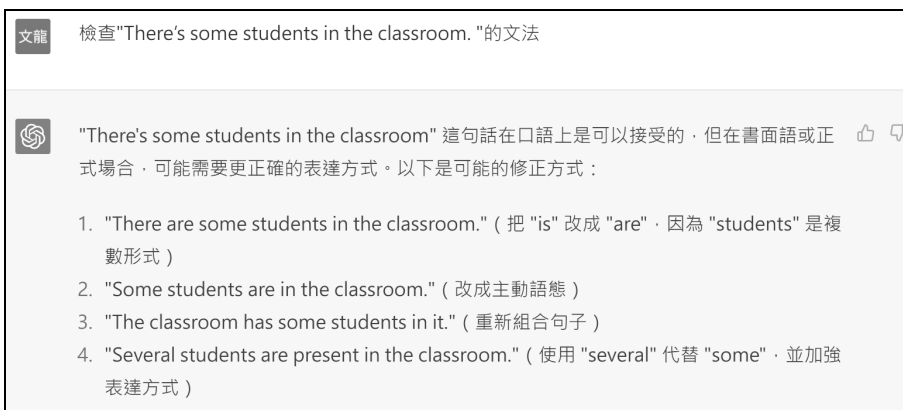
ChatGPT 主要是以文字方式，用人類自然的對話方式來進行互動，可以完成複雜的工作，例如根據輸入的提問回覆可能的答案，或根據輸入的條件產生指定的企劃文案，甚至還能編寫電腦程式。使用者註冊登入 ChatGPT 後，推廣

期間可以免費與 AI 機器人進行對話。OpenAI 會根據使用者運用的情形，繼續訓練和微調 ChatGPT，使得其功能和正確性可以不斷地增強。付費版 ChatGPT Plus，每月只需支付 20 美元，就可以享受優先使用權和更快的載入時間。

1. **基本問答**：詢問各種領域的基本知識，或是進行翻譯，例如：「Java 的功能為何」、「將我愛妳翻譯成日文」、「推薦五部適合情人節觀看的電影」、「跟兔子有關的笑話」。



2. **進階問答**：可以進一步詢問需要整理、分析的問題，例如：「使用 Java 設計學生成績管理系統的步驟」、「檢查 "There's some students in the classroom." 的文法」、「分析西遊記中孫悟空的個性」。



3. **創意發想**：提供各種具體條件後，由 ChatGPT 來構思和完成創意性工作，例如：「規劃由台中出發到花蓮三日遊的行程」、「以雨、霧、太陽、微風為主題撰寫情歌歌詞」、「設計國小高年級體驗省水重要性的教案」、「寫一篇關於友情和團隊合作的童話故事」。



4. **情境模擬**：ChatGPT 可以模擬指定風格、角色，來解決、建議或完成工作，例如：「請你扮演資深 CEO 為手工餅乾店構思 3 個商業模式」、「請以周杰倫風格創作一首檸檬汽水的廣告歌曲」。

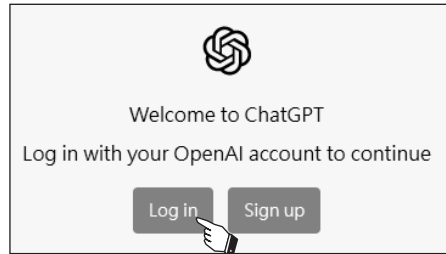




5. **程式開發**：ChatGPT 也可以進程式開發，例如讀程式碼、讀程式碼並幫程式加上註解、解 Bug 與重構程式碼。例如：「請提供 Java 大樂透程式碼」、「提供 Java 程式碼產品類別擁有編號、品名、單價成員」、「設計 Java 視窗程式擁有帳號、密碼與登入按鈕」。

17.2 ChatGPT 聊天初體驗

了解 ChatGPT 的功能和優缺點後，接著說明使用 ChatGPT 的基本步驟。

- Step 1 進入 ChatGPT 網站：**開啟瀏覽器進入 ChatGPT 網站，輸入網址：<https://chat.openai.com/auth/login>，然後點按「Log in」登入。



- Step 2 提問：**輸入問題內容後按  鍵，或是按  向 ChatGPT 提問。例如輸入「有一整數陣列 [56,78,54,23,90]，請提供 Java 程式碼將上述整數陣列進行由小到大排序。」：

有一整數陣列 [56,78,54,23,90]，請提供Java程式碼將上述整數陣列進行由小到大排序。

ChatGPT 回答使用 `Array.Sort()`方法進行排序整數陣列，如下：(ChatGPT 回答的內容可能每次都不同)

```
java Copy code

import java.util.Arrays;

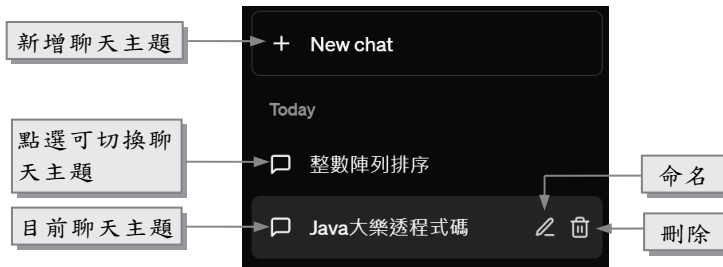
public class SortArray {
    public static void main(String[] args) {
        int[] arr = {56, 78, 54, 23, 90};
        Arrays.sort(arr);
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

Step 3 追問：對於 ChatGPT 的回答滿意，可以複製下來做進一步運用。如果不滿意可以按「Regenerate response」，請 ChatGPT 重新回答。也可以將題目再詳細說明，尤其是比較複雜的程式可分拆成幾個步驟單元分別提問。例如延續上例再輸入追問「請使用氣泡排序法排序」：



提問的內容越詳細，ChatGPT 的回答就會越精確，如果有需要可以繼續提問。

Step 4 管理聊天主題：提問後 ChatGPT 會自動為聊天主題命名，可以點按  修改名稱，也可以點按  刪除目前聊天主題，如果要刪除所有聊天主題可點按左邊的  **Clear conversations** 指令。若要改變聊天的主題時，可以點按左上角的「+ New chat」新增一個聊天主題。



Step 5 **登出 ChatGPT 網站**：若要離開 ChatGPT，可以點按左下角的「Log out」登出，然後關閉 ChatGPT 網頁。

17.3 ChatGPT 協作開發 Java 程式

ChatGPT 的功能強大，在前面已經略加介紹，本節將聚焦在如何使用 ChatGPT 來協助開發人員撰寫 Java 程式碼，使快速生成 Java 程式碼讓開發上更加方便。但要注意的是 ChatGPT 所產生的程式碼並不一定正確，必須經過偵錯才能確認。如果程式碼有錯，除了可以自行修改外，也可以重新審視提問內容是否正確、周延，修正後再重新提問。



實作 請使用 ChatGPT 生成 Java 的大樂透號碼程式碼。



結果

大樂透號碼：15 29 36 9 44 10

向 ChatGPT 詢問「請使用 Java 產生大樂透號碼的程式」，結果 ChatGPT 生成下圖 Java 程式碼，請依需求複製對應程式碼。



本例生成 Java 類別名稱 Lotto，因此請將程式檔名命名為 Lotto.java。(注意 ChatGPT 所產生的程式碼可能不同)



檔名：\ex17\src\LottoDemo\Lotto.java

```
01 package LottoDemo;
02 import java.util.Random;
03 public class Lotto {
04     public static void main(String[] args) {
05         //建立 Random 類別物件 random 來產生亂數
06         Random random = new Random();
07         //使用 numbers 陣列來記錄產生的 6 個號碼
08         int [] numbers = new int [6];
09
10         System.out.print ("大樂透號碼：");
11         //產生 1 到 49 之間的亂數，確保每個號碼只出現一次
12         for (int i = 0; i < 6; i++) {
13             int num = random.nextInt (49) + 1;
14             for (int j = 0; j < i; j++) {
15                 if (numbers [j] == num) {
16                     num = random.nextInt (49) + 1;
17                     j = -1;
18                 }
19             }
20             numbers [i] = num;
21             System.out.print (num + " ");
22         }
23     }
24 }
```



說明

1. ChatGPT 生成的 Java 程式碼使用 Random 類別來產生亂數，並使用 numbers 陣列來記錄產生的大樂透 6 個號碼。程式會產生 1 到 49 之間的亂數，同時使用巢狀迴圈檢查，以確保每個號碼只會出現一次，最後輸出隨機產生的大樂透號碼。

**實作**

使用 ChatGPT 生成產品類別有編號、品名、單價欄位。並使用產品類別生成 5 筆產品記錄，最後將 5 筆產品記錄顯示出來。

**結果**

```
1, Product A, 10.0
2, Product B, 20.0
3, Product C, 30.0
4, Product D, 40.0
5, Product E, 50.0
```

向 ChatGPT 詢問「請提供 Java 程式碼。產品類別有編號、品名、單價欄位，請使用產品類別建立 5 筆產品記錄，並將 5 筆產品記錄顯示出來。」，結果 ChatGPT 生成下圖 Java 程式碼，請依需求複製對應程式碼。(注意 ChatGPT 所產生的程式碼可能不同)

以下是一個簡單的 Java 程序碼，實現了產品類別的定義、建立 5 筆產品記錄和顯示所有產品記錄的功能：

```
java
public class Product {
    private int id;
    private String name;
    private double price;

    public Product(int id, String name, double price) {
        this.id = id;
        this.name = name;
        this.price = price;
    }
}
```

按 Copy code 可複製所有程式碼

本例生成 Java 類別名稱 Product，因此請將程式檔名命名為 Product.java。

**程式碼**

檔名：\ex17\src\ProductDemo01\Product.java

```
01 package ProductDemo01;
02
03 public class Product {
04     private int id;
```

```
05     private String name;
06     private double price;
07     public Product(int id, String name, double price) {
08         this.id = id;
09         this.name = name;
10         this.price = price;
11     }
12
13     public int getId() {
14         return id;
15     }
16
17     public String getName() {
18         return name;
19     }
20
21     public double getPrice() {
22         return price;
23     }
24
25     public static void main(String[] args) {
26         //建立 Product 產品陣列 products，並存放五筆 Product 產品記錄
27         Product[] products = new Product[5];
28         products[0] = new Product(1, "Product A", 10.0);
29         products[1] = new Product(2, "Product B", 20.0);
30         products[2] = new Product(3, "Product C", 30.0);
31         products[3] = new Product(4, "Product D", 40.0);
32         products[4] = new Product(5, "Product E", 50.0);
33         //顯示 products 內所有產品記錄
34         for (Product p : products) {
35             System.out.println(p.getId() + ", " + p.getName() + ", " +
36                 p.getPrice());
37         }
38     }
```



說明

1. 第 03~23 行：ChatGPT 生成 Product 產品類別，此類別有 id(編號)、name(品名)、price(單價)三個欄位(屬性)，以及一個建構式用於初始化這三個欄位。同時提供 getId()、getName()與 getPrice()用來取得產品 id、name、price。
2. 第 25~37 行：在 main() 方法中建立包含 5 筆產品記錄的 products 物件陣列，並使用 getId()、getName()、getPrice() 方法取得產品編號、品名、單價，最後透使用 System.out.println() 方法輸出所有產品記錄內容。



實作

延續上例，改使用視窗呈現產品記錄，同時提供上一筆與下一筆按鈕巡覽所有產品記錄。



結果

Product Viewer	
編號:	1
品名:	Product A
單價:	10.0
[上一筆] [下一筆]	

Product Viewer	
編號:	3
品名:	Product C
單價:	30.0
[上一筆] [下一筆]	

本例繼續沿用 Product.java 產品類別但不會使用到此類別的 main() 方法，因此請刪除 main() 方法，僅保留 Product 產品類別的 id、name、price 欄位，以及 getId()、getName()、getPrice() 方法。



程式碼

檔名：\ex17\src\ProductDemo02\Product.java

```
01 package ProductDemo02;
02
03 public class Product {
04     private int id;
05     private String name;
06     private double price;
07     public Product(int id, String name, double price) {
08         this.id = id;
09         this.name = name;
```

```
10         this.price = price;
11     }
12
13     public int getId() {
14         return id;
15     }
16
17     public String getName() {
18         return name;
19     }
20
21     public double getPrice() {
22         return price;
23     }
24
25     public static void main(String[] args) {
26         //建立 Product 產品陣列 products，並存放五筆 Product 產品記錄
27         Product[] products = new Product[5];
28         products[0] = new Product(1, "Product A", 10.0);
29         products[1] = new Product(2, "Product B", 20.0);
30         products[2] = new Product(3, "Product C", 30.0);
31         products[3] = new Product(4, "Product D", 40.0);
32         products[4] = new Product(5, "Product E", 50.0);
33         //顯示 products 內所有產品記錄
34         for (Product p : products) {
35             System.out.println(p.getId() + ", " + p.getName() + ", " +
36                                 p.getPrice());
37         }
38     }
```

向 ChatGPT 詢問「上面 Java 程式請使用視窗呈現，視窗中有編號、品名、單價文字欄，可使用上一筆與下一筆按鈕進行瀏覽每筆產品記錄。」，結果 ChatGPT 生成下圖 Java 程式碼，請依需求複製對應程式碼。(注意 ChatGPT 所產生的程式碼可能不同)



以下是一個使用Java GUI (圖形使用者介面) 的程式，該程式可以顯示多筆產品記錄，並提供上一筆和下一筆按鈕，讓使用者可以瀏覽這些記錄：

```
java Copy code

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ProductGUI extends JFrame implements ActionListener {
    private JTextField idField;
    private JTextField nameField;
    private JTextField priceField;
    private JButton prevButton;
    private JButton nextButton;
    private int currentProductIndex = 0;
```

本例生成 ProductGUI 類別 (Jframe 視窗) Java 程式碼，因此請將程式檔名命名為 ProductGUI.java。ProductGUI 類別繼承自 JFrame 類別，表示此類別是一個視窗。在 ProductGUI 類別中建立了一個 Product 產品陣列包含了 5 筆 Product 產品記錄。在 ProductGUI 建構式中建立了代表編號、品名、單價 JTextField 物件，以及上一筆與下一筆按鈕。ProductGUI.java 程式碼如下：



程式碼

檔名：\ex17\src\ProductDemo02\ProductGUI.java

```
01 package ProductDemo02;
02
03 import javax.swing.*;
04 import java.awt.*;
05 import java.awt.event.ActionEvent;
06 import java.awt.event.ActionListener;
07
08 public class ProductGUI extends JFrame implements ActionListener {
09     private JTextField idField;
10     private JTextField nameField;
11     private JTextField priceField;
12     private JButton prevButton;
13     private JButton nextButton;
```