第一章　簡介

1. 越高階的程式語言就代表語法越接近人類的語言、思維；越低階的程式語言就越接近電腦實際運作的機械碼。高階程式語言可讀性越高，程式較容易除錯與維護；低階程式語言可讀性低、不易除錯與維護，但相對的執行效率高。

2. 電腦無法直接執行高階語言，必須透過解譯器（Interpreter）或編譯器（Compiler）將程式編譯成低階語言（如機械碼），電腦才能執行。

3. 程式碼是以類別為基礎；物件是類別建構的實體，也就是程式實際運作的實際個體。

4. 屬性是物件所擁有的資料描述或項目，也就是物件內部所擁有的變數、常數或物件；方法是物件的行為，也就是物件內部所擁有的函數。

5. Java 程式是先將原始碼編譯成位元碼（Byte code），執行時由Java虛擬機器將位元碼轉換成該作業系統可以執行的程式碼。由於位元碼可以在不同作業系統或平台環境的Java虛擬機器上執行，而不需要經過重新編譯，所以 Java 程式語言可以跨平台，也就是能夠在不同的作業系統上執行。

6. JVM：Java 虛擬機器（Virtual Machine），以軟體模擬的執行環境，用來載入位元碼（Byte Code）並轉換成該作業系統可以執行的程式碼，使得 Java 程式可以跨平台執行。

（以下尚有）

第二章　JDK安裝與測試

1. Java程式存檔的副檔名為 .java。編譯成功會產生一個與程式檔名相同，但副檔名為 .class 的位元碼檔案。

2. Java 使用 javac 指令執行編譯；使用 java 指令執行。

3. public static void main（String args[]）。

4. 類別 物件名稱 = new 類別建構子。

5. 類別圖（Class Diagram）結構分為三個區塊，第一格：類別名稱；第二格：表列該類別的屬性；第三格：表列該類別的方法。

6. Java Development Kit。

（以下尚有）

第三章　基本語法

1. 變數與常數均是由資料型態宣告，且只能存放一個值。變數與常數的差別在於變數的內容可以改變，而常數的內容則為固定不變。

2. byte、short、int、long。主要是差別在可儲存數值大小的差異。

|  |  |  |
| --- | --- | --- |
| 資料型態 | 可儲存之數值範圍 | 佔用空間 |
| byte | -128~127 | 1個位元組 |
| short | -32768~32767 | 2個位元組 |
| int | -2147483648~  2147483647 | 4個位元組 |
| long | -9223372036854775808  ~9223372036854775807 | 8個位元組 |

3. Java 指定浮點數值，預設為double型態。因此如果便因此如果變數的資料型態為float時，並須在該數值後方指定一「f」字母，表示該數值為float型態。

4. a+=b++ 運算式等於 a=a+b++。運算式包含三個執行的步驟，因為變數 b 後置 ++，因此其執行優先權最低：

(1) a+b

(2) a 的值與 b 的值相加後指定給 a

(3) b++

因此結果 a 的值為 15，b 的值為 6。

5. a+=++b 運算式等於 a=a+ ++b。運算式包含三個執行的步驟，因為變數 b 前置 ++，因此其執行優先權最高：

(1) b++

(2) a+b

(3) a 的值與 b 的值相加後指定給 a

因此結果 a 的值為 16，b 的值為 6。

6. k=a++ + ++b。運算式包含三個步驟，其中變數 a 後置 ++，其優先權最低；變數 b 前置++，其優先權最低：

(a) ++b

(b) a+b 指定給k

(c) a++

因此結果 a 的值為13，b 的值為6，k 的值為18。

接續運算 k-=a/b 表示 k=k-a/b 依據先乘除後加減的數學運算優先權，先運算 a/b。基於Java 「整數與整數運算，其結果為整數」的運算規則，a/b 也就是 13/6 的值是 2 而非 2.1666…。因此運算結果 k 的值為16。

（以下尚有）

7.

|  |
| --- |
| class ch03wk04{  public static void main(String args[]){  int i=10,k=20;  float j=10.5f;    System.out.println("第一小題運算式答案: "+(2\*2+3\*2) );  System.out.println("第二小題運算式答案: "+(2\*i+j+3\*k) );  System.out.println("第三小題運算式答案: "+((i-3)\*(j+2)) );  System.out.println("第四小題運算式答案: "+((5\*i)/2.15) );  }  } |

（以下尚有）

第四章　流程控制

1.

|  |
| --- |
| import java.util.\*;  class ch04wk01{  public static void main(String args[]){  Scanner sc = new Scanner(System.in);  System.out.print("請輸入一個整數:");  int value=sc.nextInt();  if (value%2==0||value%3==0||value%6==0)  System.out.println("輸入的數是2、3或6的倍數");  else  System.out.println("輸入的數不是2、3或6的倍數");  }  } |

2. 執行結果：

|  |
| --- |
| i=17 j=8 k=7value of i is:odd |

3.

|  |
| --- |
| import java.util.\*;  class ch04wk03{  public static void main(String args[]){  Scanner yr = new Scanner(System.in);  System.out.print("請輸入年度:");  int year=yr.nextInt();  if (year%4 == 0){  System.out.println("依據Julian算法，該年是閏年");  if (year%100==0 && year%400 !=0)  System.out.println("但依據Gregorian算法，不是閏年");  }else  System.out.println("不是潤年");  }  } |

4.

|  |
| --- |
| import java.util.\*;  class ch04wk04{  public static void main(String args[]){  Date dt = new Date();  Scanner sc = new Scanner(System.in);  System.out.print("請輸入身高:");  int height\_cm=sc.nextInt();  System.out.print("請輸入體重:");  float weight=sc.nextFloat();  float height\_m=(float)height\_cm/100;  float bmi=weight/(height\_m\*height\_m);  System.out.println("測量時間："+dt.toString()+", 你的BMI值是:"+bmi);  if (bmi>=24){  System.out.println("過重");  }else  System.out.println("正常");  }  } |

5. 質數（prime number）是除了 1 之外，只能被1和自己整除的整數。因為一個數的最小倍數是 2，因此只要檢驗是否能整除一個整數時，超過該整數的 1/2 就不可能發生整除該數的情況，因此檢驗的迴圈只需處理由 2～該整數的 1/2 即可。

|  |
| --- |
| class ch04wk05 {  public static void main(String args[]){  boolean prime; //若是true表示該數值是質數  for (int value=3; value<=1000; value++){  prime=true; //執行下列迴圈，若執行完畢後prime為false表示此數不是為質數  for (int i=2; i<=value/2; i++){  if((value%i)==0)  prime=false;  if (prime==false) break; //已確定此數不是質數，不需再執行迴圈驗證  }  if (prime)  System.out.println("整數:"+value+"是質數");  }  }  } |

第五章　陣列與字串

1.

|  |
| --- |
| class ch05wk01{  public static void main(String[] args){  //宣告一個二維陣列，第一維代表學生人數，第二維代表修課科目數  int[][] score = {{1,85,83},{2,92,75},{3,90,95},{4,77,92},{5,83,65}};  System.out.println("二維陣列的內容為");  for (int i=0; i<score.length; i++){  for (int j=0; j<score[i].length; j++)  System.out.print("\t"+score[i][j]);  System.out.println();  }  }  } |

2.

|  |
| --- |
| import java.util.\*;  class ch05wk02{  public static void main(String[] args){  if (args.length>=0){  int[] A1= new int[args.length]; //宣告A1整數陣列  int sum=0;  for (int i=0; i<args.length; i++){  A1[i] = Integer.parseInt(args[i]);  sum+=A1[i];  }  System.out.println("內容總計為:"+sum);  int[] A2 = new int[A1.length]; //宣告A2整數陣列  A2=A1.clone();  if (Arrays.equals(A1,A2))  System.out.println("陣列A1與陣列A2內容相同");  else  System.out.println("陣列A1與陣列A2內容不同");    Arrays.sort(A1); //將陣列A1內容排序  System.out.println("最小值:"+A1[0]+", 最大值:"+A1[A1.length-1]);    }else System.out.println("命令提示字元未輸入資料");  }  } |

3.

|  |
| --- |
| import java.util.\*;  public class Prog03{  public static void main(String args[]){  //args -- 答案  int itemNum=args.length;  if (itemNum>0) {  System.out.println("共有"+itemNum+"題，請逐一輸入答案：");  String answer="";  int num=0;  try{  String Ans[]= new String[itemNum];  Scanner sc = new Scanner(System.in);  for (int i=0; i<itemNum; i++){  System.out.print("請輸入 No."+(i+1)+"的答案:");  answer=sc.next().toUpperCase();  Ans[i]=answer;  if (args[i].equals(answer)){  System.out.println("正確!!");  num++;  }else  System.out.println("錯誤!!");  }  System.out.println("答對題數："+num );  }catch (Exception e){System.out.println("System error!!"); }  }  }  } |

4.

|  |
| --- |
| class ch05wk04{  public static void main(String[] args){  //建構StringBuffer物件  StringBuffer strBuff = new StringBuffer("This is a book.");  System.out.println("目前strBuff的內容為：" + strBuff);  //利用setCharAt設定字串緩衝區的內容  strBuff.setCharAt(2, 'a');  strBuff.setCharAt(3, 't');  System.out.println("使用setCharAt後，strBuff的內容為：" + strBuff);    //使用append新增字串緩衝區的內容  strBuff.append(" That is a pen.");  System.out.println("使用append後，strBuff的內容為：" + strBuff);    //使用insert新增字串到緩衝區中的特定位置  strBuff.insert(0, "I wish ");  System.out.println("使用insert後，strBuff的內容為：" + strBuff);    //使用delete刪除字串緩衝區的內容  strBuff.delete(0, 23);  System.out.println("使用delete後，strBuff的內容為：" + strBuff);    //使用replace取代字串緩衝區的內容  strBuff.replace(0, 4, "This");  System.out.println("使用replace後，strBuff的內容為：" + strBuff);    //使用reverse將字串緩衝區的內容反轉  strBuff.reverse();  System.out.println("使用reverse後，strBuff的內容為：" + strBuff);  }  } |

（以下尚有）

5.

|  |
| --- |
| class Convert  public static void main(String args[]){  //將args陣列內容組成一字串  String text="";  for(int i=0; i<args.length;i++)  text+=args[i]+" ";  text=text.trim(); //去除前後空白字元    int pos=-1,cnt=0; //pos:紀錄「台」的位置，cnt:紀錄「台」的字數  while ((pos=text.indexOf("台",pos+1))>=0){  cnt++;  }  text=text.replace("台","臺"); //將字串內容「台」取代為「臺」  System.out.println("發現字數:"+cnt+"\t取代結果:"+text);  }  } |

（以下尚有）

第六章　例外的處理

1.

|  |
| --- |
| public class ch06wk01{  public static void main(String args[]){  try{  int num\_1 = Integer.parseInt(args[0]);  int num\_2 = Integer.parseInt(args[1]);  float quo = (float)num\_1/num\_2;  System.out.print("商數為:"+quo);  }catch(NumberFormatException e){  System.out.println("輸入的資料不是整數");  }catch(ArrayIndexOutOfBoundsException e){  System.out.println("輸入的數量不足，請輸入兩個整數");  }  }  } |

2.

|  |
| --- |
| import java.util.\*;  class Guess{  public static void main(String args[]){  Scanner sc = new Scanner(System.in);  int answer=(int)(Math.random()\*100);  boolean bBingo = false; //*true表示猜中*  int cnt=0, guess;  while (cnt<7 && !bBingo){  System.out.print("請輸入要猜的數:");  try{  guess=sc.nextInt();  if (guess<0 || guess>=100)  throw new Exception("必須介於0~100之間");  if (answer==guess)  bBingo=true;  else if (guess<answer)  System.out.println("太小");  else  System.out.println("太大");  cnt++;  }catch(InputMismatchException e){  System.out.println("必須是整數");  sc.next();  }catch(Exception e){  System.out.println( e.getMessage() );  }  }  if (bBingo)  System.out.println("恭喜! 猜中了!!");  else  System.out.println("猜太久了，答案是:"+answer);  }  } |

3.

|  |
| --- |
| public class ch06wk02{  public static void main(String[] args) {  System.out.println(divide(4, 2));  if (args.length > 1) {  try{  int num01 = Integer.parseInt(args[0]);  int num02= Integer.parseInt(args[1]);  System.out.println("輸入的被除數為："+num01+", 除數為:"+num02);  System.out.println("計算結果為:"+divide(num01, num02));  }catch (Exception e){  System.out.println("發生例外，例外訊息："+e.getMessage());  }  }  }  public static float divide(int a, int b) {  if (b == 0) {  ArithmeticException myException = new ArithmeticException("分母不能為零!!");  throw myException;  } else  return (float)a / b;  }    } |

（以下尚有）

第七章　常用類別

1. 套件（package）也就是類別庫，JDK提供的許多套件，例如：java.awt、ava.io、java.lang、java.sql……等等。套件分門別類地存放了各類別的 ByteCode，使用 import 引入套件時，編譯器便可以將指定的類別程式碼加入編譯好的 .class 檔案中。

但是為了簡化程式，Java 將經常會使用的類別，例如：String、Math、和System……等類別集中放置在 java.lang 套件內，使用時已預設自動 import，所以無需自行指定。所以，我們可以直接使用這一類的類別，而無需在程式開始時使用 import 引入 java.lang 套件。

2.

|  |
| --- |
| import java.util.\*;  import java.text.\*;  public class SetBirth{  public static void main(String args[]){  Scanner sc = new Scanner(System.in);  int nYear,nMonth,nDay;  System.out.print("請輸入生日資訊：");  System.out.print("民國年?");  nYear=sc.nextInt();  System.out.print("月?");  nMonth=sc.nextInt();  System.out.print("日?");  nDay=sc.nextInt();  Calendar cBirth = Calendar.getInstance();  SimpleDateFormat sdFormat = new SimpleDateFormat("yyyy-MM-dd");  cBirth.set(nYear+1911, nMonth-1, nDay);  System.out.println( sdFormat.format(cBirth.getTime() ));  }  } |

（以下尚有）

3.

|  |
| --- |
| public class GetProbability{  public static void main(String args[]){  int nNum;  *//陣列[0]紀錄1出現次數，[1]紀錄2出現次數，餘此類推*  int nProb[]= new int[6];  for (int i=1;i<=100;i++){  nNum =(int)(Math.random()\*6)+1; *//亂數值:0~5*  nProb[nNum-1]++;  }  *//顯示*  for(int i=0; i<6; i++){  System.out.print( (i+1)+"出現次數："+nProb[i]+"\t");  for (int j=0; j<nProb[i];j++)  System.out.print("\*");  System.out.println(); //換行  }  }  } |

4.

|  |
| --- |
| import java.util.\*;  public class VectorSort{  public static void main(String args[]){  Vector<Integer> vector = new Vector<Integer>();  int nNum;  for(int i = 0; i < 20; i++) {  nNum=(int)(Math.random()\*(51-10))+10;  vector.add( nNum );  }  System.out.println("原始資料：\t"+vector);  Collections.sort(vector,Collections.reverseOrder());  System.out.println("反向排序：\t"+vector);  Collections.sort(vector);  System.out.println("正向排序：\t"+vector);  }  } |

第八章　類別與物件

1.

|  |
| --- |
| public class ch08wk01{  public static void main(String args[]){  Book bk = new Book(101);  bk.setTitle("撒哈拉歲月","三毛");  bk.setPublish("台北市","皇冠","民99");  System.out.println(bk.getInfo());  }  }  class Book{  private String title, author, place, publisher, year;  private int brn;  Book(int no){  brn=no;  }  public void setTitle(String title, String author){  this.title=title;  this.author=author;  }  public void setPublish(String place, String publisher, String year){  this.place=place;  this.publisher=publisher;  this.year=year;  }  public void setBrn(int no){  brn=no;  }  public String getInfo(){  return "書號:"+brn+"\n"+title+"/"+author+".-- "+place+": "+publisher+", "+year;  }  } |

2.

|  |
| --- |
| 陣列內容:10 20 30 40 50 60  40在陣列的位址：4 |

3.

|  |
| --- |
| 住家房子樓高1層  b1的住家總共有1層樓高  10樓公寓房子樓高10層  b2的10樓公寓總共有10層樓高 |

（以下尚有）

第九章　繼承

1. 多載（Overloading）：在同一類別中，宣告多個方法，這些方法名稱相同，但參數個數不同或參數的資料型態不同。

覆寫（Overriding）：子類別繼承父類別，改寫父類別的方法。覆寫的方法名稱、參數和型態都必須和父類別的方法相同。

2.

|  |
| --- |
| b物件的x屬性內容=20  c物件的x屬性內容=30 |

（以下尚有）

第十章　多型

1.

● 抽象類別是指類別內的方法或屬性宣告為抽象，由繼承的類別實作其程式碼；而介面則是宣告整個類別是一個介面。因此，抽象類別成員可以只有部分是抽象，但介面的成員則全部是抽象。

● 抽象類別裡可以宣告屬性，但介面無法宣告屬性。

● 抽象類別可以宣告有建構子，但介面沒有建構子。

● 抽象類別的屬性、方法可以宣告為private、protected或public，但介面只能宣告為puiblic。

● 抽象類別被繼承時可以不實作，但介面被繼承時一定要實作才能使用。

● 抽象類別（包括一般類別）只能單一繼承，但介面可以多重繼承，也就說一個類別可以implemt很多個介面，但只能extends一個父類別。

2.

|  |
| --- |
| abstract class GraphicObj {  int x, y;  abstract String getInfo();  abstract float getArea();  }  class Circle extends GraphicObj{  final float pi=3.1416f;  Circle(int x, int y){  this.x=x;  this.y=y;  }  public String getInfo(){  return "這是一個圓形";  }  public float getArea(){  //橢圓面積=2pi\*y+4(x-y)  if (x<y){ //確定x為長軸，y為短軸  int temp;  temp=x;  x=y;  y=temp;  }  return 2\*pi\*y+4\*(x-y);  }  }  class Rectangle extends GraphicObj{  Rectangle(int x, int y){  this.x=x;  this.y=y;  }  public String getInfo(){  return "這是一個方形";  }  public float getArea(){  //方形面積=x\*y  return (float)x\*y;  }  }  public class ch10wk02{  public static void main(String args[]){  Circle c = new Circle(5,10);  System.out.println(c.getInfo()+"\n面積="+c.getArea());  Rectangle r = new Rectangle(5,10);  System.out.println(r.getInfo()+"\n面積="+r.getArea());  }  } |

（以下尚有）

3.

|  |
| --- |
| interface Sports{  public void setHomeTeam();  public void setVisitingTeam();  }  interface Football extends Sports{  public void homeTeamScored();  public void visitingTeamScored();  public void endOfQuarter();  }  interface Hockey extends Sports, Event{  } |

（以下尚有）

**第十一章**　**網站開發簡介**

1. Web是由歐洲粒子物理研究中心（European Particle Physics Laboratory，法文縮寫CERN）服務的英國物理學家Tim Berners-Lee發明的。

三項關鍵協定為：

(1) 超文傳輸協定（Hypertext Transfer Protocol，HTTP）。

(2) 統一資源定位器 （Universal Resource Locator，URL）。

(3) 超文標示語言（Hypertext Markup Language，HTML）。

2. 連線（Connection）、提出請求（Request）、回應結果（Response）、切斷聯線（Disconnection）。

3. Web 是全球資訊網（World Wild Web）的簡稱，是 internet 上的一個服務。而網際網路（internet）是網路與網路之間所串連成的龐大網路，且使用 TCP/IP 為主要網路封包傳輸與定址協定。簡單的說 internet 是網網相連的巨大網路環境，Web 是在此環境上運行的服務。

4. Java Script 是由網景（Netscape）公司所發展的網頁程式語言，用於內嵌在 HTML 文件內，下載至使用者端，由瀏覽器直譯（Interpret）後執行；Java Server Pages（JSP）則是由 Sun公司（現已由Oracle公司併購）開發，執行於 Web Server 上的 Java 程式語法。

5. 80。

（以下尚有）

第十二章　JSP 元素分類

1.

(1) 指引元素（Directives Element）。

(2) 手稿元素（Scripting Element）。

(3) 動作元素（Action Element）。

2. <%@ page contentType="text/html;charset=utf-8" %>

3. 指引元素的宣告是用於告知 Web Server 的處理標示；而<meta>元素的宣告則是告知瀏覽器的處理標示。也就是說：

<%@ page contentType="text/html;charset=big5" %> 是告訴Web Server 以big5處理文件的字碼。

<meta http-equiv="Content-Type" content="text/html; charset=big5"> 則是告訴樓覽器以big5處理文件的字碼。

4. errorPage 屬性用於指定該 JSP 程式負責處理例外的程式檔案名稱；該負責處理例外的程式必須在 isError 屬性指定為 true，Web Server 才會將例外狀況交與執行。

5. 套件的檔案（\*jar）應置於 resin 或 Tomcat 伺服器的 WEB-INF 目錄內的 lib 子目錄內。若是 Java Bean 則置於 WEB-INF 目錄內的 classes 子目錄內。

（以下尚有）

第十三章　HTML 簡介與 JSP 跨網頁資料傳遞

1. 標籤是以小於符號（<）、大於符號（>），以及其內的名稱組成。標籤分為起始標籤（Start tag）、結束標籤（End tag）。

起始標籤、結束標籤加上其內的內容，稱為元素（Element）。

2. <FORM>表單元素的 method 屬性值：get 表示不允許在 message-body 中傳遞資料的，也就是傳遞參數的資訊不能封裝在 message-body 中，所以從瀏覽器的網址列就可以看見表單要傳送的資料。而 method 的屬性值指定為 post 時，表示將表單傳遞參數的資訊放在 message-body 進行傳送，因此在瀏覽器的網址列就不會看見表單要傳送的資料。

3.

(1) name。

(2) value。

4. getParameterValue()。

5. 無論網頁視覺化表單元素輸入的內容是字串、日期、數字……，request物件接收到的資料型態均是字串。

（以下尚有）

第十四章　JSP 隱含物件

1. out 物件的 print( ) 方法僅是將指定的資料輸出至瀏覽器，println( ) 方法則是輸出資料後會加上換行（Linefeed）和游標歸位（Carriage Return）的控制碼。

基於「網頁顯示效果由 HTML 標籤決定」的原則，輸出資料使用 print( ) 或 println( ) 並不會改變網頁顯示的結果。也就是說資料於瀏覽器上是否換行，由標籤決定，而非由資料是否有換行決定。

2. session 是每一個連線使用者各別配置一個session 物件，當使用者(瀏覽器)離線時，Web Server 便刪除該 session 物件，因此儲存在其內的資料會一直保存，直到使用者（瀏覽器）離線為止。

而 application 則是整個 Web Server 僅宣告一個物件，因此是所有連線使用者共用這一個 application 物件。儲存在其內的資料會一直保存，直到網站關閉（shutdown）為止。

3. 存入的資料必須是物件，不可以是資料形態所宣告的變數。因為存入後會被轉換成 Object 類別的物件，因此取出時必須強制型別轉換，還原回原來型態的物件。

4. request 物件的 getParameter( ) 方法是用來取得由瀏覽器將網頁表單元素所輸入 request 物件的資料；getAttribute( ) 方法則是取出由 setAttribute( ) 方法所存入 request 物件的資料。

5. Excepton 是一個例外類別；而 exception 則是 Web Server 所建構的一個例外物件，該物件記錄最後發生的例外狀況。

（以下尚有）

第十五章　檔案存取

1. Cookie。

2. Cookie 儲存於使用者的電腦內，儲存的檔名包含網站 URL 及登入電腦作業系統的使用者名稱。為考量安全性，儲存的目錄位置由瀏覽器決定，無法由伺服器端的 JSP 程式決定儲存的檔案名稱。

3. 使用者可以透過瀏覽器提供的工具清除 Cookie 的資料。若是由 Web Server 端執行刪除，只能刪除該網站寫入的 Cookie 資料，刪除的方式指定該 Cookie 物件的生命周期設定為0。

4.

(1) FileReader 類別建構的檔案物件，只能一個自元一個字元逐一的讀取檔案的內容，如果需要一次讀取一行內容，必須使用 BufferedReader 類別建立的物件。

(2) FileReader 物件是指向實際目錄的檔案，因此建構子傳入的參數是目錄+檔案名稱；而 BufferedReader 則是指向 FileReader 物件，因此建構子傳入的參數是 FileReader 物件。

（以下尚有）

第十六章　網站資料庫應用程式開發-初級篇

1. 為了讓電腦可以彼此相溝通，必須定義一套雙方都了解的處理方式或對談語言，這種方式或語言便稱為協定。例如全球資訊網（Web）至少須具備 TCP/IP 網路封包的傳輸協定、HTTP 超文傳輸的協定、HTML 超文標示語言。

2. java.sql。

3. WEB-INF 目錄的 lib 目錄內。

4.

(1) Statement：用於執行靜態的SQL敘述。

(2) PreparedStatement：用於執行動態的SQL敘述。

(3) CallableStatement：用於執行預儲程序。

5.

(1) executeQuery（*SQL敘述*）

執行有回應資料集的 SQL 敘述，例如select。傳回值為執行結果傳回的資料集（DataSet）物件。

(2) executeUpdate（*SQL敘述*）

執行無回應資料集的 SQL 敘述，例如 insert、update、delete。回傳值若為 1 表示執行成功，若為 0 表示沒有資料新增/更改或刪除。

(3) execute（*SQL敘述*）

可以執行任何 SQL 敘述，傳回值為布林（boolean），true 表示執行成功，false 表示執行不成功。

（以下尚有）

第十七章　網站資料庫應用程式開發-進階篇

1. 因為程式存取資料需要經常連接資料庫，導致浪費大量系統資源。透過事先建立多個資料庫連結的物件。當某一程式需要存取資料庫時，便拿取某一個連結物件，並將這個連結標記為「使用中」；用完後再歸還，並清除「使用中」的標記。如此，這些連結物件就如同在一個池中，提供各程式共享，因此稱為連線池。

2. <jsp:useBean id="pool" scope="application" class="myBean.PoolBean"/>

3.

ResultSet.TYPE\_SCROLL\_INSENSITIVE：資料集的指標可雙向移動

ResultSet.CONCUR\_READ\_ONLY：資料集的內容不可更動

完整的程式碼可撰寫成：

Statement st = con.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);