

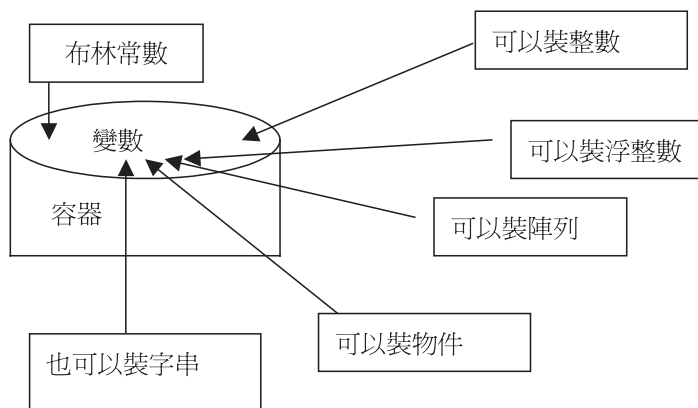


Chapter [2]

原始資料型態
和操作

2-1 簡單的程式

Java 支援的資料型態有浮整數 (floating point)、字元 (char)、整數 (integer)、物件 (object)、布林常數 (Boolean)、空值 (null) 和字串 (string)。記憶體就像是容器可以裝很多東西一樣。在 Java 程式語言中，每一個變數和每一個運算式在編譯時都有一種型態。這個資料包含了原始資料型態和參考型態。參考型態包含了類別型態和介面型態。



範例

Circle.java

第一行我們使用import來輸入javax.swing.JOptionPane套件。

第二行到第十三行我們則宣告了circle類別。

第三行為circle類別的主要方法main()。

第四行宣告了radius半徑的資料型態為雙精度浮點數。

第五行宣告了area面積的資料型態為雙精度浮點數。

第六行宣告了pi的資料型態為雙精度浮點數，並且給予初始值3.14159。

第八行會得到圓的面積。

第九行會顯示圓的面積。

```
1 import javax.swing.JOptionPane;
2 public class Circle{
3     public static void main(String[] args){
4         double radius;
5         double area;
6         double pi=3.14159;
7         radius=10;
8         area=radius*radius*pi;
9         JOptionPane.showMessageDialog(null,"圓的面積是"+area,
10                                     "圓的面積",JOptionPane.INFORMATION_MESSAGE);
11     }
12 }
13 }
```

圓的面積是314.159。

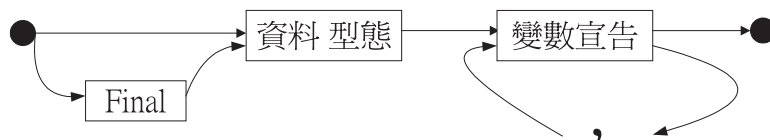


2-2 識別名稱

每一個變數都有識別名稱，我們變數宣告時，就是給該變數一個識別名稱。在 circle.java 範例中，radius 半徑為識別名稱，area 面積也是識別名稱。所有的識別名稱是由字元、數字、下底線 (_) 和錢符號 (\$) 所組成。識別名稱的命名一定要以字元或下底線 (_) 開始，而不能以數字開始。識別名稱也不可以是 Java 的保留字。識別名稱也不可以是布林值 (true, false 或 null)。識別名稱沒有長度的限制。Area、radius、good 都是合法的識別名稱。而 356a 則不是識別名稱，因為它以數字開頭，Java 編譯器會偵測出錯誤。Java 是有大小寫的區隔，a 和 A 是不一樣的變數。

2-3 變數

這是區域變數宣告。先宣告變數的資料類型，再宣告變數。我們可以將變數宣告寫在同一行，並且用逗號(,)分開。



在 circle.java 範例中，radius 半徑為變數，並且它的資料型態為 double(雙精度浮點數)；area 面積為變數，並且它的資料型態為 double(雙精度浮點數)。先有資料型態，接下來才是接變數。

```

1 double radius;
2 double area;
3 double pi=3.14159;
  
```

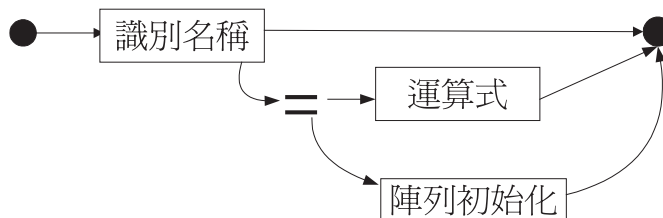
我們可以將 circle.java 範例中的 radius 和 area 變數宣告寫在同一行，並且用逗號(,)分開。

```

1 double radius, area;
2 double pi=3.14159;
  
```

2-3-1 宣告變數

這是變數宣告，我們也可以使用分配符號(=)分配給它初始值。



在 circle.java 範例中，radius 半徑為變數，並且它的資料型態為 double(雙精度浮點數)；area 面積為變數，並且它的資料型態為 double(雙精度浮點數)。先有資料型態，接下來才是接變數。我們宣告 pi 變數為雙精度浮點數，並且給它初始值 3.14159。

```
1 double radius;  
2 double area;  
3 double pi=3.14159;
```

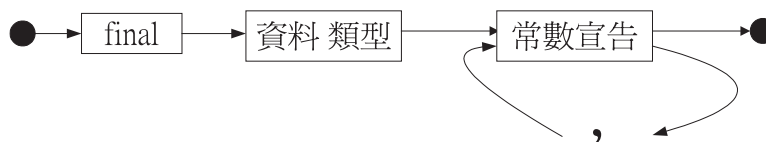
2-4 分配敘述與分配運算式

這是分配敘述。分號；代表運算式結束。



2-5 常數

變數的值是會改變的，但常數的值是不會改變的。我們在資料類型前加上 final 關鍵字，這樣資料的值就無法改變，而成為常數。





範例 Circle_constant.java

我們第六行加上關鍵字final，這樣pi就會被當作常數。pi加上final關鍵字後，它的值就不能被改變。

```
1 import javax.swing.JOptionPane;
2 public class Circle_constant{
3     public static void main(String[] args){
4         double radius;
5         double area;
6         final double pi=3.14159;
7         radius=10;
8         area=radius*radius*pi;
9         JOptionPane.showMessageDialog(null,"圓的面積是"+area,
10                                     "圓的面積",JOptionPane.INFORMATION_MESSAGE);
11     }
12 }
```

這是執行的情況。

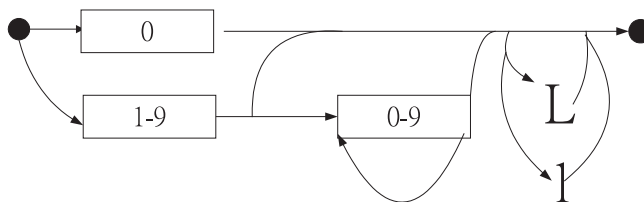


2-6 數值資料型態與運算子

變數 `int a=5, c`，`c=a+5`，變數 `c` 的值為 10 在這個中，`a=5` 是一個運算式，`c=a+5` 是一個運算式，`+` 和 `=` 是運算子，變數 `a` 和數值 5 是運算元。我們可以了解運算式就是由運算子和運算元所組成。

`byte` 為位元組資料型態，`int` 是整數資料型態，`float` 是浮點數資料型態，`double` 是雙精度浮點數資料型態。位元大小是指該資料型態在記憶體所佔的大小。最小數值和最大數值是該資料型態的範圍大小。`E+38` 是 10 的 38 次方。

資料型態	位元大小	最小數值	最大數值
byte	8位元	-128	127
short	16位元	-32768	32767
int	32位元	-2147483648	2147483647
long	64位元	-9223372036854775808	-9223372036854775807
float	32位元	-3.4E+38	3.4E+38
double	64位元	-1.7E+308	1.7E+308



範例 Datatype.java

我們在第7行將pi2的資料型態設為浮點數，因此將3.14159後面加上浮點數的符號F。因為Java將每一個有小數點的數當作雙精度浮點數，因此我們在該數後面加上F或f來確定這數是浮點數。

```
1 import javax.swing.JOptionPane;
2 public class Datatype{
3     public static void main(String[] args){
4         int grade=55;
5         byte smallnumber=127, smallnumber2=12;
6         long stars=88888888888888L;
7         float pi2=3.14159F;
8         double radius=10;
9         double area=radius*radius*pi2;
10        JOptionPane.showMessageDialog(null, "圓的面積是"+area,
11        "圓的面積", JOptionPane.INFORMATION_MESSAGE);
12    }
13 }
```

這是數值。



2-6-1 運算式

我們的程式主要是由運算子與運算元所組成。我們的想法 轉換成程式的架構，再由運算子來架構骨架與運算元的組成，這樣就可以組成大部份的程式邏輯。

算數的加 + 減 - 乘 * 除 ÷ 和餘數 % 稱作為算術運算子。我們常用算術運算子來作運算式，而程式的百分之五十以上都是由算數運算子所組成。

範例 Arithmetic.java

第三行的一加一等於二，再將2指派給變數a，使用加法運算子“+”。

第四行的二減一等於一，再將1指派給變數b，使用減法運算子“-”。

第五行使用除法運算子“/”，再將3指派給變數div。

第八行使用乘法運算子“*”，再將314.159分配給area變數。

第九行使用餘數運算子，5%3(5取3的餘數為2)值為2。

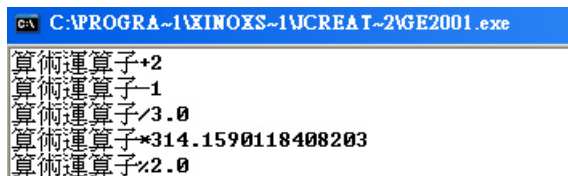
第十行到第十四行我們使用println()函數。

```

1 public class Arithmetic{
2     public static void main(String[] args){
3         int a=1+1;
4         int b=2-1;
5         double div=6/2;
6         float pi2=3.14159F;
7         double radius=10;
8         double area=radius*radius*pi2;
9         double c=5%3;
10        System.out.println("算術運算子+"+a);
11        System.out.println("算術運算子-"+b);
12        System.out.println("算術運算子/"+div);
13        System.out.println("算術運算子*"+area);
14        System.out.println("算術運算子%"+c);
15    }
16 }

```

這是執行的情況。



```

C:\PROGRAMS\1\XINOXS~1\CREAT~2\GE2001.exe
算術運算子+2
算術運算子-1
算術運算子/3.0
算術運算子*314.1590118408203
算術運算子%2.0

```

2-7 數值型態轉換

有時一個數值運算式包含了數種的資料型態，Java 允許在不同的資料型態上作運算，Java 會自動轉換運算元。下面是運算式中，運算元轉換的原則。

假如在數值運算式中，有一個運算元是雙精度浮點數 (double)，則其它的資料型態就會轉成雙精度浮點數；否則，如果有一個運算元是浮點數 (float)，這其它的運算元就會被轉成 float；否則，如果有一個運算元是 long 資料型態，這其它的運算元就會被轉成 long；否則，這兩個運算元就會被轉成整數 int 資料型態。

這些都是 Java 的自動轉型。運算元會轉型成精密度最高的數值資料型態再作運算。

範例

Numeric_type.java

第三行宣告變數i為位元組資料型態，並給予30的數值。

第四行宣告s為long資料型態。

第五行因為t為雙精度浮點數，所以它會將變數i和變數s轉為雙精度浮點數，再作運算。

```
1 public class Numeric_type{
2     public static void main(String[] args){
3         byte i=30;
4         long s=i*5+3;
5         double t=i*5.5+s/3;
6         System.out.println(i);
7         System.out.println(s);
8         System.out.println(t);
9     }
10 }
```

因為t為雙精度浮點數，所以它會將變數i和變數s轉為雙精度浮點數，再作運算，因此得到雙精度浮點數的值216.0，後面有小數點。

```
C:\C:\PROGRAMS\JAVAX\JDK1.4\JRE\bin\java.exe
30
153
216.0
Press any key to continue...
```

範例 Force_Numeric_type.java

第八行我們使用強制轉型(int)，將t由浮點數轉為整數。我們使用(int)來強制轉型成小括號裏的型態。Java會將精度較低的數值型態，自動轉成精度較高的型態。但要由較高的精度轉較低時，則要用強制轉型，前面要加上要強制轉型的型態。

```
1 public class Force_Numeric_type{
2     public static void main(String[] args){
3         byte i=30;
4         long s=i*5+3;
5         double t=i*5.5+s/3;
6         System.out.println(i);
7         System.out.println(s);
8         System.out.println((int)t);
9     }
10 }
```

這是強制轉型成整數，雙精度浮點數的小數點不見了。

```
F:\java2\3\example>javac Force_Numeric_type.java
F:\java2\3\example>java Force_Numeric_type
30
153
216
```

範例 Cast.java

我們可以將資料作強制()型態轉換。

第四行將浮點數f轉成整數資料型態使用(int)，再分配給整數i。

第六行將數值5給整數資料型態s。第六行的浮點數資料型態k被強制轉換成整數。

```
1 public class Cast{
2     public static void main(String[] args){
3         float f = (float) 5.3;
4         int i =(int) f;
5         double k =5.5;
6         int s=(int)k;
7         System.out.println(i);
8         System.out.println(k);
9         System.out.println(s);
10     }
11 }
```

執行的結果，浮點數s沒有被強制轉換，因此顯示5。



A screenshot of a Windows command prompt window. The title bar is blue and contains the text "C:\Program Files\Xinox Software\JCreator\J3LE\GE2001.exe". The command prompt shows three lines of output: "5", "5.5", and "5".

```
C:\Program Files\Xinox Software\JCreator\J3LE\GE2001.exe  
5  
5.5  
5
```

2-8 字元資料型態與運算子


char 資料型態是用於單一字元。我們使用單引號來將它包住。而字串資料型態是使用雙引號將它包住。

範例 Cha_data.java

我們在第三行宣告letter變數的資料型態為字元，並且將A字元給letter變數，我們使用單引號將該資料包住。

```
1 public class Cha_data{
2     public static void main(String[] args){
3         char letter='A';
4         char num='5';
5         System.out.println(letter);
6         System.out.println(num);
7     }
8 }
```

這是執行的情況。



2-8-1 Unicode與ASCII碼

Java 使用 Unicode 來編碼，它是屬於 16-bit 的編碼方式。Unicode 是由兩個位元組來組成一個字，前面為 \u，表示方法是 4 個 16 進位的數，從 '\u0000' 到 '\uFFFF'。

'\u0042' 是大寫的 B。Unicode 可以表示 65536 個字元，因為 FFFF 是 16 進位的 65536。一般的程式軟體是使用 ASCII，七位元的編碼。Unicode 也包含了 ASCII，它是從 '\u0000' 到 '\u007F'。

範例 Unicode_data.java

第三行使用Unicode來代表，'\u0042' 是Unicode，而它是代表大寫的B。大寫B的Unicode，第一個位元組是00，第二個位元組是42。

```

1 public class Unicode_data{
2     public static void main(String[] args){
3         char letter='\u0042';
4         System.out.println(letter);
5     }
6 }

```

這是大寫的 'B'。大寫B的Unicode，第一個位元組是00，第二個位元組是42。

```

C:\Program Files\Xinox Software\JCreator\3LE\GE2001.exe
B

```

2-8-2 跳脫字元

Java 允許我們使用跳脫字元來代表特殊的符號。

說明	跳脫字元	Unicode
後退鍵	\b	\u0008
Tab鍵	\t	\u0009
反斜線	\\	\u005c
單引號	\'	\u0027
雙引號	\"	\u0022

範例 Unicode_escape.java

我們在第4行使用跳脫字元 \" 來顯示雙引號。

```

1 public class Unicode_escape{
2     public static void main(String[] args){
3         System.out.println("小明說:\"Java很好用\"");
4     }
5 }

```

這是執行的情況，出現雙引號。

```

C:\Program Files\Xinox Software\JCreator\3LE\GE2001.exe
小明說:"Java很好用"

```

2-9 布林資料型態和運算子

比較運算子和邏輯運算子，所得到的結果就是一個布林常數 (true 或 false)。

當 $3 < 5$ 時會回傳 true，當 $3 < 2$ 時會回傳 false。比較運算子就是用來比較兩個運算式。

比較運算子	功能	用法
<	小於(less than)	Expr < expr
>	大於(greater than)	Expr > expr
>=	大於或等於	Expr >= expr
==	相等(equal)	Expr == expr
!=	不相等	Expr != expr
<=	小於等於	Expr <= expr
<>	不等於	Expr <> Expr

範例 Less.java

當 $3 < 5$ 時會回傳 true(1)，當 $3 > 5$ 時會回傳 false。

```
1 public class Less{
2     public static void main(String[] args){
3         System.out.println(3<5);
4         System.out.println(3>5);
5     }
6 }
```

$3 < 5$ 為布林值的 true， $3 > 5$ 為布林值的 false。

```
C:\Program Files\Xinox Software\JCreator\3LE\GE2001.exe
true
false
```

2-9-1 邏輯運算子

邏輯運算子可以結合條件，以一個表達式判斷許多條件，而這些條件的結果不是真 true 就是假 false。

`&&` 稱為“與邏輯運算子”，只有當所有條件都成立時才會回傳真 `true`，否則回傳假。

`||` 稱為或邏輯運算子，只要運算式中一個條件成立就會回傳真 `true`，只有當所有的條件都為假 `false` 時，才會回傳假 `false`。

`!` 為相反邏輯運算子，真 `true` 的條件加上 `!` 相反邏輯運算子時，就會變成假 `false`；當假 `false` 的條件加上 `!` 相反邏輯運算子時，就會變成真 `true`。

`Xor` 為互斥運算子，當只有條件都不相同（互斥）時才會回傳真 `true`，其它都回傳 `false`。當（條件 `A(true)`）`Xor`（互斥）（條件 `B(false)`）傳回真，或當（條件 `A(false)`）`Xor`（互斥）（條件 `B(true)`）傳回真，其它則傳回 `false`。

邏輯運算子	功能	用法
<code>&&</code>	邏輯運算and(與)	<code>Expr && expr</code>
<code> </code>	邏輯運算or(或)	<code>Expr expr</code>
<code>!</code>	邏輯運算子not(否)	<code>!expr</code>
<code>^</code>	互斥exclusive or	<code>Expr ^ expr</code>
<code>&</code>	位元且運算子	<code>Expr&expr</code>
<code> </code>	位元或運算子	<code>Expr expr</code>

這些運算子的結果不是真（成立）就是假（不成立），在寫程式時的邏輯判斷經常用到，可以多練習。在這裏 `expr` 指的是運算式 `expressions`。

布林 Boolean 代數定義在一個二元素的集合上，即 $B=\{true,false\}$ ，`true` 為真，`false` 為假，再加上對兩個二元運算子 `&&` 及 `||` 的規則表。在 `&&` 運算子中只有當 `A` 和 `B` 為真時才為真。在 `OR` 運算子中，只要 `A` 或 `B` 有任何一個為真就會為真。`NOT(!)` 就是相反的意義，當 `A` 為真時 `!A` 就會為假；當 `B` 為真時 `NOT B` 就會為假，剛好和原來的 Boolean 值相反。`A^B` 就是 `A EXCLUSIVE OR B`，當只有 `A` 和 `B` 的值不相等時，才會回傳真 `true`。

在下面的真值表格 A，B 為兩運算式的結果布林值。

A	B	A AND(&&) B	A OR() B	NOT(!) B	A XOR(^) B
真	真	真	真	假	假
假	假	假	假	真	假
真	假	假	真	真	真
假	真	假	真	假	真

範例

Logical.java

第三行不為假所以為true。

第四行5大於3所以true。

第五行兩個都相等所以true。

第六行因為5>3成立，在(3>5)|| (5>3)只要有一個條件成立就為真，所以為true。

第七行3不等於5所以成立true。

第八行3>5不成立所以不成立false。

第九行為exclusive or運算子^，當兩個運算元不相等時，就會回傳true，否則會回傳false。

```
1 public class Logical{
2     public static void main(String[] args){
3         System.out.println(!(3>5));
4         System.out.println(5>3);
5         System.out.println(5==5);
6         System.out.println((3>5)|| (5>3));
7         System.out.println(3!=5);
8         System.out.println((3>5)&&(5>3));
9         System.out.println(true^false);
10        System.out.println(true^true);
11    }
12 }
```

這是執行的結果。

```
C:\Program Files\Xinox Software\JCreatorV3\JEGE2001.exe
true
true
true
true
true
false
true
false
```

2-9-2 位元邏輯運算子

這些為位元邏輯運算子，它們是以布林值作運算。

!	邏輯運算子not(否)	!expr
^	互斥exclusive or	Expr ^ expr
&	位元且運算子	Expr&expr
	位元或運算子	Expr expr

範例練習：And.java

&為位元邏輯運算子。當左邊為0而右邊為1，或左右都為0，或右邊為1左邊為0時，會傳回0。只有當左右都為1才會傳回為1。

15的二進位表示法為1111，9的二進位表示法為1001，因此在第三行15&9後，會得到1001，而1001就是十進位的9。

語法：

運算式1 & 運算式2

```
1 public class And {
2     public static void main(String[] args) {
3         System.out.println(15&9);
4     }
5 }
```

這是&位元邏輯運算的結果。



範例練習：Or.java

^為位元邏輯運算子。當左邊為0而右邊為1，或右邊為1左邊為0時，才會傳回1，其他傳回為0。

15的二進位表示法為1111，9的二進位表示法為1001，因此在第三行15^9後，會得到0110，而0110就是十進位的6。

語法：

運算式 1 ^ 運算式 2

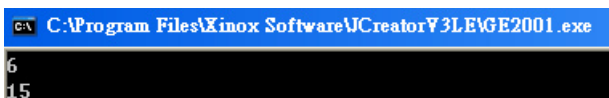
| 為位元邏輯運算子。當左邊為0或右邊為1，或右邊為1左邊為0時，或當左邊為1右邊為1，都會傳回1，只有當左右兩邊都為0時才會傳回0。15的二進位表示法為1111，9的二進位表示法為1001，因此在第三行15 | 9後，會得到1111，而1111就是十進位的15。

語法：

運算式 1 | 運算式 2

```
1 public class Or {
2     public static void main(String[] args){
3         System.out.println(15^9);
4         System.out.println(15|9);
5     }
6 }
```

15 ^ 9 為 6。15 | 9 為 15。



2-9-3 其它運算子

我們也會經常在Java中用到一些運算元。

其它運算子	說明
boolean 運算元? 運算元1: 運算元2	這就像是if else的判別式，假如運算元為真，則運算元1，否則是運算元2。運算元1和運算元2可以是原始資料型態的資料，也可以是物件。
物件 instanceof 類別	instanceof為型態比較運算子。如果物件是右邊類別的子類別或本類別的物件，則回傳，true，否則會發生編譯錯誤。

範例 Bo.java

第三行bo1為真true，所以第四行會顯示5。

第五行bo2為假false，所以第六行會顯示8。

第九行bo3為假false，所以第十行會顯示字串物件str2” 你好”。

```
1 public class Bo {
2     public static void main(String[] args){
3         Boolean bo1=true;
4         System.out.println( bo1?5:8);
5         Boolean bo2=false;
6         System.out.println(bo2?5:8);
7         String str1=new String("大家好");
8         String str2=new String("你好");
9         Boolean bo3=false;
10        System.out.println(bo2?str1:str2);
11    }
12 }
```

這是執行的情況。



範例 InstanceOf.java

Object類別為我們自訂類別的繼承類別，我們自訂的類別會直接或間接的繼承Object類別。

第一行自訂Element類別，它繼承了Object類別。extends Object，可以省略不寫。

第四行新增Element類別的物件e。

第五行比較e是否為Object類別型態的物件或子類別型態的物件，因為是真true，所以會執行第六行。

第八行到第九行則是錯誤的，因為e不是布林類別型態的物件。

```
1 class Element { int atomicNumber; }
2 class InstanceOf {
3     public static void main(String[] args) {
4         Element e = new Element();
5         if (e instanceof Object) {
6             System.out.println("e屬於Object物件");
7         }
8         /*if (e instanceof Boolean){ //會發生編譯時期錯誤
9             System.out.println("e屬於布林類別");
10        }else{
11            System.out.println("e不屬於布林類別");
12        }*/
13     }
14 }
```

這是執行的情況。



```
C:\Program Files\Xinox Software\VCreator\3LE\GE2001.exe
e屬於Object物件
```

2-10 遞增遞減運算子

遞增 (++) 和遞減 (--) 運算子提供一個方便的記號，用來將變數加一或減一，它們經常用來遞增或遞減索引值。根據運算子放置在變數的前後又可分，運算子放置在變數前面稱為前置運算子；運算子放置在變數後面稱為後置運算子。

2-10-1 遞增運算子

遞增運算子	名稱	意義
++a	前置遞增	變數a先加1再回傳
a++	後置遞增	變數a先回傳再加1

範例

Increase.java

第四行為前置遞增；將變數a加1後再輸出，所以值為2。

第六行為後置遞增；將變數a先輸出再將變數a加1。

```
1 public class Increase{
2     public static void main(String[] args){
3         int a=1;
4         System.out.println(++a);
5         a=1;
6         System.out.println(a++);
7     }
8 }
```

第四行的變數a先加1為2，再輸出顯示。

第六行的變數a++先輸出顯示為1

C:\Program Files\Xinox Software\VCcreator\3LE\GE2001.exe

```
2
1
```

2-10-2 遞減運算子

遞減運算子就是將變數減 1，根據遞減運算子的所在位址，可分為前置遞減，與後置遞減。

遞減運算子	名稱	意義
--a	前置遞減	先將變數\$a減1再傳回
a--	後置遞減	先將變數\$a傳回再減1

範例 decrease.java

第四行前置遞減運算子，先將變數a減1後再輸出顯示，所以為0。

第六行後置遞減運算子，先將變數a輸出顯示為1，所以顯示為1。

```

1 public class Decrease{
2     public static void main(String[] args){
3         int a=1;
4         System.out.println(--a);
5         a=1;
6         System.out.println(a--);
7     }
8 }
```

第六行後置遞減運算子，先將變數a輸出顯示為1，再作遞減的動作，所以顯示為1。

```

C:\Program Files\Xinox Software\VCcreator\3LE\GE2001.exe
0
1
```


2-11 運算子的優先順序

我們在作數學運算時，規則是先乘除後加減。例如 $1+2*5$ 答案大家都知到，11。程式的設計規則也是一樣，可以把運算子看作是乘和加，是有先後順序的處理。

這是運算子優先順序，++ 後置遞增的優先權最高，然後依次往下。

運算子	功能	用法
++	後置遞增	a++
--	後置遞減	a--
++	前置遞增	++a
--	前置遞減	--a
資料型態轉換	(type Casting)	
!	Not	
*	乘	$2*3$
/	除	$6/2$
%	餘數	$5\%3$
+	加	$1+1$
-	減	$2-1$
<、<=、>、>=	小於、小於等於、大於	$3<5$ 、 $3<=5$ 、 $5>=2$
=、!=	等於、不等於	$2==2$ 、 $1!=3$
&、^、 、&&、	條件運算子	true false
=、+=、-=、*=、/=、%=	指派運算子	$5=3+2$

範例

Order.java

第三行5先乘2再加3，再加上15先除以5在對2取餘數。

第四行括號裏面的先作，5乘2為10，5對2取餘數為1，15再除1

```
1 public class Order{
2     public static void main(String[] args){
3         System.out.println(3+5*2+15/5%2);
4         System.out.println(3+(5*2)+15/(5%2));
5     }
6 }
```

C:\Program Files\Xinox Software\JCreator\3LE\GE2001.exe

```
14
28
```

2-12 字串型態

char 為字元資料型態，如果要顯示字串，則要使用 String 字串參考型態。String 字串型態為參考型態，而不是原始資料型態。我們可以使用”+”運算子來將兩個字串連接起來，因此”+”在字串運算元的處理為連接運算子。”+”在數值處理則為加運算子。

範例

String_type.java

第三行我們宣告message為字串參考型態，並且將字串”歡迎來到java的世界”分配給message字串變數。

第五行使用+字串連接運算子來將字串”第”和字串”章”連接。中文字為雙位元組，每一個中文字可以當作是一個字串。

第七行使用+=運算子來將”Java簡介”和message2的字串資料連接，再給message2變數。

```
1 public class String_type{
2     public static void main(String[] args){
3         String message="歡迎來到java的世界" ;
4         System.out.println(message);
5         String message2="第"+2+"章";
6         System.out.println(message2);
7         message2 += "Java簡介";
8         System.out.println(message2);
9     }
10 }
```

這是執行的情況。

C:\Program Files\Xinox Software\JCreator\3LE\GE2001.exe

```
歡迎來到java的世界
第2章
第2章Java 簡介
```

2-13 程式設計的技巧

寫程式要養成好的寫程式習慣。在每一個重要程式的區段要加上註解，讓以後閱讀程式時可以方便了解。命名變數、物件、類別或方法時，要按照該變數的名稱，而且命名時要有意義的名子。變數或方法的識別名稱最好用英文的小寫字母開頭。類別的識別名稱最好用英文的大寫字母作開頭。常數的命名最好都是用大寫。

在程式的區塊中，它包含了許多運算式，而且使用大括號”{ “和”}”包起來。這個程式的寫法是將第一個大括號”{ “寫在行的後面。

```
1 public class Unicode_escape{
2     public static void main(String[] args){
3         System.out.println("小明說:\" Java很好用\"");
4     }
5 }
```

這個程式的寫法是將大括號換行寫。

```
1 public class Unicode_escape2
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("小明說:\" Java很好用\"");
6     }
7 }
```

範例 Programming.java

第一行我們設定Programming類別，類別名稱第一個字最好是大寫。

第五行我們設定常數PI為大寫。

```
1 public class Programming{
2     public static void main(String[] args){
3         int a=10;
4         float b=6;
5         final double PI=3.14;
6         System.out.println("圓的半徑"+a*a*PI);
7     }
8 }
```

這是執行的結果。



```
C:\Program Files\Xinox Software\Creator\Y3LE\GE2001.exe
圓的半徑314.0
```

習題

1. Java的資料包含了哪兩種資料型態？

【答案】

在Java程式語言中，每一個變數和每一個運算式在編譯時都有一種型態。這個資料包含了原始資料型態和參考型態。參考型態包含了類別型態和介面型態。

2. 何謂識別名稱？

【答案】

每一個變數都有識別名稱，我們變數宣告時，就是給該變數一個識別名稱。在circle.java範例中，radius半徑為識別名稱，area面積也是識別名稱。所有的識別名稱是由字元、數字、下底線(_)所組成。識別名稱的命名一定要以字元或下底線(_)開始，而不能以數字開始。識別名稱也不可以是Java的保留字。識別名稱也不可以是布林值(true,false或null)。識別名稱沒有長度的限制。Area、radius、good都是合法的識別名稱。而356a則不是識別名稱，因為它以數字開頭，Java編譯器會偵測出錯誤。Java是有大小寫的區隔，a和A是不一樣的變數。

3. 如何宣告變數？

【答案】

這是區域變數宣告。先宣告變數的資料類型，再宣告變數。我們可以將變數宣告寫在同一行，並且用逗號(,)分開。

4. 請簡述變數宣告？

【答案】

這是變數宣告，我們也可以使用分配符號(=)分配給它初始值。



5. 何謂分配敘述？

【答案】

6. 何謂常數？

【答案】

變數的值是會改變的，但常數的值是不會改變的。我們在資料類型前加上final關鍵字，這樣資料的值就無法改變，而成為常數。

7. 請簡述資料型態？

【答案】

byte為位元組資料型態，int是整數資料型態，float是浮點數資料型態，double是雙精度浮點數資料型態。位元大小是指該資料型態在記憶體所佔的大小。最小數值和最大數值是該資料型態的範圍大小。E+38是10的38次方。

8. 請簡述比較運算子？

【答案】

比較運算子就是用來比較兩個運算式。

9. 請簡述邏輯運算子？

【答案】

10. 請繪製真值表？

【答案】

11. 請簡述遞增遞減運算子？

【答案】

12. 請簡述運算子的優先順序？

【答案】

我們在作數學運算時，規則是先乘除後加減。例如 $1+2*5$ 答案大家都知到，11。程式的設計規則也是一樣，可以把運算子看作是乘和加，是有先後順序的處理。

13. 請簡述運算子的結合順序？

【答案】

Java是由左到右的結合優先順序。

【是非題】

1. () 在Java程式語言中，每一個變數和每一個運算式在編譯時都有一種型態。
2. () 資料包含了原始資料型態和參考型態。
3. () 參考型態包含了類別型態和介面型態。



4. () 每一個變數都有識別名稱，我們變數宣告時，就是給該變數一個識別名稱。
5. () 識別名稱的命名一定要以字元或下底線(_)開始，而不能以數字開始。
6. () 這是區域變數宣告。先宣告變數的資料類型，再宣告變數。
7. () 變數的值是會改變的，但常數的值是不會改變的。我們在資料類型前加上final關鍵字，這樣資料的值就無法改變，而成爲常數。
8. () 位元大小是指該資料型態在記憶體所佔的大小。
9. () 雙精度浮點數的位元大小是32位元。
10. () 有時一個數值運算式包含了數種的資料型態，Java允許在不同的資料型態上作運算，Java會自動轉換運算元。

【選擇題】

1. () 假如在數值運算式中，有一個運算元是雙精度浮點數(double)，則其它的資料型態就會轉成_____。
(1) 雙精度浮點數 (2) byte (3) char。
2. () 我們的程式主要是由_____與運算元所組成。
(1) 運算子 (2) 邏輯 (3) 例外
3. () 算數的加+ 減- 乘* 除÷和餘數%稱作爲_____。
(1) 算術運算子 (2) 邏輯運算子 (3) 流程運算子。

4. () 假如在數值運算式中，有一個運算元是雙精度浮點數(double)，則其它的資料型態就會轉成雙精度浮點數；否則，如果有一個運算元是浮點數(float)，這其它的運算元就會被轉成float；否則，如果有一個運算元是long資料型態，這其它的運算元就會被轉成long；否則，這兩個運算元就會被轉成整數int資料型態。這些都是Java的_____。運算元會轉型成精密度最高的數值資料型態再作運算。
- (1) 自動轉型 (2) DAO物件 (3) 多型(應用程式)。
5. () char資料型態是用於單一字元。我們使用_____來將它包住。而字串資料型態是使用雙引號將它包住。
- (1) 單引號 (2) 分號 (3) ()括號。
6. () Java使用Unicode來編碼，它是屬於16-bit的編碼方式。Unicode是由_____位元組來組成一個字，前面為\u，表示方法是4個16進位的數，從'\u0000'到'\uFFFF'。
- (1) 一個 (2) 兩個 (3) 三個。
7. () Unicode也包含了ASCII，它是從'\u0000'到_____。
- (1) '\u007A' (2) '\u00FF' (3) '\u007F'。
8. () Java允許我們使用_____來代表特殊的符號。
- (1) 整數 (2) 字串 (3) 跳脫字元。
9. () 比較運算子和邏輯運算子，所得到的結果就是一個_____ (true或false)。
- (1) 布林常數 (2) 整數 (3) 字串。



10. () 邏輯運算子可以結合條件，以一個表達式判斷許多條件，而這些條件的結果不是_____就是假false。

(1)真true (2)整數 (3)字串。

【填充題】

1. _____稱爲”與邏輯運算子”，只有當所有條件都成立時才會回傳真true，否則回傳假。
2. _____稱爲或邏輯運算子，只要運算式中一個條件成立就會回傳真true，只有當所有的條件都爲假false時，才會回傳假false。
3. _____定義在一個二元素的集合上，即 $B=\{true, false\}$ ，true爲真，false爲假，再加上對兩個二元運算子&&及||的規則表。
4. $A ______ B$ 就是A EXCLUSIVE OR B，當只有A和B的值不相等時，才會回傳真true。
5. _____和_____運算子提供一個方便的記號，用來將變數加一或減一，它們經常用來遞增或遞減索引值。根據運算子放置在變數的前後又可分，運算子放置在變數前面稱爲前置運算子；運算子放置在變數後面稱爲後置運算子。
6. 我們在作數學運算時，規則是先乘除後加減。例如 $1+2*5$ 答案大家都知到，11。程式的設計規則也是一樣，可以把運算子看作是乘和加，是有_____的處理。這是運算子優先順序，++後置遞增的優先權最高，然後依次往下。
7. char爲字元資料型態，如果要顯示字串，則要使用String字串_____型態。String字串型態爲參考型態，而不是原始資料型

態。我們可以使用”+”運算子來將兩個字串連接起來，因此”+”在字串運算元的處理為連接運算子。”+”在數值處理則為加運算子。

8. 在程式的區塊中，它包含了許多運算式，而且使用_____”{ “和” }”包起來。這個程式的寫法是將第一個大括號”{ “寫在行的後面。

