國立屏東商業技術學院 資訊工程系 物件導向程式設計

4. Java語言基礎

4.1 Data Types, Variables and Constants

4.1.1 Primitive Types

table 1彙整了java語言支援的primitive types與其記憶體空間大小、數值範圍和預設值。與C語言不同，java語言是在Virtual Machine上執行的，所以資料型態在不同平台上的大小皆相同。

Type	Size	Range	Default		
boolean	1 bit	true or false	false		
byte	8 bits	-128 ~ 127	0		
short	16 bits	-32768 ~ 32767	0		
char	16 bits	\u0000 ~ \uffff or 0~65535	\u0000		
int	32 bits	-2147483648 ~ 2147483647	0		
long	64 bits	<latex>-2	**{63}</latex> ~ <latex>2**	**{63}-1</latex>**	0
float	32 bits	32 bits IEEE 754 floating point	0.0		
double	64 bits	64 bits IEEE 754 floating point	0.0		

Tab. 1: Java Primitive Types

4.1.1.1 整數型態/Integer Types

顧名思義，整數型態就是用以表示整數的資料。Java語言中的整數型態，以integer的前三個字母int表示，唸做int或是integer都可。一個int佔記憶體32bits。其中最左邊的bit代表正負數 – 0代表正整數或0，1代表負整數。以32bits為例，最大的正整數為<latex>(0111 1111 1111 1111 1111 1111 1111 1111)_2=2,147,483,647</latex>。也就是<latex>2^{31}-1</latex>。至於最小的負數並不是<latex>(1111 1111 1111 1111 1111 1111 1111 1111)$_2$= -2,147,483,647</latex>。而是<latex>(1000 0000 0000 0000 0000 0000 0000 0000)$_2$</latex>。其值為<latex>-2,147,483,648=- 2^{32}</latex>。

Java語言還提供另外三種整數型態。byte。short與long。分別佔用不同大小的記憶體空間。其中byte為8 bits。short為16bits。long則為64bits。

4.1.1.1.1 整數數值的表達

除了宣告變數為某種型態外，我們也可以直接在程式碼中使用整數數值。整數數值可依其所使用的進位系統分成十進制(decimal, base 10)。二進制(binary, base 2)。八進制(octal, base 8)與十六進制(hexadecimal, base 16)等四種表示法。

- Decimal

    - 除正負號外，以數字0到9組成，除了數值0之外，不可以0開頭。
    - 例如：0, 34, -99393皆屬之
- Binary
    - 除正負號外，僅由數字0與1組成，必須以0b開頭。
    - 例如：0b0, 0b101, 0b111皆屬之
- Octal
    - 除正負號外，僅由數字0到7組成，必須以0開頭。
    - 例如：00, 034, 07777皆屬之
- Hexadecimal
    - 除正負號外，由數字0到9以及字母(大小寫皆可)a到f組成，必須以0x或0X開頭。
    - 例如：0xf, 0xff, 0X34A5, 0X3F2B01皆屬之

我們還可以在數值後面加上L或l強制該數值為long型態，例如：13L, 376l, 0374L, 0x3ab3L, 0xffffffL, 03273L等皆屬之。

4.1.1.2 浮點數型態/Floating Types

顧名思義，浮點數型態就是用以表示小數的資料，Java語言中有2種符點數的型態：float與double，分別實作了IEEE 754當中的單精確度與倍精確度：

- float: 單精確度浮點數(single-precision floating-point)
- double: 倍精確度浮點數(double-precision floating-point)

一般而言，float型態適用於對小數的精確度不特別要求的情況，例如體重計算至小數點後兩位、學期成績計算至小數點後一位等情況。而double則用在重視小數的精確度的場合，例如台幣對美金的匯率、工程或科學方面的應用等。

Java語言以IEEE 754標準實作了float與double，其數值範圍與精確度　請參考table 2)

Type	smallest positive value	largest value	precision		
float/single-precision	<latex>$1.17549x10^{-38}$</latex>	<latex>$3.40282x10^{38}$</latex>	6 digits		
double/double-precision	<latex>2.22507×10	**{-308}</latex>	<latex>1.79769×10	**{308}</latex>	15 digits

Tab. 2: IEEE 754標準的浮點數型態數值範圍與精確度

4.1.1.2.1 數值的表達

浮點數數值的表達有兩種方式:

- Decimal
    - 除正負號外，以數字0到9以及一個小數點組成。
    - 例如：0.0, 34.3948, 3.1415926, -99.393皆屬之。
- scientific notation
    - 由一個decimal數字與exponent組成
    - decimal數字前可包含一個正負號，數字中可包含一個小數
    - exponent表示10的若干次方，以一個E或e後接次方數表達
    - 在E或e的後面可接一個正負號，表示該次方數為正或負
    - 例如：345E0, 3.45e+1, 3.45E-5皆屬之

Java語言默認的浮點數型態為double，如果您要特別強制一個數值之型態為float，可以在數值後接上一個F

或f例如3.45f, 3.45e-3f, 1.32e+4F等皆屬之。

4.1.1.3 字元型態/Character Types

所謂的字元型態就是用以表示文字、符號等資料，在Java語言中只有一種字元型態：

- char

Java語言的char型態之大小為16 bits採用unicode編碼，可處理範圍含蓋\u0000至\uffff其實在java語言中，一個char型態的數值就是一個整數，故其範圍又可表達為整數值0~65535。例如'A'的數值為65，'0'為48，'吳'為21555等。

因此，我們可以把char型態的數值當成整數來進行運算，例如：

```
char c;
int i;

i ='a';     // i的值為97
c = 65;     // c的值為'A'
c = (char)(c + 1); // c的值為'B'
```

4.1.1.3.1 字元數值的表達

字元數值的表達方法有兩種:

- 字元值
  - 以一對單引號將字元放置其中。*例如'A', '4', ' ', '&'皆屬之。*以一對單引號將unicode的編碼放置其中。
  - 例如'\u5433'皆屬之。(unicode查詢)
- 整數值
  - 對應在字元集中的整數值
  - 例如：65, 97等皆屬之

Java語言針對一些特殊字元，提供一組escape sequence如table 3

Escape Sequence	意義
\b	Backspace
\f	form feed
\n	new line
\r	carriage return
\t	Horizontal tab
\\	backslash
\'	'
\“	”

Tab. 3: Escape Sequence

除此之外，還可以使用八進制或十六進制來表達字元：

- 八進制的escape sequence
  - 以0開頭，後面接八進制數值
  - 例如： 0101 或 0133皆屬之
- 十六進制的escape sequence
  - 以0x開頭，後面接十六進制數值
  - 十六進制的數值可以使用大寫或小寫的英文字母
  - 例如 0x41 或 0x4A 皆屬之

4.1.1.4 字串/String

所謂的字串是指一些文數字符號的集合，在Java語言中以雙引號加以包裹。例如“This is a string!”由於在Java語言中，字串是以類別的方式供我們使用，並不是資料型態的一種，在此不加以說明，細節請參考本書第XX章。

4.1.2 Variable Declaration

Java是一個strongly typed的程式語言，所有變數在使用之前必須宣告其所屬型態，但不強制在程式區塊前宣告，只要在第一次使用前宣告即可，其宣告語法如下：

```
DataType variableName[=expression][,variableName[=expression]]*;
```

<note tip> 在上面的語法說明中，「[]」為選擇性的語法單元，其後接續「*」表示該語法單元可出現0次或多次；「?」表示出現0次或1次。另外還有「+」代表1次或多次。本書將使用這種表示法做為語法的說明 </note>

其中DataType為型態，variableName為變數的名稱，中括號內的部份則是選擇性的(可以有，也可以忽略)，為該變數的初始數值。

下面的程式碼片段宣告了一個名為x的整數變數，並且在後續設定其數值為38。

```
int x;

...

x=38;
```

我們也可以將變數宣告與數值給定同時以一行程式碼來完成：

```
int x=38;
```

我們也可以同時宣告有多個相同型態的變數，例如下面的程式碼，同時宣告了三個整數變數x，y與z，其中x不指定初始值，y與z的初始值則分別為3與6。要注意的是，我們在兩個變數宣告的中間，是以','加以隔開。

```
int x, y=3, z=6;
```

variableName為變數名稱，代表在程式執行階段的某個資料項目，因此建議使用較具意義的變數名稱，才容易理解、提升程式碼的可讀性(readability)。變數的命名規則須符合識別字(identifier)命名規則：

- 只能使用英文大小寫字母、Unicode編碼字(含中文)、數字、＄與底線(_)
- 不能使用數字開頭
- 不能與Java語言的keywords或是reserved words相同

以下是一些正確的變數命名：

```
x,  _i,   $j,  sum, total, a2d, twoNumbers, averageScore, _name, c1, c0, 成績, english成績
```

以下則是一些錯誤的變數命名：

```
3x,  for,   2numbers,  average Score
```

<note tip>變數名稱、類別名稱、方法名稱等，皆屬於Java語言中的識別字(identifier)。</note>

<note important>

使用中文字為識別字命名

因為Java支援unicode編碼，所以包含中文在內的各種語言文字都可以拿來為識別字命名。支持與反對這樣做的正反意見都有，當中我認為最具說服力的理由有：

- 贊成：使用中文可以更適切地為識別字命名，提昇程式的可讀性！
- 反對：程式設計是teamwork的工作，你用中文命名讓寫出“你自己”認為很有可讀性的程式，但有沒有想過別人的想法？你有沒有試過在瀏覽外國網頁時，全部是亂碼或是全部看不懂的經驗？在不同的系統上，很有可能中文全部變成亂碼！

> 我英文不太靈光，以英文命名的識別字可能除了我之外，也沒人看得懂，還是讓我用中文吧！
>
> > 英文不靈光可以慢慢加強，用中文命名你就能保證別人一定看得懂你的意思嗎？你今天用Java寫程式，用中文用的很開心，以後換到不支援中文的語言，你又該怎麼辦？
>
> 那我該怎麼辦？
>
> > 為你的識別字用英文命名，再搭配中文註解是一個過渡期的好辦法。
>
> 我要怎麼做才能把英文學好？
>
> > 這有點離題了！不過累積與環境是很重要的！你愈不開始使用英文，你就愈不會使用！逼自己用英文寫程式，也是一種情境式的英文學習法！加油！好嗎？！

</note>

Java語言是case-sensitive的語言，意即大小寫會被視為不同的字元，因此以下的宣告其變數名稱皆是正確且不相同的：

```
int JUN, jun, Jun, JUn, JuN, jUn, jUN, juN;
```

為了讓程式碼的可讀性提升，使用有意義的變數名稱是相當重要的，有時我們甚至會使用一個以上的英文單字為變數命名，此時可以適當地調整大小寫或加上底線，例如下面是正確的宣告：

```
int bestStudent, BestStudent, best_student;
```

建議使用良好的命名規則，例如Camel Case或Hungarian Notation。目前java程式設計師通常以lower camel case法為變數、屬性、方法及物件命名，使用upper camel case法為類別命名。

<note tip>

Camel Case命名規則

以英文命名，可由多個英文單字組成，每個單字除首字母外一律以小寫表示。任何兩個單字連接時，第二個單字的首字母必須使用大寫。第一個單字的首字母若以小寫表示，則稱為lower camel case反之若第一個單字的首字母以大寫表示時，稱為upper camel case例如：

- lower camel case
  - amy, userName, happyStory, setData, getUserInput等皆屬之
- upper camel case
  - Student, FulltimeStudent, CourseTime等皆屬之

</note>

4.1.3 Keywords and Reserved Words

Java語言的Keywords與Reserved Words彙整如table 4與table 5

abstract	assert	boolean	break	byte	case	catch	char
class	continue	default	do	double	else	enum	extends
final	finally	float	for	if	implements	import	instanceof
int	interface	long	native	new	package	private	protected
public	return	short	static	strictfp	super	switch	synchronized
this	throw	throws	transient	try	void	volatile	while

Tab. 4: Java Keywords

const	false	goto	null	true

Tab. 5: Java Reserved Words

4.1.4 Constants

在程式碼中，經宣告並給定初始值後，就不再(也不允許)變更其數值的資料，就稱為常數(constant)。Java語言的常數宣告語法如下：

```
final DataType constantName=value[,constantName=value]*
```

其實，常數的宣告就如同變數宣告一樣，只要在最前面加上final這個保留字即可，同時所有常數的宣告都必須給定初始值。