

ABAP DEVELOPMENT

SAP ABAP 程式設計基礎課程 第二堂課：ABAP基本語法及編寫格式

歡迎您成為ABAP程式開發者,在本課程中,我們將瞭解到:

ABAP語言的變數定義及宣告和各常用語句

製作：周台誠
noah.chou@outlook.com

THE BEST-RUN BUSINESSES RUN SAP



- 如果要將整行變為注解，則在該行開始處輸入星號（*）
- 如果要將某行的一部分變為注釋，則在注釋之前輸入雙引號（"）。系統將由雙引號標明的注解解釋為空格。

```
1 PROGRAM SAPMTEST.  
2 * The following line contains a WRITE statement  
3 WRITE 'First Program'. " Output on List
```

範例程式的第二行是不執行的注解。注解由行開始處的星號（*）標明。
在第三行，雙引號（"）之後全部都是注解並且不執行。
程式的其餘部分為帶關鍵字**PROGRAM** 和**WRITE** 的可執行語句。

- **ABAP** 語言允許將相同起始部分的連續語句使用鏈語句來簡化程式。
將要連接分開的語句，只要寫一次相同部分，並且在其後設置冒號（:）。在冒號後 列出語句的其餘部分並用逗號（,）分開。並請在最後部分之後設置句號（.）以通知系統鏈語句的結束處。

```
1 PROGRAM SAPMTEST.  
2 * 未使用鏈語句  
3 WRITE SPFLI-CITYFROM.  
4 WRITE SPFLI-CITYTO.  
5 WRITE SPFLI-AIRPTO.
```

```
1 PROGRAM SAPMTEST.  
2 * 使用鏈語句的範例  
3 WRITE: SPFLI-CITYFROM, SPFLI-CITYTO, SPFLI-AIRPTO.  
4 * 這樣的寫法也可以  
5 WRITE: SPFLI-CITYFROM,  
6         SPFLI-CITYTO,  
7         SPFLI-AIRPTO.
```

- 基礎資料類型
是指系統內部所定義的資料類型
- 局部資料類型
在**ABAP**程式中用基礎資料類型所定義的局部資料類型
- 全域資料類型
是指所有程式都可以使用的資料類型，通常會用**SE11**來新增和定義。

- **ABAP**中預先定義的基本資料類型（有效大小以位元組為單位）
資料類型 **D**、**F**、**I** 和 **T** 都是預定義的，但資料類型 **C**、**N**、**P** 和 **X** 可以有附加選項。例如:可以在程式中定義大小。

資料類型	預設大小	有效大小	初始值	說明
C	1	1 - 65535	SPACE	文本、字元 (字母數位字符)
D	8	8	'00000000'	日期 (格式:YYYYMMDD)
F	8	8	0	浮點數
I	4	4	0	整型 (整數)
N	1	1 - 65535	'00...0'	數字字元
P	8	1 - 16	0	Packed數，用在小數點數
T	6	6	'000000'	時間 (格式:HHMMSS)
X	1	1 - 65535	X'00'	十六進制

- 整數類型 I

- 類型 I 資料的數值範圍是 -2^{31} 到 $2^{31}-1$ 並且僅包括整數。
- 對算數運算的非整型結果（如分數）進行四舍五入，而不是截斷。
- 類型 I 資料可用於計數器、項目編號、索引和時間期區段等。

● 小數類型 P

- 類型 P 資料允許在小數點後有數位。
- 類型 P 資料的數值範圍取決於大小和小數點後的位數。有效大小可以是從 1 到 16 位元組的任何值。將兩個十進位數字字壓縮到一個位元組，而最後一個位元組包含一個數字和符號。在小數點後最多允許 14 個數位。關於定義小數點後位元數的詳細資訊，參見 **DATA 語句的基本**。
- 使用類型 P 資料時，最好將程式屬性設置為“定點運算”否則，將類型 P 數位作為整數型對待。
- 類型 P 資料可用於如距離、重量和錢數等數值。

ABAP: Program Attributes ZNOAH_01_01 更改

標題	使用ABAP/4 編輯器撰寫第一支程式	
原始語言	ZF 繁體中文	
已建立	NQAH	2013/06/27
最後更改者		
狀態		

屬性	
類型	1 可執行的程式
狀態	T 測試程式
應用程式	R 未知的應用程式
權限群組	
邏輯資料庫	
選擇畫面	
☐ 編輯器鎖住	☒ 定點運算
☒ Unicode Checks Active	☐ 經由變式開始

儲存

- 浮點數類型 **F**

- 對於正數和 負數，類型 **F** 數的數值範圍是 1×10^{-307} 到 1×10^{308} ，包括 0（零）。
- 精確範圍近似為 15 位十進位數字，這取決於硬體平臺的浮點演算法。
- 由於是在內部將類型 **F** 資料轉換為二進位系統，因此可能出現舍入誤差。儘管 **ABAP/4** 處理器試圖將這些影響減至最小，但是如果要求高精度，則不應該使用類型 **F** 資料。而應代之以類型 **P** 資料。
- 如果需要大的數值範圍，而且舍入誤差不重要，則可以使用類型 **F** 資料。

- 數字類型**F,I,P**在計算中所使用的運算子和關鍵字

符號	說明	與符號相同的關鍵字
+	加法	ADD <n> TO <m>.
-	減法	SUBTRACT <n> FORM <m>.
*	乘法	MULTIPLY <m> BY <n>.
/	除法	DIVIDE <m> BY <n>.
DIV	取整數	
MOD	取余數	
**	乘積	

- 在定義物件的資料類型，有下列兩種方式：
 - 直接使用 `<declaration>...TYPE <datatype> ....`
 - 間接使用 `<declaration>...LIKE <dataobject> ....`
- 利用 **TYPE** 選項，可以直接將資料類型 `<datatype>` 分配給已宣告的資料類別。
- 利用 **LIKE** 選項，可以將另一個資料類別`<dataobject>` 的資料類型分配給已聲明的資料物件。這意味著間接引用資料類型。

- **DATA** 語句的基本格式

DATA <name>[(<length>)] <type> [<value>] [<decimals>].

- <name>變數名稱

- ◆ 最長可達 **30** 個字元。除了加號「+」，句點「.」，逗號「,」，冒號「:」，圓括號「()」這些字元外，可以使用任何字母數字字元。
- ◆ 請勿宣告完全由數字字元組成的名稱
- ◆ 不使用系統的保留名稱

- <length>變數長度

- ◆ 僅對資料類型為：**C**、**P**、**N** 和 **X** 才可以使用

- <type>變數的資料類型

- <value>指定初始值

- <decimals>指定小數位數

```
1 DATA: NUMBER          TYPE P,  
2   DATE                 TYPE D,  
3   HEXADECIMALFIELD     TYPE X,  
4   COUNT                TYPE I,  
5   LINE (72)            TYPE C,  
6   WIDTH                TYPE C LENGTH 2.
```

*宣告變數名稱為LINE的變數，其資料類型為「C」，長度為72

*也可以使用 LENGTH 語句來指定字段長度

```
9 DATA NUMBER_1 TYPE P.
```

```
10 DATA NUMBER_2 LIKE NUMBER_1.
```

*NUMBER_2 使用LIKE語句來引用NUMBER_1的資料類型

```
13 DATA: COUNTER TYPE P VALUE 1,  
14   DATE        TYPE D VALUE '19920601',  
15   FLAG        TYPE C VALUE IS INITIAL.
```

*宣告變數並給予初始值，FLAG 給予初始值為：SPACE

```
18 DATA WEIGHT TYPE P DECIMALS 2 VALUE '1.225'
```

*宣告變數及設定小數位數並予初始值

- 請使用 **DATA** 語句並用 **BEGIN OF** 和 **END OF** 標記該欄位群組的起始和結束。語法如下：

```
1 DATA: BEGIN OF ADDRESS,  
2         NAME (20)    TYPE C,  
3         STREET (20)   TYPE C,  
4         NUMBER       TYPE P,  
5         POSTCODE (5)  TYPE N,  
6         CITY (20)     TYPE C,  
7     END OF ADDRESS.
```

*該示例定義了一個名稱為：ADDRESS欄位字串其長度為73

*可以通過 ADDRESS-NAME、 ADDRESS-STREET 等來使用。

- 將已宣告的變數進行初始化恢復基本資料類型的初始值，語法如下：

```
1 CLEAR: TEST1,TEST2.
```

- 在**ABAP**中針對變數進行賦值可以使用**MOVE** 語句來執行，但一般常會用「**=**」來簡寫。而在處理偏移量時才改用**MOVE** 語句。
- 為變數賦值
語法: **MOVE** <F1> **TO** <F2>.
將**F1**的值存至變量**F2**中，也可寫成: **F2 = F1**.
- 使用偏移量
語法: **MOVE** <F1>[+<偏移量>(取位數)] **TO** <F2>[+<偏移量>].

```
1 DATA: F1(10) VALUE 'ABCDEFGHIJ',  
2         F2(5).  
3 F2 = F1+3(5). "自第4個位置開始取出5個字符  
4 WRITE / F2.   "F2的內容會變成DEFGH  
5  
6 DATA: F1(10) VALUE 'ABCDEFGHIJ',  
7         F2(5) VALUE '12345'.  
8 MOVE F1+3(5) TO F2+2. "自第4個位置開始取出5個字符，  
9                        "從F2的第三個位置開始替換  
10 WRITE / F2. "F2的內容會變成12DEF
```

- 在**ABAP**中可以針對資料結構群組進行賦值，相同的資料結構群組之間可以使用等號(=)來相互複製，並可能使用**WRITE** 語句來進行輸出，也可以使用<Structure>-<field name>來指定所要輸出的字段。

```
1  DATA: BEGIN OF UserInf,  
2      SID(10)  TYPE C, "編號  
3      NAME(20) TYPE C, "姓名  
4      TEL(20)  TYPE C, "電話  
5  END OF UserInf.  
6  
7  UserInf-SID   = 'A00001'.  
8  UserInf-NAME = 'Noah'.  
9  UserInf-TEL   = '12345678'.  
10 *使用<Structure>-<field name>來進行賦值  
11 *注意！等號(=)前後要加上空格  
12 WRITE UserInf.
```

A00001	Noah	12345678
--------	------	----------

```
1 DATA: BEGIN OF UserInf,  
2     SID(10) TYPE C, "編號  
3     NAME(20) TYPE C, "姓名  
4     TEL(20) TYPE C, "電話  
5     END OF UserInf.  
6  
7 UserInf-SID = 'A00001'.  
8 UserInf-NAME = 'Noah'.  
9 UserInf-TEL = '12345678'.  
10  
11 DATA UserInf2 LIKE UserInf.  
12 UserInf2 = UserInf.  
13 *宣告新的資料結構群組並引用UserInf的數據類型，再使用等號(=)來相互複製  
14 WRITE: UserInf2-SID,  
15     / UserInf2-NAME,  
16     / UserInf2-TEL.  
17 *使用<Structure>-<field name>來指定所要輸出的字段。
```

```
1 DATA: BEGIN OF UserInf,  
2     SID(10) TYPE C, "編號  
3     NAME(20) TYPE C, "姓名  
4     TEL(20) TYPE C, "電話  
5     END OF UserInf.  
6 UserInf-SID = 'A00001'.  
7 UserInf-NAME = 'Noah'.  
8 UserInf-TEL = '12345678'.  
9  
10 DATA: BEGIN OF UserDetail,  
11     SID(10) TYPE C, "編號  
12     NAME(20) TYPE C, "姓名  
13     BIRDATE TYPE D, "出生日期  
14     TEL(20) TYPE C, "電話  
15     ADDR(70) TYPE C, "地址  
16     END OF UserDetail.  
17 MOVE-CORRESPONDING UserInf TO UserDetail.  
18 *使用MOVE-CORRESPONDING來相互複製.  
19 *如果字段名稱相同就進行複製，其他不同名稱的字段就為空白。  
20 WRITE: UserDetail.
```

- 使用**INCLUDE STRUCTURE**語句來繼承

```
1 DATA: BEGIN OF UserInf,  
2     SID(10) TYPE C, "編號  
3     NAME(20) TYPE C, "姓名  
4     TEL(20) TYPE C, "電話  
5 END OF UserInf.  
6  
7 DATA: BEGIN OF UserDetail.  
8     INCLUDE STRUCTURE UserInf.  
9 DATA: BIRDATE TYPE D, "出生日期  
10     ADDR(70) TYPE C, "地址  
11 END OF UserDetail.  
12 *使用INCLUDE STRUCTURE語句來繼承UserInf  
13 *並新增二個字段
```

- 資料結構群組之間可以使用**MOVE-CORRESPONDING <n> TO <m>**.來相互複製，利用下面語句可以進行資料結構群組之間的計算操作。
 - **ADD-CORRESPONDING <n> TO <m>**.
 - **SUBTRACT-CORRESPONDING <n> FORM <m>**.
 - **MULTIPLY-CORRESPONDING <m> BY <n>**.
 - **DIVIDE-CORRESPONDING <m> BY <n>**.

- **CONSTANTS** 語句的基本格式

CONSTANTS <name>[<length>] <type> <value> [<decimals>].

- 如果要宣告資料結構群組定義為常數

CONSTANTS: BEGIN OF <fstring>,
 <component declaration>

.....

END OF <fstring>.

```

1  CONSTANTS PI TYPE P DECIMALS 5 VALUE '3.14159'.
2  CONSTANTS: MYNAME(10)           VALUE 'Fred',
3  BIRTHDAY TYPE D VALUE '19600110',
4  ZERO TYPE I VALUE IS INITIAL.
5  CONSTANTS: BEGIN OF MYADDRESS,
6  NAME(20) TYPE C VALUE 'Fred Flintstone',
7  STREET(20) TYPE C VALUE 'Cave Avenue',
8  NUMBER TYPE P VALUE 11,
9  POSTCODE(5) TYPE N VALUE 98765,
10 CITY(20) TYPE C VALUE 'Bedrock',
11 END OF MYADDRESS.
```

- 使用 **TYPES** 語句可新增用戶定義的基本資料類型和結構化數據類型。
- 透過**TYPES** 語句的參數可以被其他變數或是常數所引用，但不能直接賦值。語法如下：
TYPES <t>[<length>] <type> [<decimals>].
<t>該語句所定義資料類型。
- 要定義資料結構群組，請寫：
TYPES: BEGIN OF <fstring>,
 <component declaration>,

END OF <fstring>.

```
1 TYPES:  NAME (10) TYPE C,  
2          TEAMS (20) TYPE C.
```

3 **使用TYPES語句自行定義了NAME和TEAMS這兩個數據類型*

```
4 DATA:   PLAYER TYPE NAME VALUE 'Noah',  
5          NOCKNAME LIKE PLAYER.
```

6 **在宣告變數時引用自行定義的NAME數據類型，需使用TYPE語句*

7 **LIKE語句只能用在引用已宣告的變數*

```
8 CONSTANTS: TEAM1 TYPE TEAMS VALUE 'Team1',  
9              TEAM2 LIKE TEAM1 VALUE 'Team2'.
```

10 **在宣告常數時引用自行定義的TEAMS數據類型。*

```
12 DATA     COUNTS TYPE I.
```

```
13 TYPES:   COMPANY LIKE SPFLI-CARRID,  
14          NO_FLIGHTS LIKE COUNTS.
```

15 **說明如何通過引用已有資料物件類型來新增數據類型。*

16 **並且還可以新增引用至ABAP/4 詞典對象的類型。*

17 **COMPANY 所引用 ABAP/4 詞典對象CARRID是資料庫表 SPFLI 的一列。*

- **STATICS** 靜態變數宣告

宣告的變量僅在目前的**FORM**中使用，結束後會自動釋放。

語法: **STATICS** <c> [<length>] <type> [<value>] [<decimals>]

- **TABLES**宣告

用來宣告資料表工作區的數據，對應至**ABAP/4**字典對象(Dictionary Object)，由**SQL**命令加載所需數據

語法: **TABLES** <dbtab>