

Programming

Pro Language

- Pro語言是一種有結構化程序設計，具有變量作用域以及遞歸功能的過程式編程語言。
- Pro語言的代碼支持Unicode字符集，且區分大小寫。但是你無須為忘記關鍵字的大小寫擔心，因為所有的關鍵字都是首字母大寫的。
- Pro語言的主過程為Main函數。
- Pro語言可以通過頭文件導入調用其他Pro編程文件中的數據類型函數與過程。
- Pro語言的所有函數和過程（頭文件導入函數除外）均調用了其它Pro編程文件，用戶可以根據自己的喜好更改所有Pro語言中的數據類型、函數與過程的名稱，實現了名稱高度自由化。
- Pro語言同時支持調用其他編程語言來編程，但是在你調用該語言前，你得先提供該語言的編譯器。
- 由於支持Unicode字符集且可以更改所有數據類型、函數與過程的名稱，不同國家和地區的用戶也可以通過下載對應的語言包來輕鬆通過自己的母語來編程（語法還需按照Pro語言的語法）。
- 我們正在努力開發新型的IDE，將來Pro語言的用戶能使用程式模塊編程。屆時，即使不懂得Pro語言的語法的用戶也能使用Pro語言進行編程，也能讓來自不同國家和地區的Pro語言用戶交流共享自己的代碼。

Basic Data Type

Integer

數據類型

- Integer類型就是整數類型（簡稱整型）。它允許用戶儲存一個在其數據範圍內的整數。
- Integer類型是Pro語言中最重要數據類型之一。在程序中將會經常用到這種數據類型（尤其是在循環計數和數組下標中）。
- 請注意區分Integer類型和Float類型之間的差別，一些函數或過程只允許Integer類型作為其參數，而不允許Float類型作為其參數。

定義方式

```
Dim [Varname] As [Signed/Unsigned] [Shortint/Integer/Longint/Number];
```

注意事項

1. Varname為變量名（必填）。

2. Signed/Unsigned爲可選項。Signed爲有符號整型，Unsigned爲無符號整型。當這一選項被省略（不填）時，將默認爲Signed。
3. ShortInt/Integer/Longint?Number爲必選項。Shortint爲短整型，Integer爲標準整型，Longint爲長整型，Number爲大整型。

例子

1. 定義無符號長整型變量i，應使用以下語句：

```
Dim i As Unsigned Longint;
```

Float

數據類型

- Float類型就是浮點數類型（簡稱浮點型）。它允許用戶儲存一個在其數據範圍內的浮點數。
- 請注意，浮點數不是實數。浮點數可能會跟實際數字存在誤差（尤其當該數字是一個高位小數時），所以計算結果也會存在差異（甚至不同的浮點類型都可能帶來計算結果上的巨大差異）。
- 浮點數計算不符合部分數學規律（或定理）。若要進行精密計算，建議使用高精度計算。

定義方式

```
Dim [Varname] As [Signed/Unsigned] [Float/Single/Double/Real];
```

注意事項

1. Varname爲變量名（必填）。
2. Signed/Unsigned爲可選項。Signed爲有符號浮點型，Unsigned爲無符號浮點型。當這一選項被省略（不填）時，將默認爲Signed。
3. Float/Single/Double/Real爲必選項。Float爲標準浮點型，Single爲單精度浮點型，Double爲雙精度浮點型，Real爲高精度浮點型。

例子

1. 定義（有符號）單精度浮點型變量f，應使用以下語句：

```
Dim f As Single;
```

Char

數據類型

- Char類型就是字符類型（簡稱字符型）。它允許用戶儲存一個字符。
- 一個字符是指在字符集編碼中占一個字元的符號或對其它顯示設備的命令。

定義方式

```
Dim [Varname] As [Char/Unicode];
```

注意事項

1. Varname為變量名（必填）。
2. Char/Unicode為必選項。Char為ASCII碼字符型，Unicode為Unicode字符型。

例子

1. 定義Unicode字符型變量u，應使用以下語句：
Dim u As Unicode;

String

數據類型

- String類型就是字符串類型（簡稱字符串型）。它允許用戶儲存一個字符串。
- 一個字符串是指一個由若干個字符組成的有序序列。

定義方式

```
Dim [Varname] As [Limited/Endless] [String/Unistring];
```

注意事項

1. Varname為變量名（必填）。
2. Limited/Endless為可選項。Limited為長度為255字符的字符串，Endless為無限長度的字符串（最大字符量根據可分配的內存決定）。當這一選項被省略（不填）時，將默認為Limited。
3. String/Unistring為必選項。String為ASCII碼字符串型，Unistring為Unicode字符串型。

例子

1. 定義不限長度的ASCII碼字符串型變量s，應使用以下語句：

```
Dim s As Endless String;
```

Enumeration

數據類型

- Enumeration類型就是枚舉類型（簡稱枚舉型）。它允許用戶儲存一個（標識符常數表示的）枚舉標識符。

定義方式

```
Dim [Varname] As Enum Of
{
    [Enum Label] [= Label Index];
    [Enum Label] [= Label Index];
    [Enum Label] [= Label Index];
    ...
};
```

注意事項

1. Varname為變量名（必填）。
2. Enum Label為枚舉標識符（必填）。
3. Label Index為標識符常數（選填）。Label Index應為無符號整型數值。若不填寫Label Index，則默認為比上一個Label Index大的最小整數（第一個Label Index默認為0）。
4. 枚舉標識符應根據Label Index從小到大的書序排列，否則可能發生嚴重錯誤。

例子

1. 以下定義語句中，標識符為abc的Label Index為0，標識符ssd的Label Index為9，標識符trd的Label Index為10，標識符foh的Label Index為30：

```
Dim z As Enum Of
{
    abc;
    ssd=9;
    trd;
    foh=30;
};
```

2. 定義標識符爲first（Label Index=1）、second（Label Index=2）、third（Label Index=3）、fourth（Label Index=4）的枚舉型變量e，應使用以下語句①，語句②是錯誤的：

①.

```
Dim e As Enum Of
{
    first=1;
    second=2;
    third=3;
    fourth=4;
};
```

②.

```
Dim e As Enum Of
{
    second=2;
    fourth=4;
    first=1;
    third=3;
};
```

Boolean

數據類型

- Boolean類型就是布爾類型（簡稱布爾型）。它允許用戶儲存一個布爾值。
- 布爾值有兩個，分別是True（真）和False（假）。
- Boolean是計算機科學中的邏輯類型。

定義方式

```
Dim [Varname] As Boolean;
```

注意事項

1. Varname爲變量名（必填）。

例子

1. 定義布爾型變量b，應使用以下語句：

```
Dim b As Boolean;
```

Address

數據類型

- Address類型就是地址類型（簡稱地址型）。它允許用戶儲存一個物理內存地址。

定義方式

```
Dim [Varname] As Address;
```

注意事項

- Varname爲變量名（必填）。

例子

- 定義地址型變量g，應使用以下語句：

```
Dim g As Address;
```

Variant

數據類型

- Variant類型就是變體類型（簡稱變體型）。它允許用戶儲存任意一種基本數據類型（包括Integer、Float、Char、String、Enumeration、Boolean、Address等）的一個值。

定義方式

```
Dim [Varname] As Variant;
```

注意事項

- Varname爲變量名（必填）。

例子

1. 定義變體型變量v，應使用以下語句：

```
Dim v As Variant;
```

Advanced Data Type

Array

數據類型

- Array類型就是數組類型（簡稱數組）。它允許用戶在一段連續的內存裡存儲若干個相同數據類型的值。
- Array類型裡不同值的訪問是通過數組的下標實現的，訪問這些值時應該給出其對應的數組下標。

定義方式

```
Dim [Varname] As Array [First Index..Last Index] Of [Data Type];
```

注意事項

1. Varname為變量名（必填）。
2. First Index..Last Index為數組下標範圍（選填）。First Index為首項數組下標，Last Index為末項數組下標。當這一選項被省略（不填）時，將默認為可變數組（在程序中可改變數組長度）。
3. First Index和Last Index均只能為字符型或整型數值。
4. Data Type為數組元素的數據類型（必填）。數組元素可以為所有數據類型。
5. 欲定義多維數組，請使用數組的嵌套。

例子

1. 定義一維半精度浮點型數組f[1..4]，應使用以下語句：

```
Dim f As Array [1..4] Of Float;
```

2. 定義一維地址型指針可變數組l，應使用以下語句：

```
Dim l As Array Of ^Address;
```

3. 不能定義一維變體型指針數組k[(1.1)..(1.2)]，即以下語句是錯誤的：

```
Dim k As Array [(1.1)..(1.2)] Of ^Variant;
```

4. 定義三維布爾型可變數組t[1..5][1..6]（第三維可變），應使用以下語句：

```
Dim t As Array [1..5] Of Array [1..6] Of Array Of Boolean;
```

Set

數據類型

- Set類型就是集合類型（簡稱集合）。它允許用戶存儲一個由若干個由一種數據類型的值構成的元素構成的集合。
- Set類型允許用戶對其進行三種基本操作：向集合內添加新的元素、刪除集合內已有的元素、判斷一個元素是否在集合內。

定義方式

```
Dim [Varname] As Set Of [Basic Data Type];
```

注意事項

- Varname為變量名（必填）。
- Basic Data Type為集合元素的數據類型（必填）。集合元素只能為Integer、Float、Char、String、Boolean、Enumeration、Address、Variant等數據類型，不能為Array、Set、Record、Pointer等數據類型。

例子

- 定義ASCII碼字符型集合c，應使用以下語句：

```
Dim c As Set Of Char;
```

- 不能定義變體型指針集合w，即以下語句是錯誤的：

```
Dim w As Set Of ^Variant;
```

Record

數據類型

定義方式

```
Dim [Varname] As Record Of  
{
```



```
[Record Name] As [Data Type];  
[Record Name] As [Data Type];  
[Record Name] As [Data Type];  
...  
};
```

注意事項

- 1. Varname為變量名（必填）。
- 2. Record Name為記錄的屬性名（必填）。
- 3. Data Type為記錄對應屬性的數據類型（必填）。記錄屬性可以為所有數據類型。

例子

- 1. 定義包含單精度浮點型屬性.ert、字符串型集合屬性.hom的記錄r，應使用以下語句：

```
Dim f As Record Of  
{  
    ert As Single;  
    hom As Set Of String;  
};
```

Pointer

數據類型

定義方式

```
Dim [Varname] As ^[Data Type];
```

注意事項

- 1. Varname為變量名（必填）。
- 2. Data Type為指針元素的數據類型（必填）。指針元素可以為所有數據類型。

例子

- 1. 定義標識符為right、wrong的枚舉型指針p，應使用以下語句：

```
Dim g As ^Enum Of  
{  
    right;  
    wrong;  
};
```

2. 定義短整型可變數組指針q，應使用以下語句：

```
Dim q As ^Array Of Shortint;
```