



LOGALI

TEORÍA

Datos en ABAP

SAP ABAP Programación





Contenido

1. Buffer de Programa y el Área Roll.....	3
2. Objetos de Datos	3
3. Variables	4
4. Tipos estándar ABAP	5
4.1. Tipos estándar ABAP completos	5
4.2. Tipos estándar ABAP incompletos	6
5. Constantes	6



1. Buffer de Programa y el Área Roll

Los Programas son almacenados en el servidor de aplicaciones en un buffer de programa. Cuando un usuario pide ejecutar un programa, una búsqueda es realizada en el buffer. Si es encontrado, y no ha sido modificado en la base de datos, la copia almacenada es usada. Si no se encuentra, o si la copia de la base de datos es más nueva, el programa se vuelve a cargar.

Si varios usuarios quieren ejecutar un programa, se usa la misma copia. La diferencia está en un área separada de memoria llamada roll área. Un área roll es creada por cada usuario y programa.

El sistema usa el roll área para almacenar la información sobre la ejecución del programa y las asignaciones de memoria. La información de las variables y sus valores, el puntero al programa, y la lista de salida son guardadas en el roll área.

Por ejemplo, suponemos que un usuario ejecuta un programa y un área roll es asignada. Si, sin esperar para terminar, el usuario cambia a otra sesión y empieza el mismo programa otra vez, otra área roll es asignada para la segunda ejecución del programa.

El usuario tiene dos áreas roll, una para cada ejecución del programa. Si el usuario ejecuta otro programa, el usuario tendrá dos roll áreas, una para cada programa.

2. Objetos de Datos

Los objetos de datos son localizaciones de memoria que usamos para almacenar datos mientras el programa se está ejecutando.

- Objetos de datos Modificables
 - Variables
 - Field strings=estructuras
 - Internal tables=arrays
- Objetos de datos No Modificables
 - Literales
 - Constantes

Cuando comienza el programa, el espacio de memoria para cada objeto de datos se ubica en el área roll del programa.

Cuando el programa se está ejecutando, se puede leer el contenido de los objetos de datos y modificar los objetos de datos modificables. Cuando el



programa termina, el sistema libera la memoria para todos los objetos y su contenido se pierde.

3. Variables

Todas las variables deben ser declaradas antes de ser utilizadas. La forma de declarar una variable es: nombre, tipo, longitud (si aplica), modificadores adicionales (por ejemplo, número de decimales para los datos empaquetados) y opcionalmente el valor inicial.

Dos sentencias son las usadas comúnmente para definir variables en un programa ABAP/4: DATA y PARAMETERS.

Declarar variables con DATA, consiste en dar un nombre, tipo y opcionalmente inicializar una variable.

Declarar variables PARAMETERS, es parecido a data, con la diferencia que hay que indicar como se presentan las variables en pantalla.

- Nombres de las variables:
 - Los nombres de variable pueden ser de 1 a 30 caracteres de largo.
 - Puede contener cualquier carácter excepto: () + . , :
 - Tienen que contener al menos un carácter alfabético.
- SAP recomienda
 - Siempre comenzar el nombre de variable con un carácter.
 - No utilizar guiones, en su lugar usar subrayado.
 - No usar USING o CHANGING como nombres de variables.
- Otras cuestiones sobre variables:
 - La longitud por defecto depende del tipo de dato.
 - El tipo por defecto es c (carácter).
 - El valor inicial por defecto es 0, salvo el tipo c que es blanco.
 - La inicialización, sólo acepta literales o constantes (más tarde las definiremos) no se pueden usar variables.
 - La sentencia data puede aparecer en cualquier lugar del programa. La única restricción es que no puede usarse una variable antes de ser declarada por la sentencia data.



4. Tipos estándar ABAP

ABAP es un lenguaje que maneja un amplio espectro de datos y de tipos de datos. Posee algunos tipos de datos elementales sobre los cuales se pueden crear otros muy complejos.

Los tipos estándar ABAP predefinidos por SAP (tipos implementados) están divididos en dos grupos:

- Tipos completo
- Tipos incompleto

Los tipos estándar ABAP completos contienen una especificación de longitud fija específica del tipo.

En cambio, los tipos incompletos no tienen una longitud fija. La longitud se debe especificar junto con la declaración de la variable.

Como la mayoría de los lenguajes modernos, ABAP tiene un conjunto de tipos de datos primitivos sobre los cuales se crean los tipos de datos más complejos.

4.1. Tipos estándar ABAP completos

- D - tipo para fechas, formato: AAAAMMDD, longitud 8 (fija)
- T - tipo para hora, formato: HHMMSS, longitud 6 (fija)
- I - tipo para entero, longitud 4 (fija)
- F - tipo para número de punto flotante, longitud 8 (fija)
- DECFLOAT16 - tipo de número DECimal de punto FLOTante, longitud 8 (fija)
- DECFLOAT34 - tipo de número DECimal de punto FLOTante, longitud 16 (fija)
- STRING - tipo de cadena de caracteres de longitud dinámica
- XSTRING - tipo para secuencia de bytes de longitud dinámica (cadena hexadecimal)



4.2. Tipos estándar ABAP incompletos

- C - tipo para cadena de caracteres para la que se debe especificar la longitud fija
- N - tipo para cadena de caracteres numérica para que se debe especificar la longitud fija
- X - tipo para secuencia de bytes (cadena hexadecimal) para la que se debe especificar la longitud fija
- P - tipo para número empaquetado para el que se debe especificar la longitud fija. En la definición de un número empaquetado, es posible que también se especifique la cantidad de puntos decimales.

5. Constantes

Al contrario que una variable, las constantes que se declaran al principio del programa nunca podrán modificar su valor durante la ejecución del programa. En un programa se utilizan para evitar escribir valores literales. La sentencia utilizada es la sentencia **CONSTANTS**.

Ejemplo de la sentencia CONSTANTS:

```
CONSTANTS: c_vuelo(5) TYPE c VALUE 'Vuelo',  
           c_fecha   TYPE d VALUE '20121231'.
```

En este caso vemos que la declaración de constantes y variables es igual, la diferencia radica en la cláusula **VALUE**. Así dotamos de contenido a nuestra constante durante toda la ejecución del programa. No se puede asignar un valor a una constante durante la ejecución del programa ya que el lenguaje ABAP no acepta.