



aprenderaprogramar.com

Arrays estáticos y arrays dinámicos (arreglos). Ventajas e inconvenientes (CU00212A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº11 del Curso Bases de la programación Nivel II

24

MANEJO DE DATOS CON ARRAYS. LÍMITES SUPERIOR E INFERIOR.

No resultará difícil comprender que los arrays estáticos gozan de una serie de ventajas frente a los dinámicos como:

- Son más seguros: al no haber redimensionamientos, el riesgo de errores es menor. También hay menos problemas en cuanto a que datos válidos sean eliminados por error o a que existan datos no válidos en el array.
- Son más claros: no hay equívocos en cuanto al número de elementos que los componen y son más fáciles de seguir.
- Ocupan menos recursos del ordenador, que tiene que gestionar su contenido pero no cambio de dimensiones.

Por el contrario, los arrays estáticos nos limitan en cuanto a capacidad de maniobra frente a las circunstancias y en cuanto a posibilidades de una adaptación perfecta entre el número de datos válidos y el número de elementos del array. La forma de gestión del array será decisión del programador, pero en general interesará que el número de elementos del array se ajuste al número de datos válidos.

Supongamos que guardamos en un array la cantidad de unidades de un producto a fabricar, de acuerdo con el número de pedido:

```
Cantidad(1) = 3200 [El pedido nº 1 es de 3200 unidades]
Cantidad(2) = 2750
Cantidad(3) = 2650
Cantidad(4) = 3325
Cantidad(5) = 2950
```

Supongamos ahora que los pedidos 1, 3 y 4 son cancelados por los clientes. Actualizaríamos el array.

```
Cantidad(1) = 0
Cantidad(2) = 2750
Cantidad(3) = 0
Cantidad(4) = 0
Cantidad(5) = 2950
```

Con un array estático, así nos quedaríamos al no poder variar el número de elementos. La gestión de estos datos es posible introduciendo un control para no procesar los datos que son cero. Pero más allá de que sea posible, ¿cuáles son los problemas potenciales?

- Almacenar datos no válidos es ineficiente, al ocupar memoria y tener que realizar la gestión de elementos que no aportan nada. Si extrapolamos un caso como el anterior, con $(3 / 5) * 100 = 60\%$ de datos no válidos, a un array de 1500 elementos, resultaría que 900 elementos no son válidos. El 60% de la memoria y tiempo de gestión empleados serían inútiles.
- Aumenta el riesgo de errores de interpretación. El no ajustar los datos existentes y datos válidos nos obligará a llevar una contabilidad paralela consistente en determinar:

$$\text{Datos válidos} = \text{Datos totales} - \text{Datos nulos}$$

¿Para qué llevar una doble contabilidad cuando podemos llevar sólo una? Por otro lado, téngase en cuenta que es frecuente prestar atención a datos totales como datos válidos. Por ejemplo:

El director de una fábrica consulta en el ordenador el número de pedidos y constata que es 5, pero no se percató de que tres de ellos han sido anulados. Sabe que por cada pedido se gastan 50 Kgs de colorante, independientemente de la cantidad de unidades a fabricar (en cada lote se renueva desechándose el anterior) y por ello encarga 250 Kgs de colorante erróneamente.

En una consulta médica hay previstas 3 consultas durante una tarde. Al no aparecer los pacientes, el médico pregunta por lo que ocurre. Tras chequear diferentes libros y apuntes la enfermera responde: “Los pacientes llamaron y cancelaron la consulta, pero olvidé traspasar mis anotaciones al libro de citas”.

Errores u olvidos asociados a arrays mal gestionados suelen aparecer en programas largos. Lo anterior no quiere decir que obligatoria y continuamente hayamos de estar redimensionando arrays para ajustarlos al número de datos válidos. Sabiendo la problemática asociada, habrá que tomar decisiones oportunas en función de las circunstancias.

Un recurso habitual en la gestión de arrays es el uso de los valores máximos de los localizadores para determinar el número de elementos del array cuando este dato no es conocido o evidente. Así, es habitual usar expresiones como:

- Tengo una lista de 17 libros. En vez de tengo un libro nº 1, tengo un libro nº 2, tengo un libro nº 3, etc.
- Estoy trabajando con una matriz de 2 x 4 elementos. En vez de estoy trabajando con el elemento (1,1), el elemento (1,2), el elemento (1,3), el elemento (1,4), el elemento (2,1), ...
- Tengo una tabla con siete tipos de aceites y tres posibles valoraciones del consumidor. En vez de tengo una tabla con aceite de oliva, aceite de palma, aceite de cacahuete, ... y valoración excelente, valoración buena y valoración mala.

Y a su vez: Voy a quedarme con 15 libros. Voy a ampliar la matriz a 3 x 4. Voy a añadir dos tipos más de aceite.

Próxima entrega: CU00213A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60