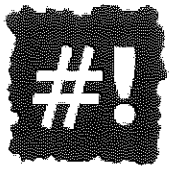


Preparador Informática

www.preparadorinformatica.com

TEMA 11. INFORMÁTICA / S.A.I.

ORGANIZACIÓN LÓGICA DE LOS DATOS. ESTRUCTURAS ESTÁTICAS.

TEMA 11 INF / SAI. ORGANIZACIÓN LÓGICA DE LOS DATOS. ESTRUCTURAS ESTÁTICAS.

1. INTRODUCCIÓN

En el estudio de la informática es fundamental el conocimiento de las estructuras de datos, ya que estas se consideran la base de todo el proceso de información. Para procesar la información es necesario hacer una abstracción de los datos que tomamos del mundo real, estructurándolos de una determinada forma y formato (organización lógica de los datos) para obtener un rendimiento adecuado en su tratamiento.

Por tanto, la eficiencia y velocidad de ejecución de un programa en un ordenador además de depender del tipo de ordenador en el que se ejecute, de los algoritmos empleados y del compilador, dependerá en gran medida de la organización lógica de los datos.

Actualmente existen sistemas de análisis de datos y big data que trabajan no solamente con datos estructurados sino también con datos no estructurados. Esto ha supuesto un desafío para poder organizar y homogeneizar datos de diferentes fuentes como ayuda a una toma de decisiones más eficiente por parte de las organizaciones.

El presente tema está dedicado a estudiar la organización lógica de los datos, y concretamente los tipos y características de las estructuras estáticas.

2. ORGANIZACIÓN LÓGICA DE LOS DATOS

2.1. CONCEPTOS

2.1.1. DATO

Se denomina dato a cualquier objeto manipulable por el ordenador. Es decir, un dato puede ser un carácter leído de un teclado, información almacenada en un disco, un número que se encuentra en memoria principal, etc.

2.1.2. ESTRUCTURAS DE DATOS

Una estructura de datos es una forma de organizar un conjunto de datos elementales con objeto de facilitar la manipulación de estos datos.

Las estructuras de datos se caracterizan por:

- La eficiencia en el uso de la memoria
- La velocidad de acceso a los datos

2.2. TIPOS

2.2.1. TIPOS DE DATOS

Podemos clasificar los tipos de datos que se emplean en los lenguajes de programación en dos tipos:

- **Datos simples**: tienen como característica común que cada variable representa a un elemento. Son aquellos cuyos valores son atómicos y, por tanto, no pueden ser descompuestos en valores simples. Ejemplos: entero, real, carácter, Etc.
- **Datos compuestos**: están construidos basándose en tipos de datos simples. Ejemplos: matriz, cadena de caracteres, Etc.

2.2.2. TIPOS DE ESTRUCTURAS DE DATOS

Los tipos de datos se pueden organizar en diferentes estructuras atendiendo a la forma en que ocupan el espacio en memoria:

- **Estáticas**: El tamaño ocupado en la memoria se define antes de que el programa se ejecute y no puede modificarse durante la ejecución. Ejemplos: arrays.
- **Dinámicas**: El tamaño ocupado en memoria puede cambiar en tiempo de ejecución. Ejemplos: listas (enlazadas, pilas, colas), árboles y grafos.

3. ESTRUCTURAS ESTÁTICAS

3.1. ARRAYS

Un array es una estructura de datos homogénea, es decir, está formada por una cantidad fija de datos de un mismo tipo donde cada uno de los cuales tiene asociado uno o más índices que determinan de forma univoca la posición del dato dentro del array.

En general, al número de índices del array se le denomina número de dimensiones del array. Los arrays según el nº de dimensiones pueden ser:

- Arrays unidimensionales.
- Arrays de varias dimensiones.

3.1.1. ARRAYS UNIDIMENSIONALES

Se denomina **array unidimensional** a un array de una dimensión. Por ejemplo, un array unidimensional de números reales de n elementos se puede representar de la siguiente forma:

4.45	6.30	.....	7.75

a) Declaración:

Un array unidimensional se declara de forma general de la siguiente manera:

```
<identificador>: array (<valor_inicial>….<valor_final>) de <tipo_dato>
```

donde:

- `<valor_inicial>` y `<valor_final>` delimitan el rango de elementos posibles. Estos valores han de ser constantes.
- `<tipo_dato>` hace referencia al tipo de dato común a todas las componentes del array.
- `<identificador>` hace referencia al nombre con el que se identifica al array.

Ejemplos: Las siguientes sentencias declaran un array unidimensional de números reales llamado `notas` en diferentes lenguajes de programación:

C++

```
float notas[5];
```

Java

```
float [] notas = new float[5]; /
```

Pascal

```
var notas: array[1…5] of real;
```

b) Almacenamiento:

Al declarar un array se reserva una cantidad fija de memoria e inalterable durante toda la ejecución del programa, ocupando cada componente un espacio igual al tipo de dato que aparece en la declaración, y estando todas ellas en posiciones contiguas de memoria. Dado que la cantidad de memoria que se reserva es fija e inalterable se considera una estructura estática.

c) Operaciones con arrays unidimensionales

- Asignación de datos:

  Asigna un valor a un elemento del array.

  Ejemplo en Java: Asigna el valor 99 al primer elemento del array `numeros`

  ```
  numeros[0] = 99 ;
  ```

- Acceso a datos:

  El acceso a un valor ya existente dentro de una posición del array se consigue de forma similar, simplemente poniendo el nombre del array y la posición a la cual se quiere acceder.

- Recorrido del array:

  Acción de lectura o escritura sobre todos los componentes del array. Ejemplo en pseucódigo: para recorrer el array `aux`, imprimiendo todos sus valores.

  ```
  For indice = 1 to tamaño_indice
       Escribir aux(indice)
  End
  ```

3.1.2. ARRAYS MULTIDIMENSIONALES

Un array multidimensional es una estructura de datos homogénea, es decir, formada por una cantidad fija de datos de un mismo tipo, donde cada uno de los cuales tiene asociado dos o más índices que determinan de forma univoca la posición del dato dentro del array.

a) Declaración:

Un array multidimensional se declara de forma general de la siguiente manera:

```
<identificador>: array (<vi1>….<vf1>,… <viN>….<vfN>) de <tipo_dato>
```

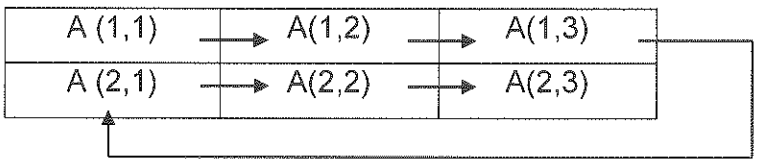
donde:

- `<vi1>, <vf1>,…,<viN>, <vfN>` representan los valores iniciales y finales que delimitan el rango de los valores que podrán tomar cada uno de los correspondientes índices.
- `<tipo_dato>` hace referencia al tipo de dato común a todas las componentes del array multidimensional.
- `<identificador>` hace referencia al nombre con el que se identifica al array.

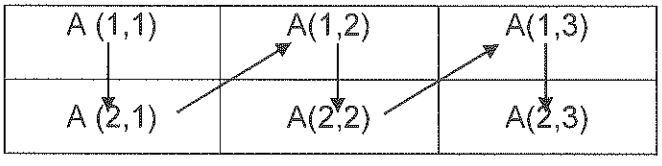
b) Almacenamiento:

Al ser la memoria del ordenador lineal, un array multidimensional debe estar linealizado para su disposición en el almacenamiento. En general, los lenguajes de programación pueden almacenar los arrays en memoria de dos formas:

- Orden de fila mayor



- Orden de columna mayor



c) Operaciones con arrays multidimensionales

Las operaciones con arrays multidimiensionales son las mismas que las descritas para los arrays unidimensionales:

- Asignación de datos
- Acceso a datos:
- Recorrido del array multidimensional:

3.2. CADENAS DE CARACTERES

Una cadena de caracteres es una estructura de datos formada por una secuencia de caracteres. Según el lenguaje de programación, las cadenas pueden ser de naturaleza dinámica (por ejemplo, en lenguaje Java) o de naturaleza estática (por ejemplo, en C). En este segundo caso, la manera más frecuente de implementar las cadenas es por medio de un array de caracteres alfanuméricos.

Las principales operaciones que se pueden realizar sobre cadenas de caracteres son:

- **Asignación**: consiste en guardar una cadena en una variable de tipo cadena.
- **Concatenación**: es la unión de dos cadenas, para formar una sola.
- **Extracción de subcadena**: permite extraer una subcadena de otra de mayor tamaño.
- **Comparación de cadenas**: permite comparar dos cadenas entre sí lexicográficamente.
- **Longitud de cadena**: número de caracteres que contiene la cadena.

3.3. REGISTROS

Un registro es un tipo de dato estructurado con un número fijo de componentes llamados campos que no son necesariamente del mismo tipo y a los que se accede por su nombre, y no por su posición.

Un registro viene definido por un conjunto de campos que tienen: un nombre, un orden y un tipo de dato asociado.

Ejemplo en pseudocódigo:

```
Registro_alumno = REGISTRO
      DNI: entero;
      Nombre: cadena;
      Apellidos: cadena;
Fin Registro_alumno;
VARIABLES
      Alumno: Registro_alumno
```

Las operaciones básicas que se pueden realizar con registros son:

- **Asignación de registros**

  Consiste en asignar un registro completo a una variable de tipo registro, definida con los mismos campos y en el mismo orden.

- **Lectura/Escritura**

  Se puede realizar, al igual que en arrays, la lectura o escritura de un campo. Para ello se debe saber el nombre del registro y el nombre del campo. Ejemplo: `Alumno.Nombre="XXXXXX"`

A. Características de los registros

Todos los registros deben tener un campo que los diferencie del resto. Dicho campo denominado clave, servirá para identificar unos registros de otros. La información contenida en los registros puede ser de contenido fijo o variable:

- ***Información fija***: Es aquella que no cambia independientemente del proceso aplicado al fichero. Por ejemplo: el campo clave.
- ***Información variable***: Es aquella que puede cambiar dependiendo del proceso aplicado al fichero.

B. Diseño de registros

Los registros pueden ser de longitud fija, variable o indefinida:

a) ***Registros de longitud fija***: Son aquellos en los que la longitud es la misma para todos los registros.

   Tipos de diseño:

   1. Con igual número de campos e idéntica longitud de cada campo en cada registro.
   2. Con igual número de campos y distinta longitud de cada campo en cada registro.
   3. Con distinto número de campos en cada registro y distinta longitud de cada campo en cada registro.

b) ***Registros de longitud variable***: La variabilidad de la longitud del registro radicará en la mayor o menor ocupación del espacio total destinado a almacenar el registro. La ocupación total siempre oscilará entre un máximo y un mínimo.

c) ***Registros de longitud indefinida***: Son aquellos cuya longitud y estructura son totalmente variables. La ocupación del soporte es óptima, ya que cada registro ocupa exclusivamente lo que tiene. Se pueden diseñar mediante:

   - Separadores de campos o banderas.
   - Indicadores de longitud.
   - Máscaras.

3.4. CONJUNTOS

Un conjunto es una estructura de datos formada por una colección de elementos, sin ningún orden concreto ni valores repetidos. Es decir, a diferencia de los arrays, un conjunto no puede tener elementos duplicados.

Las principales operaciones sobre conjuntos son:

- **Unión**: la unión de dos conjuntos A y B es el conjunto A ∪ B que contiene todos los elementos de A y de B
- **Intersección**: la intersección de dos conjuntos A y B es el conjunto A ∩ B que contiene todos los elementos comunes de A y B
- **Diferencia**: la diferencia de dos conjuntos A y B es el conjunto A – B que son todos aquellos elementos de A que no están en B.
- **Pertenencia:** permite comprobar si un elemento forma parte de un conjunto o no.

4. RECURSOS Y HERRAMIENTAS EDUCATIVAS DE INTERÉS

Como herramienta educativa de interés relacionada con este tema podemos citar a **Sololearn**, que se trata de una aplicación web para iOS y Android con una amplia comunidad de usuarios para el aprendizaje de lenguajes de programación, estructuras de datos y fundamentos de algoritmia. Fue galardonada por la FbStart (Facebook) como mejor aplicación del año 2017.

5. CONCLUSIÓN

La elección de la estructura de datos idónea dependerá de la naturaleza del problema a resolver y del lenguaje de programación a utilizar. En general, cuando la cantidad de datos a manejar sea pequeña o de un tamaño definido, resultará beneficioso utilizar las estructuras estáticas. Elegir la estructura de datos correcta es clave para diseñar un algoritmo eficiente.

6. BIBLIOGRAFÍA

- Joyanes Aguilar, L. **Fundamentos de programación. Algoritmos, estructuras de datos y objetos**. Editorial McGraw-Hill
- Prieto A., Lloris A. y Torres J.C. **Introducción a la informática**. Editorial McGraw-Hill
- De Miguel Anasagasti, P. Fundamentos de los computadores. Editorial Paraninfo
- Langsam, Augenstein y Tenenbaum: **Estructuras de datos en C y C++.** Editorial Prentice Hall.
- www.xataka.com (Web sobre actualidad TIC e informática)
- www.sololearn.com
- https://visualgo.net/es