

PROGRAMACIÓN ESTRUCTURADA

UNIDAD N° 6

REGISTROS

Contenido

| | |
|---|---|
| Conceptos a tener en cuenta | 1 |
| Registros..... | 1 |
| Definición: | 1 |
| Registros en Programación | 1 |
| Declaración de Registros..... | 2 |
| Acceso a los campos de un registro | 3 |
| Anidamiento de registros..... | 3 |
| Operaciones sobre registros | 3 |
| Arreglos de Registros | 4 |
| Ejercicios resueltos..... | 4 |
| Descubriendo cómo guardar mediante Registros la información cotidiana | 4 |
| Definición del Arreglo de Registros de Recetas | 6 |
| Actividad | 6 |
| Condiciones de la entrega..... | 6 |
| Bibliografía | 6 |

Conceptos a tener en cuenta

Registros

Definición: Un registro es una estructura de datos compuesta que permite agrupar datos de diferentes clases (reales, lógicos, caracteres, etc.) que tienen alguna conexión lógica en una única estructura. En otras palabras, un registro es un conjunto de valores, con tres características básicas:

- Los valores pueden ser de distinto tipo, un registro es una estructura heterogénea.
- Los valores almacenados en un registro se denominan campos, y cada uno de ellos tiene un identificador; los campos son nombrados individualmente, como variables ordinarias.
- El almacenamiento (memoria) ocupado por un registro es fijo; por esto, un registro es una estructura estática.

Registros en Programación

Los registros en programación nos permiten representar las entidades del mundo real teniendo en cuenta sus características, es decir se usan para definir un conjunto de datos relacionados como una única estructura.

Un registro es un tipo de dato creado por el usuario, es necesario definirlo antes de poder utilizarlo. Luego de crear el dato, podemos usarlo para definir las variables de tipo registro.

**EMPLEADO**

Nombre
Fecha de Nac.
Cargo
Salario
...

PRODUCTO

Código
Descripción
Precio
Stock
...

**Libro**

Código
Título
Autor
Edición
...

Préstamo

Código
DNI
Fecha préstamo
Fecha entrega
...

Proceso de abstracción “¿Cómo representar mediante registros las entidades del mundo real?”

Declaración de Registros

Un registro se declara identificando al tipo como un registro (`t_registro=registro`) y luego especificando el nombre y tipo de los campos individuales (`campo_1: tipo_dato`). Esta lista de campos sigue las reglas generales de declaración de variables, y se encuentra entre las palabras reservadas *registro* y *fin_registro*. Los campos pueden ser de cualquier tipo, datos simples o estructurados (arreglos, registros).

El formato general de declaración de registros es:

```
TIPOS
t_registro=registro
    campo_1: tipo_dato
    campo_2: tipo_dato
    ...
    campo_n: tipo_dato
fin_registro
VARIABLES
    nombre_variable: t_registro
```

Obsérvese que una vez **declarado el tipo registro**, se pueden definir variables de ese tipo para utilizarlas en el programa. El siguiente ejemplo ilustra la declaración de un registro que almacena la información acerca de un producto.

```
TIPOS
t_producto=registro
    cod_producto: entero
    marca_producto: cadena
    precio_producto: real
    descripción_producto: cadena
fin_registro
VARIABLES
    mercadería: t_producto
```

Acceso a los campos de un registro

Para acceder a los campos de un registro es necesario especificar tanto el nombre de la variable de tipo registro como el del campo que se desea referenciar. Esto se denomina calificar el campo. Por ejemplo, si se quiere almacenar un valor en el campo *precio_producto* de la variable *mercaderia* (ejemplo anterior) se procede como sigue:

```
mercaderia.precio_producto ← 12.36
```

Puede observarse que entre el **registro *mercaderia*** y el **campo *precio_producto*** aparece un punto. Este símbolo se denomina designador o selector de campo.

Anidamiento de registros

Las variables estructuradas, como los registros, pueden estar anidadas (una dentro de otra). Es decir, un campo de un registro puede, a su vez, ser otro registro. Un registro con uno o más campos de tipo registro se llama registro jerárquico o anidado.

El siguiente ejemplo ilustra un anidamiento de registro de 2 niveles:



Anidamiento

TIPOS

```
t_fecha=registro
    día: entero
    mes: entero
    año: entero
    fin_registro

t_persona=registro
    legajo: entero
    nombre: cadena
    f_nacimiento: t_fecha
    fin_registro
```

VARIABLES

```
empleado: t_persona
```

En la declaración precedente se especifican los registros *t_fecha* y *t_persona*. Nótese que en la declaración de *t_persona* el campo *f_nacimiento* es de tipo *t_fecha*, es decir, que este campo es también un registro.

La manera de acceder a los campos esencialmente no cambia, sin embargo, es necesario utilizar doble calificación para referenciar los campos *día*, *mes* y *año*. Por ejemplo, si se quiere almacenar un valor en el campo *día* del registro *f_nacimiento*, que es campo de la variable *empleado* se procede como sigue:

```
empleado.f_nacimiento.día ← 28
```

Observe que se destacaron en negritas y cursiva (negritas para el primer nivel y cursiva para el segundo) los dos niveles de registro presentes en esta definición. Los siguientes son ejemplos de especificaciones INCORRECTAS de esta misma jerarquía (en negritas se indican los errores):

```
t_persona.f_nacimiento.día ← 28
empleado.t_fecha.día ← 28
t_persona.t_fecha.día ← 28
```

Operaciones sobre registros

Dado que los campos de un registro son variables de algún tipo de dato, las operaciones posibles sobre un campo son las permitidas para el tipo de dato correspondiente.

Además de las operaciones sobre cada campo, existe una que puede realizarse sobre un registro completo, la *ASIGNACIÓN*. Esto es posible si las variables utilizadas en la operación son del mismo tipo de registro. Por ejemplo, si la variable *vendedor* y la variable *empleado* son del tipo *t_persona*, la siguiente operación de asignación es válida.

```
vendedor ← empleado
```

En este caso, el valor de cada campo de *empleado* se copia en cada campo de *vendedor*.

No pueden realizarse comparaciones entre registros completos, es decir, que dos variables del mismo tipo de registro no pueden ser comparadas utilizando los operadores relacionales. Para determinar si dos registros son iguales es necesario realizar la comparación campo por campo.

Sobre las variables de tipo registro no se pueden aplicar directamente las operaciones *LEER* y *ESCRIBIR*; éstas deben ejecutarse sobre campos individuales. Por ejemplo, NO ES CORRECTA la sentencia

```
ESCRIBIR vendedor
```

pero si lo es la sentencia

```
ESCRIBIR vendedor.nombre
```

Arreglos de Registros

En la práctica no es tan común el uso de registros simples. En general, los registros se agrupan en conjuntos conocidos como arreglos de registro. Por ejemplo, la siguiente declaración permite representar 100 productos:

```
CONSTANTES
MAXPROD=100
TIPOS
t_producto=registro
    cod_producto: entero
    marca_producto: cadena
    precio_producto: real
    descripción_producto: cadena
fin_registro

t_stock=arreglo [1..MAXPROD] de t_producto
VARIABLES
    inventario: t_stock
```

| Posición 1 | Posición 2 | Posición 3 | ... | Posición 99 | Posición 100 |
|----------------------|----------------------|----------------------|-----|----------------------|----------------------|
| cod_producto | cod_producto | cod_producto | | cod_producto | cod_producto |
| marca_producto | marca_producto | marca_producto | | marca_producto | marca_producto |
| precio_producto | precio_producto | precio_producto | ... | precio_producto | precio_producto |
| descripción_producto | descripción_producto | descripción_producto | | descripción_producto | descripción_producto |

inventario (variable de tipo t_stock)

Observe que cada una de las posiciones del arreglo *inventario* es un registro de tipo *t_producto*. En este caso si se quiere asignar un valor al campo *precio_producto* del registro que ocupa la posición 3 del arreglo *inventario* se procede como sigue:

```
inventario[3].precio_producto ← 69.50
```

Todas las operaciones (asignación, lectura/escritura, recorrido, actualización, ordenación, búsqueda, intercalación) vistas para arreglos son aplicables (con ligeras modificaciones) a arreglos de registros.

Ejercicios resueltos

Descubriendo cómo guardar mediante Registros la información cotidiana

La definición de registros en programación permite representar entidades del mundo real en soluciones basadas en computadora. Así, la información acerca de los empleados de una empresa (nombre, fecha de nacimiento, cargo, salario, etc.) pueden ser almacenados en una única estructura que permita manejar como un todo un conjunto de datos relacionados.

Veamos algunos ejemplos de información que puede ser almacenada en registros:

Si nos gusta cocinar podríamos tener guardadas las diferentes recetas para agilizar nuestras compras, además para hacer aún más eficientes nuestras decisiones en las compras, guardaríamos un registro de toda esta información en un libro de Excel, que sea diseñado por nosotros y almacene todos los ingredientes necesarios para nuestras recetas favoritas. Entonces al ir a realizar las compras para abastecernos sería mucho más fácil tener en nuestro celular todos los datos necesarios.

Algunos ejemplos de datos de diferentes recetas:

Dulce de leche

Ingredientes

960 grs.

3 litros leche entera

660 grs azúcar

3 grs bicarbonato. (1 cdita)



1 kg. harina 0000, 3 cdas aceite de maíz,
1 cda. manteca blanda (le da frescura a la
masa) 25 gs. Levadura, 1 cucharadita de
azúcar, Sal 4 cditas. de sal(o a gusto)



Ejemplos de recetas

| Ingrediente | Cantidad | Unidad |
|--------------|----------|-----------|
| Leche entera | 3 | l |
| Azúcar | 660 | g |
| Bicarbonato | 1 | cucharada |

Elementos de la receta para Dulce de Leche

| Ingrediente | Cantidad | Unidad |
|-------------|----------|-----------|
| harina 0000 | 1 | kg |
| manteca | 1 | cucharada |
| levadura | 25 | g |
| sal | 4 | cucharada |
| aceite | 3 | cucharada |
| Azúcar | 1 | cucharada |

Elementos de la receta para 5 Baguettes

Si lo que deseo es guardar muchas recetas y tener la lista de ingredientes de forma más rápida, puedo almacenar todas las recetas en una sola planilla, por lo que agregaría una columna con el nombre de la receta, con esta información completaría mi arreglo de registro recetas.

| # | Receta | Ingrediente | Cantidad | Unidad |
|---|----------------|--------------|----------|-----------|
| 1 | Dulce de leche | Leche entera | 3 | l |
| 2 | Dulce de leche | Azucar | 660 | g |
| 3 | Dulce de leche | Bicarbonato | 1 | cucharada |
| 4 | Baggetes | harina 0000 | 1 | kg |
| 5 | Baggetes | manteca | 1 | cucharada |
| 6 | Baggetes | levadura | 25 | g |
| 7 | Baggetes | sal | 4 | cucharada |
| 8 | Baggetes | aceite | 3 | cucharada |
| 9 | Baggetes | Azucar | 1 | cucharada |

Arreglo de Registros Recetas

Definición del Arreglo de Registros de Recetas

Por ejemplo, la siguiente declaración permite representar 100 recetas:

| PSEUDOCÓDIGO | C++ |
|--|------------------------------------|
| CONSTANTES | ... |
| MAX=100 | const int MAX=100; |
| TIPOS | ... |
| t_receta=registro | typedef char tcad[30]; |
| receta: cadena | typedef struct t_receta { |
| ingrediente: cadena | tcad receta; |
| cantidad: real | tcad ingredientes; |
| unidad: cadena | float cantidad; |
| fin_registro | tcad unidad; |
| | }; |
| t_recetazas=arreglo [1..MAX] de t_receta | typedef t_receta t_recetazas[MAX]; |
| VARIABLES | ... |
| misrecetas: t_recetazas | main() |
| | { t_recetazas misrecetas; |
| | ... |
| | } |

Actividad

- 1- Busca ejemplos de entidades (objetos, personas, funciones, etc.) que puedan ser representados por registros y arreglos de registros.
- 2- Arma un libro de Excel u otro software de manejo de planillas con datos de ejemplos de al menos 5 registros individuales y un arreglo de registros.
- 3- Fundamenta tu elección.
- 4- Investiga y menciona qué operaciones sobre vectores se asemejan a funciones de Excel que pueden ser aplicadas al arreglo de registros, describe su uso paso a paso.
- 5- Realiza la definición del registro y del arreglo de registros en Pseudocódigo y C++.
- 6- Arma un documento en un archivo de texto y conviértelo a PDF.
- 7- Coloca en la carátula el nombre de la materia, nombre y apellido del/los estudiantes/s que realizaron el trabajo.
- 8- Antes de subir revisa las intervenciones realizadas por tus compañeros00 que tu idea es original, es decir que no se repita en intervenciones anteriores.
- 9- Sube el archivo en el Foro “Descubriendo cómo guardar mediante Registros la información cotidiana”.
- 10- Ante cualquier duda escribir la consulta en el Foro “Consultas acerca de Registros”.

Condiciones de la entrega

La actividad es asincrónica y puede ser realizada de a pares. Debe ser subida hasta el día lunes 19/10/2020 a hs. 10. Se valorará la entrega del trabajo en tiempo y forma, el manejo de los conceptos referidos a registros, la claridad en la expresión, la originalidad y la prolijidad en la presentación.

Bibliografía

- Sznajdleder, Pablo Augusto. Algoritmos a fondo. Alfaomega. 2012.
- López Román, Leobardo. Programación estructurada y orientada a objetos. Alfaomega. 2011.
- De Giusti et al. Algoritmos, datos y programas, conceptos básicos. Editorial Exacta, 1998.
- Joyanes Aguilar, Luis. Fundamentos de Programación. Mc Graw Hill. 1996.
- Joyanes Aguilar, Luis. Programación en Turbo Pascal. Mc Graw Hill. 1990.