

Estructuras de Registro

```
vector<int> v;  
v.push_back(1);  
v.push_back(2);  
v.push_back(3);  
v.push_back(4);
```

```
for(int i = 0; i < v.size(); i++)  
    cout << v[i] << " ";  
cout << endl;  
// Output: 1 2 3 4  
// Size of the vector is 4  
// Accessing elements using [] operator  
// Accessing elements using at() method  
// Accessing elements using iterator
```

Accessing elements using iterator

Accessing elements using iterator

Accessing elements using iterator

Accessing elements using iterator

Accessing elements using iterator

Accessing elements using iterator

Accessing elements using iterator

Accessing elements using iterator

Accessing elements using iterator

Accessing elements using iterator

Supongamos que queremos hacer una agenda con los números de teléfono de nuestros amigos.

Necesitaríamos un array de cadenas para almacenar sus nombres, otro para sus apellidos y otro para sus números de teléfono.

Esto puede hacer que el programa quede desordenado y difícil de seguir. Y aquí es donde presenta ventajas el uso de las estructuras.

Estructura de Registro

Una estructura o registro es una estructura de datos que agrupa variables que pueden tener tipos diferentes.

Es decir, en una estructura en la que se puede definir datos o valores de diferentes tipos.

Cada componente de un registro se conoce como campo o miembro.

Definición de la Estructura

En pseudocódigo, un registro se declara de la siguiente forma:

Algoritmo declaracion_registro

tipo

***registro:** nombre_registro*

<tipo de dato>: <nombre de variable, nombre de variable, etc.>

<tipo de dato>: <nombre de variable, nombre de variable, etc.>

.

.

fin-registro

var

nombre_registro: nombre_variable

Los componentes de la estructura se declaran dentro de ella, cada uno con su tipo y nombre de variable, los tipos pueden ser diferentes.

El nombre (tiponuevo) de la estructura se considera como un tipo, no una variable.

Operaciones con Arreglos de Estructuras

- Cargar un arreglo de estructuras.
- Recorrer un arreglo de estructuras.
- Buscar un elemento en particular de estructuras.
- Acceder a un elemento en una posición determinada.
- Acceder a una posición determinada y mostrar su contenido.
- Insertar un nuevo elemento.
- Eliminar un elemento.
- Ordenar un arreglo de estructuras.

Acción Agenda es

Ambiente

tipo

registro :amigo

id:entero;

nombre[20]:caracter;

apellidp[20]:caracter;

dir[20]: caracter;

fin registro;

a: Arreglo [5] de amigo;

da:a;

i:entero;

nom[20]:caracter;

apel[20]:caracter;

dire[20]: caracter;

Algoritmo

Escribir (“Ingrese los datos de los amigos”);

para(i=0; i<5 ; i++)

da[i].id:i;

escribir(" Datos del amigo número:", i+1);

escribir("Nombre: ");

leer(nom);

da[i].nombre:=nombre;

escribir("Apellido: ");

leer(apel);

da[i].apellido:=apel;

escribir(“Dirección: ");

leer(dire);

da[i].dir:=dire;

fin para

```
escribir (“Mostrarlos datos de los amigos”);  
escribir (“*****Agenda de amigos*****”);  
escribir (“ Nombre           Apellido           Dirección”);  
para( i=0; i<5 ; i++ )  
    escribir (da[i].nombre)  
    escribir(da[i].apellido);  
    escribir (da[i].dir);  
fin para  
  
fin acción
```


Acción Cuentas es

Ambiente

tipo

registro :cuenta

 nro-cuenta:entero;

 nombre[20]:caracter;

 saldo: real;

fin registro;

a: Arreglo [5] de cuenta;

da:a;

nc:entero;

nom[20]:caracter;

s,deudor,acreedor:real;

i:entero;

Otro ejemplo

Algoritmo

deudor:=0;

acreedor:=0;

Escribir (“Ingrese los datos de los clientes”);

para(i=0; i<5 ; i++)

 escribir(" Nro de cuenta");

 leer(nc);

 da[i].nro-cuenta:=nc;

 escribir("Nombre del cliente: ");

 leer(nom);

 da[i].nombre:=nom;

 escribir(“Saldo: ");

 leer(s);

 da[i].saldo:=s;

fin para

Escribir (“Calcular el total de saldo deudor y total de saldo acreedor);

para(i=0; i<5 ; i++)

 si da[i].saldo < 0 entonces deudor:=deudor + da[i].saldo;

 sino acreedor:= acreedor + da[i].saldo ;

fin para

escribir (“Mostrarlos datos de los clientes”);

escribir (“*****Listado de Clientes*****”);

escribir (“ Nro.Cuenta Nombre Saldo”);

 para(i=0; i<5 ; i++)

 escribir (da[i].nro.-cuenta)

 escribir(da[i].nombre);

 escribir (da[i].saldo);

 fin para

Escribir ("Total saldo deudor:" ,deudor);

Escribir ("Total saldo Acreedor:" ,acreedor);

Fin acción