CADENAS – ARREGLOS - ESTRUCTURAS

CLASE 2

ESTRUCTURAS

ESTRUCTURAS

- Las estructuras permiten agrupar varios datos, que mantienen algún tipo de relación, aunque sean de distinto tipo, permitiendo manipularlos todos juntos, usando un mismo identificador, o cada uno por separado.
- Las estructuras también son llamadas *registros*.

struct

- En C++ las estructuras se declaran con la palabra reservada *struct*
- Un *struct* es un tipo de datos complejo conformado por un conjunto de campos de otros tipos (*básicos o complejos*) asociados a un identificador.

struct

- Sintaxis:

```
struct identificador {
        tipo nombre_campo1;
        ….
        tipo nombre_campoN;
} ;
```

Lista de campos

struct

- Una vez definida la estructura se pueden declarar variables, parámetros, funciones, etc. de ese tipo.

- En C++ la palabra *struct* es opcional en la declaración de objetos, al contrario de lo que sucede en C, en el que es obligatorio usarla.

- El acceso a los campos de una estructura se hace mediante el punto (.)

ESTRUCTURAS

Definición

```
struct alumno {
    long legajo;
    char ape[30],nom[30];
    char genero;
};
```

Declaración

```
alumno alu;
```

Uso

```
cout<< alu.legajo;
```

Estructuras. Operaciones

- **Lectura/Escritura**
  - Campo por campo

- **Asignación**
  - Registro a registro
  - Campo a campo

- **Lectura/Escritura**

```
cin>>alu.legajo;
cout<<alu.legajo;
```

- **Asignación**

```
alumno clase;//declaración
clase=alu;
alu.legajo=1080;
clase.legajo=alu.legajo
```

Arreglos y Registros

- Arreglos de registros
- Registros de arreglos
- Registros de registros

Arreglos de estructuras

Definición

```
struct alumno {
    long legajo;
    char ape[30],nom[30];
    char genero;
};
```

Declaración

```
alumno alu[30];
```

Uso

```
cout<< alu[0].legajo;
```

Práctica #1

- Un empleador desea llevar el registro de sus empleados. Los datos a considerar por cada empleado son: Legajo, Nro de documento, Apellido y Nombre. Desarrollar la carga de empleados y la muestra de datos mediante funciones.
- Mediante un menú el usuario podrá elegir entre
  - Cargar datos
  - Muestra de datos de un empleado en particular
  - Muestra de todos los empleados
  - Salir del programa

Arreglos y Estructuras

- En las combinaciones
  - Estructuras de arreglos
  - Estructuras de estructuras
- Existirán algunos campos que serán de tipo *struct* y/o de tipo arreglo.

Estructuras de arreglos

Definición

```
struct alumno {
    long legajo;
    char ape[30],nom[30];
    char genero;
    float promedioAnual[5];
};
```

Declaración

```
alumno alu[30];
```

Uso

```
alu[0].promedioAnual[1]=0.0;
```

Estructuras de estructuras

Definición

```
struct fecha{
    short día,mes,anio;
}

struct alumno {
    long legajo;
    char ape[30],nom[30];
    char genero;
    fecha fechaIngreso;
};
```

Declaración

```
alumno alu[30],dato;
```

Uso

```
cout<< alu[0].fechaIngreso.dia;
cout<<dato.fechaingreso.día;
```

Práctica #2

- Teniendo en cuenta la Práctica #1, modificar la estructura tal que, además, se pueda registrar por cada empleado: CUIL, fecha de Ingreso, inasistencias por mes.

Práctica #3

- Realizar un programa que gestione los datos de stock de un Kiosco, la información a recoger será: nombre del producto, precio, cantidad en stock. El kiosco dispone de 10 productos distintos. El programa debe ser capaz de:

- Dar de alta un producto nuevo.
- Buscar un producto por su nombre.
- Modificar el stock y precio de un producto dado.

Práctica #4

- Realice un programa que pueda almacenar al menos 20 registros de personas con los campos: apellido, nombre, dirección, teléfono, edad.
- Deberá aparecer un menú que permita:
  - Registrar una persona
  - Mostrar la lista de todos los nombres
  - Mostrar las personas de cierta edad
  - Mostrar las personas cuya inicial sea la que el usuario indique
  - Salir del programa.

Test

- https://es.educaplay.com/recursos-educativos/16408475-arreglos.html