

Almacenamiento de Objetos en estructuras de datos en C++

Definición del concepto (1)

Al igual que se utilizan arreglos para almacenar múltiples valores de tipos de datos primitivos como *int*, *long*, *double*, *char* o *bool*, también es posible usar una estructura de datos básica como un arreglo o matriz, o estructuras disponibles en la colección de datos del lenguaje como la clase [array](#), [vector](#) o [list](#), entre otras, para almacenar objetos de clase.

Para + info acerca de la colección de datos Standard Template Library (STL) de C++ visitar los enlaces:

- <https://www.cplusplus.com/reference/stl/>
- <https://www.geeksforgeeks.org/the-c-standard-template-library-stl/?ref=lbp>

Definición del concepto (2)

En primera instancia vamos a comenzar utilizando un arreglo (básico o primitivo) para almacenar objetos de clase.

Es decir, si un arreglo de tipo *int* almacena valores enteros, un arreglo de tipo *Estudiante* almacenara objetos de la clase *Estudiante*.

Para el ultimo caso, se dice entonces que el arreglo almacena referencias de memoria de objetos de clase.

Almacenar objetos en un arreglo

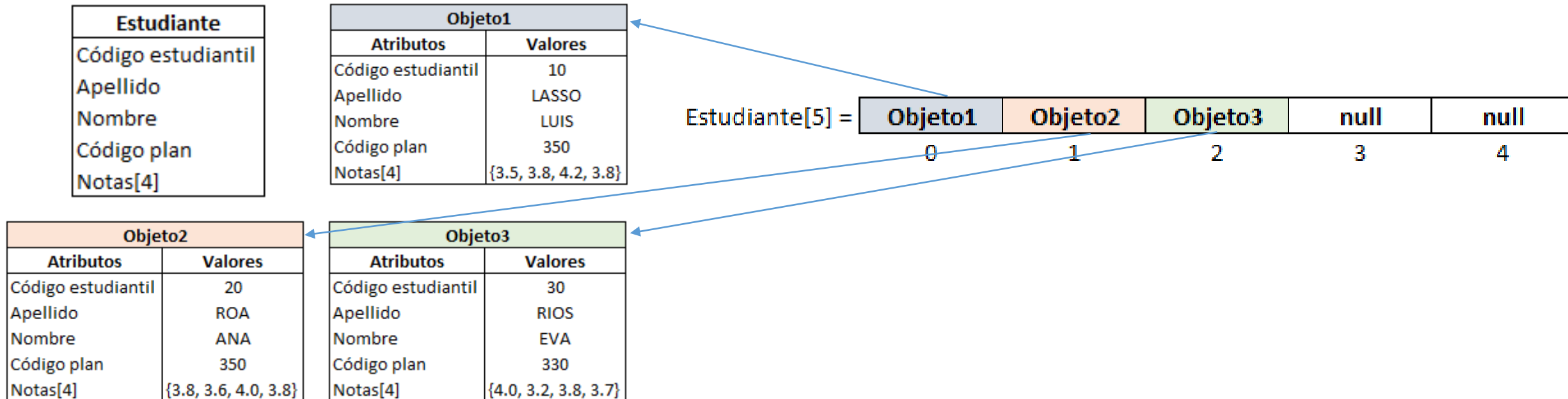
Representación grafica

La representación grafica de un arreglo de 5 posiciones de tipo *int* seria así:

`int arreglo[5] =`

| | | | | |
|----|---|----|----|-----|
| 10 | 5 | 78 | 56 | -19 |
| 0 | 1 | 2 | 3 | 4 |

Ahora, la representación grafica de un arreglo 5 posiciones de tipo *Estudiante* seria así:



Declaración de un arreglo de objetos

La sintaxis para declarar un arreglo de objetos en C++ es la siguiente:

NombreClase nombreArreglo[tamaño];

Siguiendo con el ejemplo de la clase *Estudiante*, la declaración de un arreglo que almacene 5 objetos sería así:

Estudiante arregloEstudiantes[5];

Cabe anotar que en un arreglo de tipo clase, todas sus posiciones almacenan de forma predeterminada *null*.

| | | | | | |
|-------------------------|------|------|------|------|------|
| arregloEstudiantes[5] = | null | null | null | null | null |
| | 0 | 1 | 2 | 3 | 4 |

Asignación en un arreglo de objetos (1)

Para asignar (almacenar) objetos en un arreglo de tipo clase, se deben seguir los siguientes pasos:

- 1) Declarar el arreglo de objetos de la clase
- 2) Por medio del índice de arreglo, dar valores a los atributos del objeto

Continuando con el ejemplo del arreglo de tipo *Estudiante* la asignación de un objeto seria así:

- 1) *Estudiante* arregloEstudiantes[5];
- 2) arregloEstudiantes[0].setCodEstudiantil();
arregloEstudiantes[0].setApellido();
arregloEstudiantes[0].setNombre();
arregloEstudiantes[0].setCodPlan();
arregloEstudiantes[0].setNotas();

Almacenar en el índice 0, los datos del primer objeto de la clase Estudiante. En este caso, no se conoce el nombre del objeto, solo se sabe su ubicación dentro del arreglo.

Asignación en un arreglo de objetos (2)

Generalizando el proceso anterior, podemos usar una estructura repetitiva (*for* o *while*) para asignar (almacenar) *n* objetos en un arreglo de tipo clase, así:

Estudiante arregloEstudiantes[5];

```
for(int i = 0; i < 5; i++){  
    arregloEstudiantes[i].setCodEstudiantil();  
    arregloEstudiantes[i].setApellido();  
    arregloEstudiantes[i].setNombre();  
    arregloEstudiantes[i].setCodPlan();  
    arregloEstudiantes[i].setNotas();  
}
```

Imprimir un arreglo de objetos

Igual como el proceso de asignación (almacenar), para recorrer un arreglo de *n* objetos generalmente se usa una estructura repetitiva, y se llaman a los métodos **get** de la clase, así:

```
for(int i = 0; i < 5; i++){
    cout<<"Datos del Estudiante # "<<(i+1)<<endl;
    cout<<"Cod Estudiantil:"<< arregloEstudiantes[i].getCodEstudiantil()<<endl;
    cout<<"Apellido:"<< arregloEstudiantes[i].getApellido()<<endl;
    cout<<"Nombre:"<< arregloEstudiantes[i].getNombre()<<endl;
    cout<<"Cod Plan:"<< arregloEstudiantes[i].getCodPlan()<<endl;
    for(int n = 0; n < 3; n++){//recorrer el arreglo de las notas, solo hasta la pos 2. La pos 3 es la definitiva
        cout<<"Nota Parcial "<< (n+1)<<": "<< arregloEstudiantes[i].getNota(n+1)<<endl;
    }
    cout<<"Definitiva: "<< arregloEstudiantes[i].getNota(3)<<endl<<endl;
}
```