

Ejemplo de clase

1. Ejemplo de clase: La clase Disco
2. Ejemplo de clase: La clase Colección
3. Relación entre clases: Clientismo

1

Ejemplo de clase

Una clase para modelar el tipo de dato disco

Vamos a modelar con una clase, un nuevo tipo de datos , donde los elementos de la clase, los objetos, son discos. El nombre de la clase que vamos a modelar es *Disco*.

- ✓ La información asociada a cada uno de los discos:
Titulo, nº de canciones, precio, fecha de compra

<i>Disco</i>
<i>titulo</i>
<i>num_canciones</i>
<i>precio</i>
<i>fecha_compra</i>

```
#include <iostream.h>
class Disco
{
private:
    char titulo [30];
    int num_canciones;
    float precio;
    Date fecha_compra;
};
```

struct Date

```
{
    int dia;
    int mes;
    int año;
};
```

2

Ejemplo de clase

Las operaciones necesarias para la gestión de discos:

Por ejemplo, necesitaremos un método que se encargue de la inicialización de los objetos de la clase *Disco*.



Dos métodos con el mismo nombre.
Son dos funciones distintas:

Sobrecarga

class Disco

```
{ private:
    char titulo [30];
    int num_canciones
    float precio;
    Date fecha_compra;
public:
    void inicializar (char * tit);
    void inicializar (char * tit, float price);
};
```

```
void Disco :: inicializar (char * tit)
{ strcpy(titulo, tit);
  num_canciones = precio = 0; }
```

```
void Disco :: inicializar (char * tit , float pr )
{  strcpy(titulo, tit);
   precio = pr ;
   num_canciones = 0;
}
```

3

Ejemplo de clase

Funciones (suplemento)

Polimorfismo

C++ permite realizar **sobrecarga de funciones**.

- Esto significa que puedan definirse varias funciones con el mismo nombre pero con distintos parámetros, y distintas implementaciones.
- Cuando hacemos una llamada a una función sobrecargada, el compilador es capaz de saber a cual de ellas nos estamos refiriendo analizando los parámetros que pasamos a la función.

Proceso denominado **resolución de sobrecarga**

Ejemplo:

```
int area ( int a, int b );
float area (int radio );
int area ( float c, float d, float e );
```

```
void main()
{
    int m;
    m = area( 2 , 3 );
    m = area( 2.3 , 8.3, 7.0 );
}
```

Ejemplo de clase

¿Se nos ocurren más operaciones?



Métodos que devuelven el valor de los atributos

Métodos que modifican el valor de los atributos

5

Ejemplo de clase

¿Se nos ocurren más operaciones?



```
class Disco
{ private:
    char titulo [30];
    int num_canciones
    float precio;
    Date fecha_compra;
public:
    void inicializar (char * tit);
    void inicializar (char * tit, float price);

    char * dev_titulo ( );
    int dev_num_canciones ( );
    float dev_precio ( );
    Date dev_fecha ( );

    void mod_num_canciones( int n);
    void mod_precio ( float p);
    void mod_fecha ( Date f );
};
```

6

Ejemplo de clase

¿Se nos ocurren más operaciones?

```
class Disco
{ private:
    char titulo [30];
    int num_canciones
    float precio;
    Date fecha_compra;
public:
    void inicializar (char * tit);
    void inicializar (char * tit, float price);

    char * dev_titulo ( );
    int dev_num_canciones ( );
    float dev_precio ( );
    Date dev_fecha ( );

    void mod_num_canciones( int n);
    void mod_precio ( float p);
    void mod_fecha ( Date f);
};
```

```
char * Disco :: dev_titulo ( )
{ return titulo; }

int Disco :: dev_num_canciones ( )
{ return num_canciones; }

float Disco :: dev_precio ( )
{ return precio; }

Date Disco :: dev_fecha ( )
{ return fecha_compra; }

void Disco :: mod_num_canciones( int n)
{ num_canciones = n; }

void Disco :: mod_precio ( float p)
{ precio = p; }

void Disco :: mod_fecha ( Date f)
{ fecha_compra = f; }
```

7

Ejemplo de clase

Uso de la clase Disco

```
#include <iostream.h>
class Disco
{ ...
};
...
void main();
{
    Disco cd1 , cd2;
    cd1.inicializar("El piano");
    cd2.inicializar("Dirty work", 12);
    Fecha f;
    f.dia = 2;
    f.mes = 4;
    f.year = 1985;
    cd1.mod_fecha(f);
    cd1.mod_precio(15);
    cd1.mod_num_canciones(20);
    cd2.mod_fecha(f);
    cd2.mod_num_canciones(17);
}
```

cd1

titulo
"El piano"
num_canciones
20
precio
15
Fecha_compra
2.4.1985

cd2

titulo
"Dirty work"
num_canciones
17
precio
12
Fecha_compra
2.4.1985

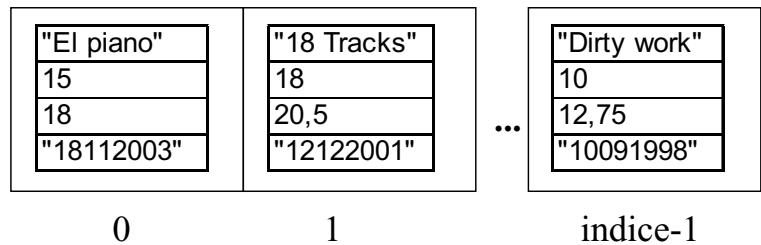
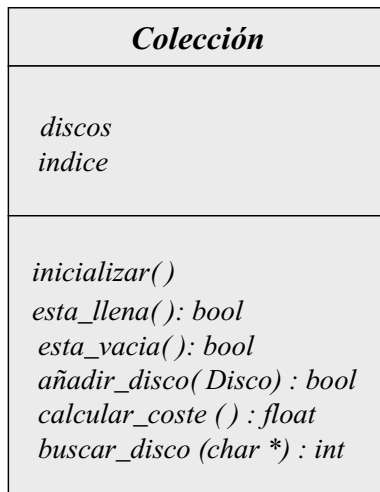
Dos objetos: mismo comportamiento
pero diferente estado

8

Ejemplo de clase

Una clase para modelar una colección de discos

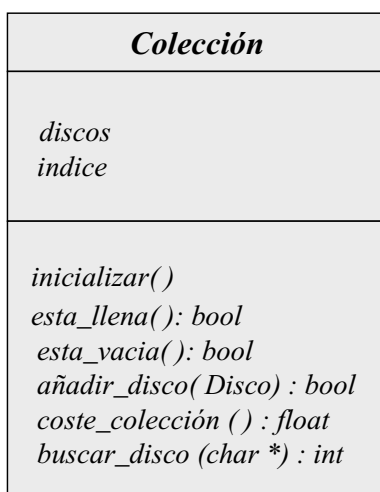
Supongamos que queremos guardar información de todos los discos que tenemos en nuestra casa. Podemos modelar con una nueva clase, una lista de discos, la clase **Colección**.



El atributo **discos** representa el array de discos.
El atributo **indice** representa el número de discos que tiene la colección.

Ejemplo de clase

Una clase para modelar una colección de discos



```
class Coleccion
{
private:
    Disco discos [100];
    int indice;
public:
    void inicializar ( ) ;
    bool esta_llena ( ) ;
    bool esta_vacia ( ) ;

    bool añadir_disco ( ) ;
    float coste_coleccion ( ) ;
    int buscar_disco( char *);
};
```

La clase Colección hace uso de la clase Disco.

La clase Colección es cliente de la clase Disco.

Cuando una clase A hace uso de otra clase B, se dice que la clase A es cliente de la clase B.

Relación de clientismo entre clases

Ejemplo de clase

Relaciones entre clases: Clientismo

Las relaciones entre clases también se representan de forma gráfica.

La relación de clientismo se representa conectando las clases mediante líneas con un rombo en un extremo. El rombo se coloca en la clase que contiene los objetos de la otra:



La clase **Colección** es cliente de la clase **Disco**.

Los objetos de la clase **Colección** contienen objetos de la clase **Disco**.

Ejemplo de clase

Una clase para modelar una colección de discos

```
class Coleccion
{
    private:
        Disco discos [100];
        int indice;
    public:
        void inicializar ( );
        bool esta_llena ( );
        bool esta_vacia ( );

        bool añadir_disco ( Disco );
        float coste_coleccion ( );
        int buscar_disco( char *);
};
```

```
void Coleccion :: inicializar ( )
{ indice = 0 ; }

bool Coleccion :: esta_llena ( )
{ if (indice == 100 ) return true;
  else return false;
}

bool Coleccion :: esta_vacia ( )
{ if (indice == 0 ) return true;
  else return false;
}

bool Coleccion :: añadir_disco (Disco cd )
{ if (!(esta_llena( ) ) )
  {
    discos[indice] = cd;
    indice++;
    return true;
  }
  else return false;
}
```

Una clase para modelar una colección de discos

```
class Coleccion
{
private:
    Disco discos [100];
    int indice;
public:
    void inicializar ( ) ;
    bool esta_llena ( ) ;
    bool esta_vacia ( ) ;

    bool añadir_disco ( Disco );
    float coste_coleccion ( ) ;
    int buscar_disco( char * );
};
```

```
float Coleccion :: coste_coleccion ( )
{
    float cost = 0;
    for (int i=0; i<indice; i++)
    {
        cost = cost + discos[i].dev_precio();
    }
    return cost;
}
int Coleccion :: buscar_disco( char * tit)
{
    int i = 0;
    while (i<indice )
    {
        if (strcmp(tit, discos[i].dev_titulo())==0)
            return i;
        else
            i++;
    }
    return -1;
}
```