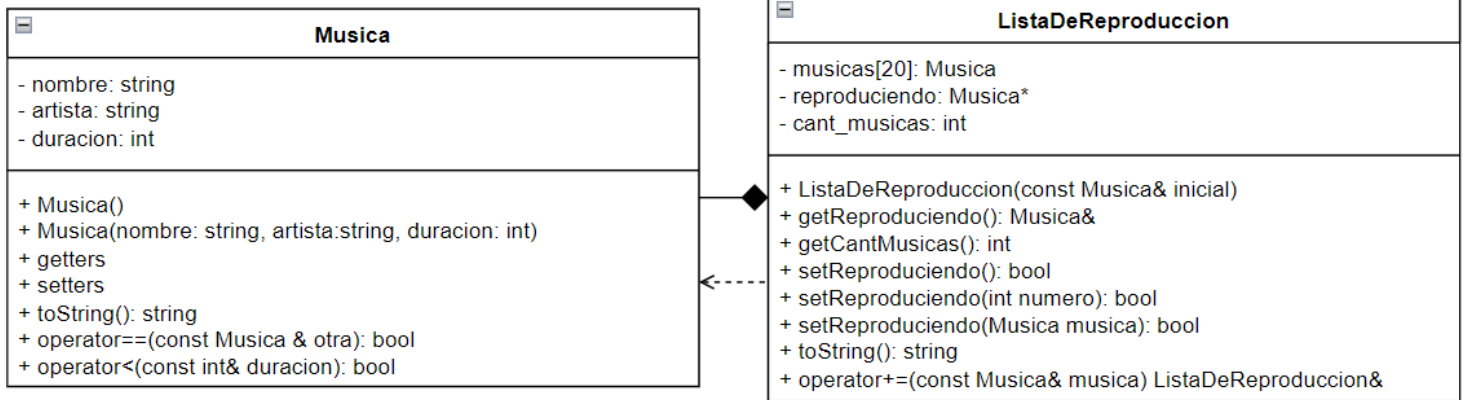


Ejercicios Tema 5. Operadores y friends

EJERCICIO 1 . Crear las clases (.h y .cpp) Musica y ListaDeReproduccion según el siguiente diagrama de clases:



Canción

- Tiene 2 constructores, 3 funciones de get y 3 de set.
- La función `toString` muestra el nombre el artista y la duración en segundos.
- El operador de igualdad verifica si tanto el nombre como la duración de 2 canciones son datos iguales.
- El operador de menor que verifica si la canción tiene una duración (atributo) menor a duración (parámetro en segundos).
- **ListaDeReproduccion** es amiga de Canción.

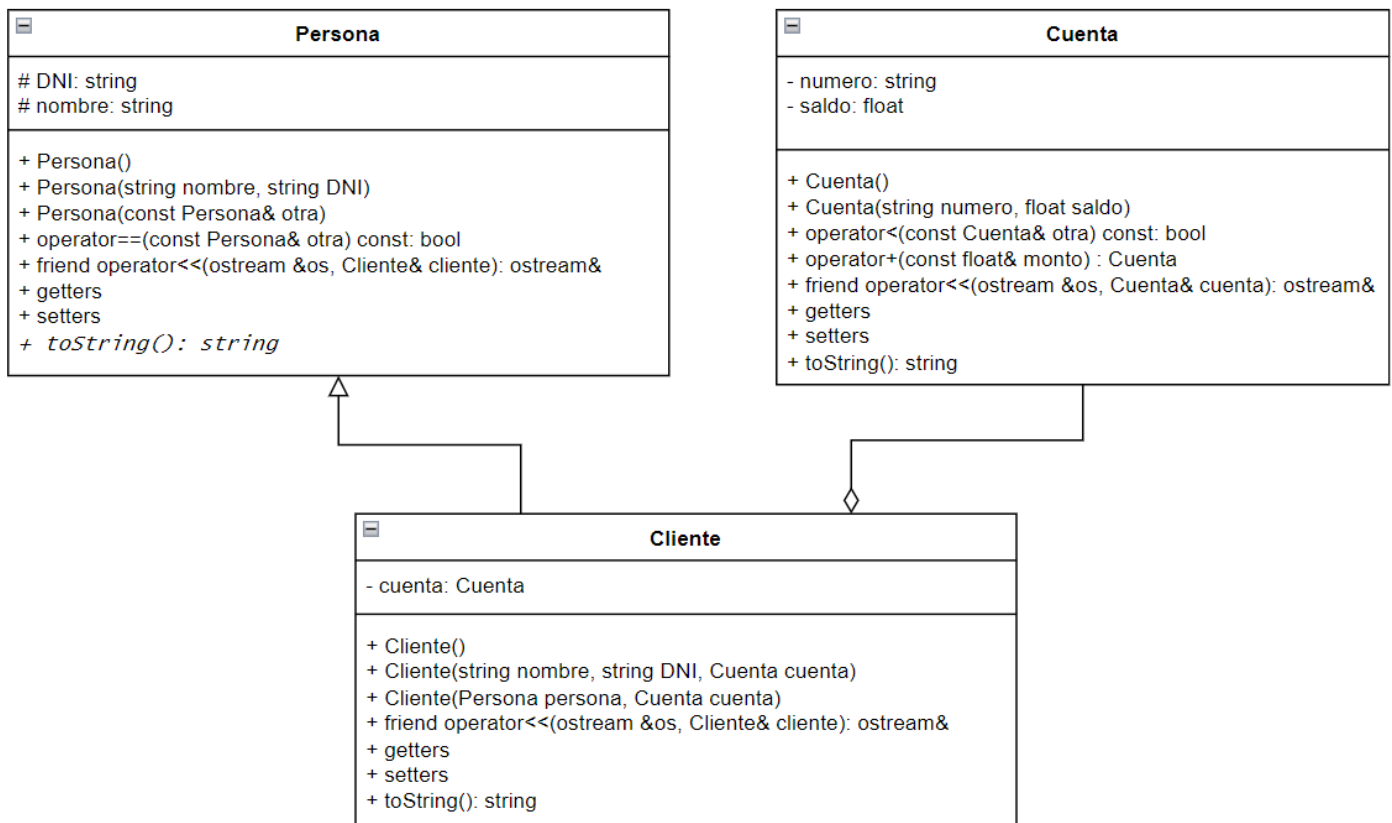
ListaDeReproducción

- Tiene un solo constructor que recibe una canción inicial como parámetro de entrada.
- Por ello, el constructor debe asignar la primera canción al primer elemento del arreglo canciones y el puntero de `reproduciendo` a `nullptr`.
- No se puede crear una lista de reproducción sin, por lo menos, una canción.
- La función `setReproduciendo` es una función sobrecargada con 3 diferentes firmas:
 - `setReproduciendo()`: realiza que el puntero `setReproduciendo()`: realiza que el puntero se asigna con la dirección del primer elemento de la variables canciones y devuelve el valor `True`.
 - `setReproduciendo(int: numero)`: verifica en primer lugar si existe `numero+1` canciones y si es así asigna el puntero al elemento `numero` del arreglo canción y devuelve `True`, en caso contrario devuelve `False`.
 - `setReproduciendo(Canción: canción)`: verifica en primer lugar si existe la canción (parámetro) como una de las canciones del arreglo y si es así asigna el puntero al elemento `numero` del arreglo canciones y devuelve `True`, en caso contrario `False`.
- El `toString` retorna un string que se compone de las `cant_musicas` canciones con sus nombres, y artistas.
- El operador `+=(const Cancion & cancion)` es la única forma que existe de agregar elementos al arreglo canción, el método primero verifica si no ha superado el número máximo de canciones. Se verifica, a continuación, si la canción existe dentro de la **ListaDeReproducción**,

si no es así entonces se agrega al arreglo y luego se aumenta en uno a la cant_canciones, en el caso de que exista no se realiza nada. Por otro lado, en el caso de que se haya superado el número máximo de canciones, se reemplaza la primera canción con la canción parámetro.

Crear un programa principal que demuestre la funcionalidad completa de las clases.

EJERCICIO 2 . Los clientes de un banco ficticio se almacenan según el siguiente diagrama:



Persona

Tiene 3 constructores, 2 funciones de get y 2 de set.

- El operador == debe comparar si dos Personas tienen el mismo DNI
- El toString devuelve "Objeto Persona: Nombre, DNI"
- El operador de igualdad verifica si tanto el nombre como la duración de 2 canciones son datos iguales.
- El operador de flujo de salida << devuelve "DNI – Nombre"

Cuenta:

- Tiene 2 constructores, 2 funciones de get y 2 de set.
- El operador de menor que verifica si el saldo de la cuenta es menor al de otra cuenta.
- El operador de suma agrega el saldo a la cuenta
- El operador de flujo de salida << devuelve "Cuenta – Saldo"
- El toString devuelve "Objeto Cuenta: numero, saldo"

Cliente:

- Hereda de la clase Persona y agrega un objeto de la clase Cuenta, tiene 3 constructores.

- El toString retorna un string que se compone de “Objeto Cliente: Nombre, DNI, Cuenta, Saldo”
- El operador de flujo de salida << devuelve “DNI – Cuenta – Saldo”

Ninguna clase es amiga de otra por lo que deberá usar siempre las funciones de getter según el caso.
Crear un programa principal que demuestre la funcionalidad completa de las clases.