

Multimedia Set-Top Box Program in C++

Author: Thalís Rocha Pestana

Date: 2023-02-2023

INF224 - Télécom Paris

Execution:

To execute the program type on the terminal: 'make run'. You will need a terminal for each directory 'swing' and 'cpp'.

To play the multimedia object you need to put the name of the multimedia object (fla.jpg, for example) and press the 'play' button. (You need to type all name of your file, example: `image.jpeg` .

To show the properties of a multimedia object or a group you need to put the name of the multimedia object (fla.jpg, for example) and press the 'show' button. It is not possible to 'play' a group. In this case the main window will show the properties of the multimedia objects in this group.

Pay attention to put the correct location for the file according to your machine. Enjoy :)

Default data:

- Images: 'ronaldin.jpg', 'fla.jpg';
- Videos: 'vizinhos.mp4', 'liberta.mp4'

C++ specifications:

This repository contains a C++ program that implements a software of a multimedia set-top box allowing to play music, videos, films, to display photos and show all the multimedia properties.

The program consists of the following files:

- `Base.h` : This file defines the Base class, which serves as an abstract base class for different types of multimedia objects. It contains common attributes and methods that are shared by all subclasses.
- `Base.cpp` : Base.h implementation.
- `Film.h` : Definition of the Film class, which is a derived class of the Video class. It adds the capability of managing chapters to the video functionality.
- `Video.h` : Contains the declaration of the Video class, which inherits from the Base class and represents a video object.

- `Photo.h` : This file contains the definition of the `Photo` class, which inherits from `Base` and represents a photo media file with a name, path, and location coordinates.
- `Group.h` : Defines the `Group` class, which represents a group of `Base` objects implemented as a list of `shared_ptr<Base>` pointers.
- `Administrator.h` : This file contains the `Administrator` class which is responsible for managing and creating multimedia and groups.

C++ Project stages:

1 - Startup:

Basic commands to open terminal and create the files needed.

2 - Base Class:

Creation of the above described class `Base`.

3 - Test Program:

Creation of a `main()` function and a few instances of the base class (using `new`) in the main to verify the smooth running of the `Base` class. Definition of the accessors as `const` function to respect the principle of encapsulation.

4 - Photos and Videos:

Creation of the `Photo` and `Video` subclasses as above described. Declaration and Implementation of the constructors, accessors, modifiers, `viewVariables()` and `playMedia()` methods to show the properties and play the multimedia objects, respectively.

The `playMedia()` method is an abstract method because each object (`Photo`, `Video` or `Film`) requires a different program to play. To declare this method the `Base` class has the following declaration:

```
virtual void playMedia() = 0;
```

and the implementation is made in the subclasses. Therefore, is no more possible to instantiate an object from `Base` because after this declaration it becomes an abstract class.

5 - Uniform processing:

Thanks to the above implementation it is possible to use polymorphism to process a list comprising photos or videos independently of its type.

In C++ we need to define the parent class methods as `virtual` in order to use the implementation declared in the child classes.

6 - Films and Arrays

To ensure that the Film object has full control over its data and that its array of durations cannot be modified without its knowledge, it is important to create a copy of the contents of the array passed as a parameter. Furthermore, to prevent any modification of the array returned by an accessor, the `const` keyword can be utilized (signature: `const int *getChapters`). It is also necessary to make a deep copy of the array, which can be achieved by creating a copy constructor and overloading the assignment operator.

7 - Object Destruction and Copying

It is essential to avoid memory leaks when working with dynamically allocated memory. In the Film destructor, the "delete" keyword must be called to free up the memory allocated for the duration array. Additionally, object copying can be problematic if a new object is created and then forgotten to be deleted. Therefore, it is important to always remember to call "delete" to avoid memory leaks.

8 - Creating Groups

The Group class inherits from the `list<>` template class. Since the objects in the group may not all be of the same size, object pointers are required. In Java, references (similar to pointers) are automatically used.

9 - Automatic Memory Management

In the upcoming stages, shared pointers (smart pointers) will be utilized for automatic memory management. When an object no longer belongs to any group, it will be automatically destroyed.

10 - Consistent Data Management

The Administrator class is responsible for managing Base objects and groups. This class provides a more consistent data management approach, enabling the creation of any type of multimedia object or group, searching and displaying multimedia objects and groups by name, and playing multimedia objects.

11 - Client/Server

The `main()` function is utilized to handle requests, with the result being returned in the "response" string. This enables effective communication between the client and server, facilitating the sharing of data and resources.

Java Project Stages:

1 - Creation of main window:

Creation of a MainWindow.java (deriving from JFrame) containing a multi-line text area (JTextArea) and three buttons (JButton).

2 - Improvement of main window:

Improvement of main window including fieldText for the user input text field, textArea for the output text area, and three Action objects for the three buttons. The "Play" button sends a request to a server to play a multimedia object specified in the text field, and the "Show" button sends a request to show the multimedia variables. The "Exit" button terminates the application. The window also has a menu bar with the same options.

3 - Client/Server Interaction

Client/Server interaction using Client.java.