

CAR DESIGNER

1.Statement of Work:

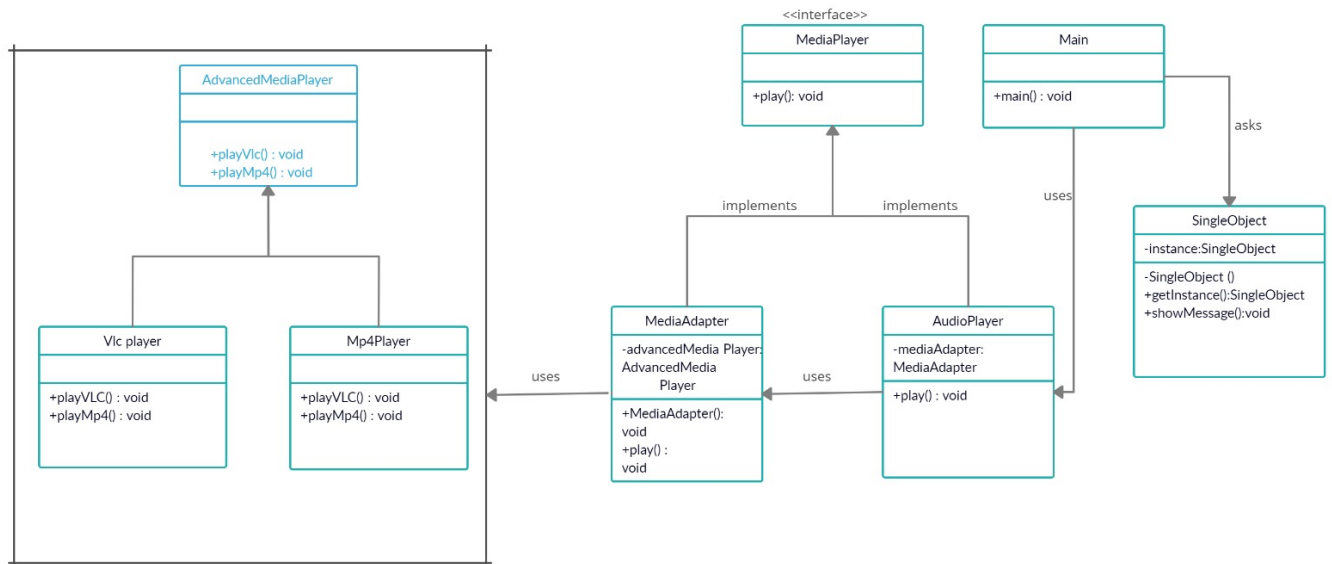
There are many audio player applications nowadays. As the technology develops, new audio file extensions are also being created every day, so the application may not satisfy the users after a few years. The app developer may then want to add these incoming features to their app. No need to rewrite the app. As a result, developers can be offered an easier code style. In short, I think that the work of program companies can be compatible with the software world. In this way, developers can adapt to technology more easily.

2. Design Patterns:

In this study, I chose to use adapter pattern and singleton pattern to solve the problem, these models can be a useful solution for him. Ok why?

Because I used a adapter pattern We want to use an advanced audio player in an audio player where the device can only play mp3 files and can play vlc and mp4 files. Second, I used a singleton pattern because This pattern involves a single class which is responsible to create an object while making sure that only single object gets created. This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class. It allows us to inform with a single message whether the file the user wants to open is compatible or not.

3. UML:



4. Research:

I searched for free and open source and saw many projects related to this on the internet. I saw that these two patterns are used in much more advanced applications, that is, in the known audio players that are available to everyone.

Testing Class

```
public class Main {
    public static void main(String[] args) {
        AudioPlayer audioPlayer = new AudioPlayer();
        Singleton object = Singleton.getInstance();

        audioPlayer.play("mp3", "tarkan.mp3");
        audioPlayer.play("mp4", "ferdi.mp4");
        audioPlayer.play("vlc", "müslüm.vlc");
        audioPlayer.play("AAC", "orhan.AAC ");
    }
}
```

FURKAN ZORLU

15803560996