Practice Peer-graded Assignment: Ungraded Assessment – Composite Pattern

Ready for the assignment?
You will find instructions below to submit.

Instructions

My submission

Review classmates

Learn how to apply the Composite pattern.

Discussions

Review criteria less ^

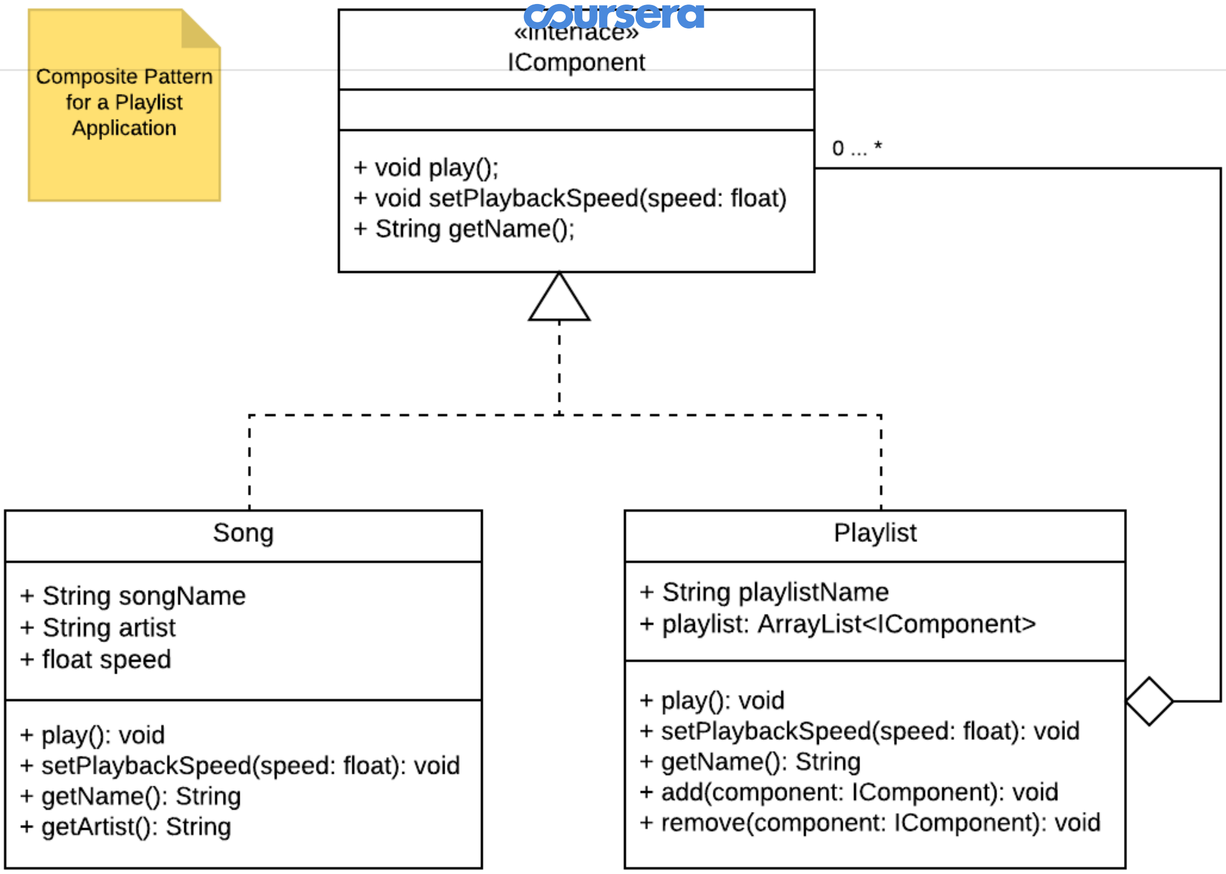
You have been asked to create a playlist application that will be used on Android devices (using the Java language). We will assume that each playlist can be composed of songs or other playlists, or a combination of both.

Your project manager has told you that the composite pattern is best used in this situation. The following UML class diagram that communicates the application's objects and relationships using the composite pattern.

In this assignment you are required to complete the provided code. (Note: With the exception of the Playlist class, you do not need to actually implement the methods, just write filler comments (eg., // play song). With the Playlist class, write out the method to add songs to the playlist).

UML Class Diagram less ^

Use the UML class diagram pictured below to help modify the provided code.


Composite Pattern for a Playlist Application
«interface»
IComponent
+ void play();
+ void setPlaybackSpeed(speed: float)
+ String getName();
0 ... *
Song
+ String songName
+ String artist
+ float speed
+ play(): void
+ setPlaybackSpeed(speed: float): void
+ getName(): String
+ getArtist(): String
Playlist
+ String playlistName
+ playlist: ArrayList<IComponent>
+ play(): void
+ setPlaybackSpeed(speed: float): void
+ getName(): String
+ add(component: IComponent): void
+ remove(component: IComponent): void

Code

less ⌃

```
--------------------
[Program.java]
--------------------
public class Program {

  public static void main(String args[]) {

  // Make new empty "Study" playlist
  Playlist studyPlaylist = new Playlist("Study");

  // Make "Synth Pop" playlist and add 2 songs to it.
  Playlist synthPopPlaylist = new Playlist("Synth Pop");
  Song synthPopSong1 = new Song("Girl Like You", "Toro Y Moi" );
  Song synthPopSong2 = new Song("Outside", "TOPS");
  synthPopPlaylist.add(synthPopSong1);
  synthPopPlaylist.add(synthPopSong2);

  // Make "Experimental" playlist and add 3 songs to it,
  // then set playback speed of the playlist to 0.5x
  Playlist experimentalPlaylist = new Playlist("Experimental");
  Song experimentalSong1 = new Song("About you", "XXYYXX");
  Song experimentalSong2 = new Song("Motivation", "Clams Casino");
  Song experimentalSong3 = new Song("Computer Vision", "Oneohtrix Point Never");
  experimentalPlaylist.add(experimentalSong1);
  experimentalPlaylist.add(experimentalSong2);
  experimentalPlaylist.add(experimentalSong3);
  float slowSpeed = 0.5f;
  experimentalPlaylist.setPlaybackSpeed(slowSpeed);

  // Add the "Synth Pop" playlist to the "Experimental" playlist
  experimentalPlaylist.add(synthPopPlaylist);

  // Add the "Experimental" playlist to the "Study" playlist
  studyPlaylist.add(experimentalPlaylist);

  // Create a new song and set its playback speed to 1.25x, play this song,
  // get the name of glitchSong → "Textuell", then get the artist of this song → "Oval"
  Song glitchSong = new Song("Textuell", "Oval");
  float fasterSpeed = 1.25f;
  glitchSong.setPlaybackSpeed(fasterSpeed);
  glitchSong.play();
  String name = glitchSong.getName();
  String artist = glitchSong.getArtist();
  System.out.println ("The song name is " + name );
  System.out.println ("The song artist is " + artist );

  // Add glitchSong to the "Study" playlist
  studyPlaylist.add(glitchSong);

  // Play "Study" playlist.
  studyPlaylist.play();

  // Get the playlist name of studyPlaylist → "Study"
  System.out.println ("The Playlist's name is " + studyPlaylist.getName() );
  }
}

--------------------
[IComponent.java]
--------------------
public interface IComponent {

// Your code goes here!




}

--------------------
[Playlist.java]
--------------------
public class Playlist implements IComponent {

  public String playlistName;
  public ArrayList<IComponent> playlist = new ArrayList();

  public Playlist(String playlistName) {
    this.playlistName = playlistName;
  }

  // Your code goes here!

}

--------------------
[Song.java]
--------------------
public class Song implements IComponent {
  public String songName;
  public String artist;
  public float speed = 1; // Default playback speed

  public Song(String songName, String artist ) {
    this.songName = songName;
    this.artist = artist;
  }

  // Your code goes here!

}

```