1) Write a Java program to create a class called "MusicLibrary" with a collection of songs and methods to add and remove songs, and to play a random song.

The above “MusicLibrary” class represents a library of songs. It uses an ArrayList to store the songs, and provides methods to add and remove songs from the library. It also has a method to get a list of all the songs in the library, and a method to play a random song from the library.

The “Song” class represents a song in a music library. It has two private instance variables, the title and the artist, and a constructor that takes these two variables as parameters. The class also provides getters and setters for the title and artist variables, allowing the client code to access and modify the song information.

The Main class creates an instance of MusicLibrary and adds several Song objects to it. It then retrieves and prints out the list of all songs in the library. Finally, it calls the playRandomSong() method of the MusicLibrary object three times to play a random song each time, printing out the details of the played song.

Expalintion

playRandomSong -> use Random class to generate random no’s and bound should be equal to the size of ArrayList so that it will generate any random index and based on index you can get the song stored atthat index ..

2) Given an unsorted list of integers, find maximum and minimum values in it.

3) Split a List into Two Halves in Java

**Input :** list = {1, 2, 3, 4, 5, 6}

**Output :** first = {1, 2, 3}, second = {4, 5, 6}

**Input :** list = {1, 2, 3, 4, 5}

**Output :** first = {1, 2}, second = {3, 4, 5}

4) Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.

5) Write a Java program to create an enum called "Weekend" with constants representing the days of the weekend.

6) Write a Java program to create a class called "Course" with attributes for course name, instructor, and credits. Create a subclass "OnlineCourse" that adds attributes for platform and duration. Implement methods to display course details and check if the course is eligible for a certificate based on duration.

Explaination:

Course Class:

* Attributes: courseName, instructor, and credits.
* Constructor: Initializes the attributes.
* displayCourseDetails(): Prints the course details.
* Getters: Methods to get the values of the attributes.

OnlineCourse Class:

* Extends Course.
* Additional Attributes: platform and duration.
* Constructor: Initializes the attributes, calling the superclass constructor for the common attributes.
* displayCourseDetails(): Overridden to include additional details specific to online courses.
* isEligibleForCertificate(): Checks if the course duration is at least 10 hours to be eligible for a certificate.
* Getters: Methods to get the values of the additional attributes.