

# Задание 1

Существуют различные MP3 плееры. Каждый из них чем-то отличается от другого. В задаче рассмотрим несколько видов плееров начиная от дешевых к дорогим. Суть задания заключается в создании классов, которые будут описывать все эти плееры. В главном классе программы необходимо создать объекты каждого класса и продемонстрировать их работу.

Плееры работают с таким понятием как “**песня**” и “**плейлист**”.

**Песню** будем рассматривать как строку, которая состоит из названия песни. К примеру вот как выглядит одна песня

```
String song1 = "The Best Song";
```

**Плейлист** выглядит как массив песен(строк).

```
String[] playlist = new String[] { "The best song", "Good song", "Super Song" };
```

Проигрывая песню, плееру необходимо вывести в консоль - "Playing: " + songName;

Например вот такой вывод должен получиться при проигрывании песни "The best song":

*Playing: The best song*

**Код логики методов не должен дублироваться в классах!**

**Плееры не могут иметь больше возможностей, чем указано!**

**Каждый новый плеер == новый класс**

Каждый плеер имеет метод **public void playSong**

Проигрывает первую имеющуюся песню на плеере.

Некоторые плееры имеют метод **public void playAllSongs()**

Проигрывает все песни на плеере

Плеер 1.

Имеет final цену(задается в конструкторе) и геттер

Имеет только 1 песню (нельзя объявить эту переменную в классе этого плеера)

**playSong** Может проиграть песню.

Плеер 2.

Имеет final цену(задается в конструкторе) и геттер

Имеет только 1 песню (нельзя объявить эту переменную в классе этого плеера)

**playSong** Пытаясь проиграть песню выдает ошибку в консоль - "error".

Плеер 3.

Имеет final цену(задается в конструкторе) и геттер

Имеет плейлист

**playSong** Может проиграть первую песню

**playAllSongs** может проиграть все песни

Плеер 4.

Имеет final цену(задается в конструкторе) и геттер

Имеет плейлист

**playSong** Может проиграть **последнюю** песню

**playAllSongs** может проиграть все песни

Плеер 5.

Имеет final цену(задается в конструкторе) и геттер

Имеет плейлист

**playSong** Может проиграть первую песню

**playAllSongs** может проиграть все песни с конца плейлиста в начало

Плеер 6.

Имеет final цену(задается в конструкторе) и геттер

Имеет плейлист

**playSong** Может проиграть первую песню

**playAllSongs** может проиграть все песни

Имеет метод **public void shuffle()** - перемешивает все песни в плейлисте местами

## Задание 2 (дополнительное)

Научить плееры отрисовывать свой графический интерфейс(User Interface) с помощью JavaFX, чтобы пользователь мог в графическом представлении работать с плеером, без консоли.

Добавить к классам плееров метод **public void show(Pane root)**.

Метод должен уметь отрисовать в окно все функции плеера. Не забываем о наследовании и переопределении методов при создании этого метода.

Добавить визуализацию проигрывания песен и кнопок(функций плееров) в любом удобном для вас виде, но не консольном.

## Задание 3 (дополнительное high-level)

Самостоятельно наугад и разобраться как проигрывать mp3 файлы.

Создать новый главный класс программы - **RealPlayer** и инициализировать там метод main который будет запускать плеер с графическим интерфейсом на JavaFX.

Научить плееры пользоваться mp3 файлами, а вместо строк выводить теперь настоящие названия песен.