**JavaScript OOP 2015 - Audio Player**

* Create an object (module) that enables creation of the following:

**Player**

that has the following:

* properties:
  + name - provided through a constructor function or init method
* methods:
  + addPlaylist(playlistToAdd)
    - Adds a playlist to the player
    - playlistToAdd must be a PlayList instance
      * Otherwise, throw an error
    - Enables chaining
  + getPlaylistById(id)
    - Finds and returns a playlist from the playlists in this player instance
    - Returns null, if a playlist with the provided id is not contained in the player
  + removePlaylist(id)
    - Removes a playlists from this player instance, and the playlist must have an id equal to the provided id
    - Enables chaining
    - Throws an error, if a playlist with the provided id is not contained in the player
  + removePlaylist(playlist)
    - Removes a playlists from this player instance, and the playlist must have an id equal to the id of the provided playlist
    - Enables chaining
    - Throws an error, if a playlist with the provided id is not contained in the player
  + listPlaylists(page, size)
    - Returns an array with at most size size and containing the playlists with indices page*size, page*size+1, page*size+2, ... (page+1)*size-1
      * These are the indices after sorting the playlist by name, then by id
    - If there are less that size playlists, return all of them
    - Throw an error if:
      * page\*size > COUNT\_OF\_PLAYLISTS\_IN\_PLAYER
      * page < 0
      * size <= 0
  + contains(playable, playlist)
    - Checks whether a playlist contains a playable
    - Returns true or false
    - **Example**:
    - //having 35 playlists:
    - //listPlaylists(2, 10) should return playlists with indices from 20 to 29, including
    - //listPlaylists(3, 10) should return playlists with indices from 30 to 34, including
  + search(pattern)
    - Returns an array of all playlists, that contain a song, which title contains the pattern
      * In the returned array, only id and title of the playlists are returned
    - Returns empty array ig no such playlists exists
    - **Example**:
    - //having player with name `Batman's playlist` with:
    - //playlist1 with id `1` and title `Cool` with playables:
    - // 'They are green' and 'I am Batman'
    - //playlist2 with id `2` and title `Green` with playables:
    - // `Green they are`, `Green is beautiful` and `To the green and beyond`
    - //player.search('green') returns:
    - // [{name: 'Cool', id: 1}, {name: 'Green', id: 2}]
    - //player.search('batman') returns:
    - // [{name: 'Cool', id: 1}]
    - //player.search('John') returns:
    - // []

**PlayList**

that has the following:

* properties:
  + id - generated when the playlist is created
  + name - provided through a constructor function or init method
* methods:
  + addPlayable(playable)
    - Adds a playable to the PlayList
    - The same playable can be added multiple times
    - Enables chaining
  + getPlayableById(id)
    - Returns the playable that has the provided id
    - Returns null, if no playable is found with the provided id
  + removePlayable(id)
    - Removes a playable from this playlist, and the playable must have an id equal to the provided id
    - Enables chaining
    - Throws an error, if a playable with the provided id is not contained in the playlist
  + removePlayable(playable)
    - Removes a playable from this playlist, and the playable must have an id equal to the id of the provided playable
    - Enables chaining
    - Throws an error, if a playable with the provided id is not contained in the playlist
  + listPlayables(page, size)
    - Returns an array with at most size size and containing the playables with indices page\*size, page\*size+1, page\*size+2, ... (page+1)\*size-1
      * These are the indices after sorting the playables by title, then by id
    - If there are less that size playables, return all of them
    - Throw an error if:
      * page\*size > COUNT\_OF\_PLAYABLE\_IN\_PLAYLIST
      * page < 0
      * size <= 0
    - **Example**:
    - //having 35 playables:
    - //rockPlaylist.listPlayables(2, 10) should return playables with indices from 20 to 29, including
    - //rockPlaylist.listPlayables(3, 10) should return playables with indices from 30 to 34, including

**Playable**

that has the following :

* properties:
  + id - generated when the playable is created
  + title - provided through a constructor function or init method
  + author - provided through a constructor function or init method
* methods:
  + play()
    - returns a string in the format: [id]. [title] - [author]

**Audio**

that inherits Playable and adds the following:

* properties:
  + length
    - provided through a constructor function or init method
    - number, greater than **0**
* methods:
  + play()
    - reuses the play() from Playable and adds: - [length] at the end

**Video**

that inherits Playable and adds the following:

* properties:
  + imdbRating
    - provided through a constructor function or init method
    - number, between **1** and **5**
* methods:
  + play()
    - reuses the play() from Playable and adds: - [imdbRating] at the end

**General requirements:**

* All id are numbers, greater than **0**
* Each player instance has an unique id
* Each playlist instance has an unique id
* Each playable instance has an unique id
* Properties name, title and author are a strings between **3** and **25** characters
* You can use any Inheritance in JavaScript you like
  + For example: Prototypal or Classical

**The module should look as follows:**

var module = {

getPlayer: function (name){

// returns a new player instance with the provided name

},

getPlaylist: function(name){

//returns a new playlist instance with the provided name

},

getAudio: function(title, author, length){

//returns a new audio instance with the provided title, author and length

},

getVideo: function(title, author, imdbRating){

//returns a new video instance with the provided title, author and imdbRating

}

};