

集合综合案例

播放器管理

- 需求分析
- 项目演示
- 详细设计
- 代码实现

需求分析

- 播放列表管理
- 播放器管理

需求分析

播放列表管理主要功能：

- **将歌曲添加到主播放列表**
- **将歌曲添加到普通播放列表**
- **通过歌曲id查询播放列表中的歌曲**
- **通过歌曲名称查询播放列表中的歌曲**
- **修改播放列表中的歌曲**
- **删除播放列表中的歌曲**
- **显示播放列表中的所有歌曲**

需求分析

播放器管理主要功能：

- **向播放器添加播放列表**
- **从播放器删除播放列表**
- **通过名字查询播放列表信息**
- **显示所有播放列表名称**

项目演示



慕课网
imooc.com



慕课网
imooc.com



慕课网
imooc.com



慕课网
imooc.com

详细设计

- **类：**
- **歌曲类 (Song)**
- **播放列表类 (PlayList)**
- **播放器类 (PlayListCollection)**
- **测试类 (TestDemo)**

详细设计

歌曲类 (Song) :

属性：歌曲id (id)、歌曲名(name)、演唱者(singer)，均为字符串类型

方法：

- 构造方法
- getter和setter方法
- hashCode()和equals()方法
- toString()方法

详细设计

- 播放列表类 (PlayList)
- 属性：
 - 播放列表名称 (playListName) : 字符串类型
 - 播放列表中的歌曲集合 (musicList) : List类型

详细设计

播放列表类 (PlayList)

方法：

- 构造方法
- getter和setter方法
- 将歌曲添加到播放列表：`public void addToPlayList(Song song);`
- 显示播放列表中所有歌曲：`public void displayAllSong();`
- 通过id查询歌曲：`public Song searchSongById(String id);`

详细设计

播放列表类 (PlayList)

方法：

- 通过名称查询歌曲：`public Song searchSongByName(String n);`
- 修改歌曲：`public void updateSong(String id, Song song);`
- 从播放列表删除歌曲：`public void deleteSong(String id);`
- 按歌曲名进行排序：`public void sortBySongName();`

详细设计

播放器类 (PlayListCollection)

属性：

- **存放播放列表的集合 (playListMap) :Map**

播放器类 (PlaylistCollection)

方法：

- **构造方法**
- **getter和setter方法**
- **添加播放列表：public void addPlayList(PlayList playList);**
- **删除播放列表：public void deletePlayList(PlayList playList);**
- **通过名字查询：public PlayList searchPlayListByName(String playListName);**
- **显示所有播放列表名称：public void displayPlayListName();**

课程总结

主要功能：

- **播放列表管理**
- **播放器管理**

课程总结

四个类：

- 歌曲类 (Song)
- 播放列表类 (PlayList)
- 播放器类 (PlayListCollection)
- 测试类 (TestDemo)

课程总结

- 对集合List、Set和Map进行了应用
- 使用了泛型
- 了解了代码的编写和测试流程