

# 文本复制检测报告单(全文标明引文)

№:ADBD2018R\_2018053015312720180530154814440173796299

检测时间:2018-05-30 15:48:14

检测文献: 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现

作者: 沈建

检测范围: 中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

CNKI大成编客-原创作品库

个人比对库

时间范围: 1900-01-01至2018-05-30

## 检测结果

总文字复制比: 30%

跨语言检测结果: 0%

去除引用文献复制比: 30%

去除本人已发表文献复制比: 30%

单篇最大文字复制比: 20.4% (Android播放媒体详解 - jimo)

重复字数: [9217]

总段落数: [7]

总字数: [30676]

疑似段落数: [6]

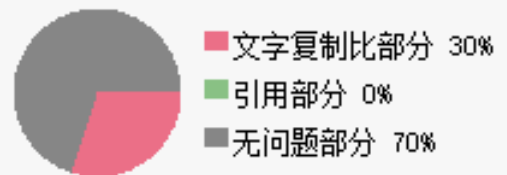
单篇最大重复字数: [6261]

前部重合字数: [642]

疑似段落最大重合字数: [7236]

后部重合字数: [8575]

疑似段落最小重合字数: [92]



指标: ☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 疑似自我剽窃 ☐ 疑似整体剽窃 ☐ 过度引用

表格: 0

公式: 0

疑似文字的图片: 0

脚注与尾注: 0

6.4% ( 121 ) 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第1部分 ( 总1895字 )

0% ( 0 ) 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第2部分 ( 总411字 )

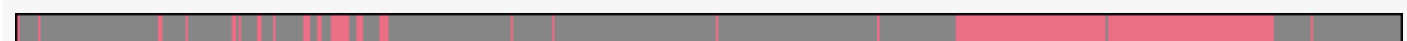
6.5% ( 167 ) 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第3部分 ( 总2570字 )

37.4% ( 1506 ) 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第4部分 ( 总4027字 )

1.1% ( 95 ) 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第5部分 ( 总9038字 )

64.4% ( 7236 ) 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第6部分 ( 总11235字 )

6.1% ( 92 ) 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第7部分 ( 总1500字 )



( 注释: 无问题部分 文字复制比部分 引用部分 )

相似文献列表    文字复制比：6.4%(121)    疑似剽窃观点：(0)		
1	车商APP的设计与实现 顾彬 - 《大学生论文联合比对库》 - 2017-05-22	3.4% ( 64 ) 是否引证：否
2	04110122_李思民_每日新闻ios客户端设计与实现 李思民 - 《大学生论文联合比对库》 - 2015-04-28	3.0% ( 57 ) 是否引证：否
3	基于iOS的车载Wi-Fi客户端设计 程风 - 《大学生论文联合比对库》 - 2015-05-14	2.4% ( 46 ) 是否引证：否
4	基于IOS的新闻应用的设计与实现 王浩 - 《大学生论文联合比对库》 - 2016-05-17	2.4% ( 46 ) 是否引证：否
5	基于iOS的减肥健身软件的开发与实现 谭靖珩 - 《大学生论文联合比对库》 - 2016-06-10	2.4% ( 46 ) 是否引证：否
6	基于iOS系统平台的手机APP开发 贺进立 - 《大学生论文联合比对库》 - 2016-05-29	1.8% ( 34 ) 是否引证：否
7	基于iOS的视频点播软件的设计与实现 刘梦凯 - 《大学生论文联合比对库》 - 2015-05-22	1.7% ( 33 ) 是否引证：否
8	基于iOS平台的热点资讯APP设计与实现 孙赵凯 - 《大学生论文联合比对库》 - 2016-06-01	1.6% ( 30 ) 是否引证：否

原文内容    红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

摘要

基于Android平台的音乐播放器设计与实现

如今智能手机日渐普及，手机的功能越来越多样化，不再只是用来打电话、发短信的通讯工具，人们对手机的娱乐功能要求越来越高，而其中一个必不可少的功能就是播放音乐。本文要研究的就是如何设计并实现基于Android平台的音乐播放器。

Android OS应用层采用了Java语言来编程。Java是一种面向对象的半编译半解释型语言，有简单易用、功能强大、跨平台的特点。在数据存储方面，Android OS内置了嵌入式数据库SQLite。

此次的应用程序设计，本人选用了Android Studio集成开发环境，主要针对MP3格式的音乐文件，实现了一些音乐播放的基础功能，如播放、暂停、停止、上下跳转、歌词显示、图片显示。考虑到用户对手机音乐播放的功能需求，以及触屏手机的交互方式，设计了较为友好的用户界面。文中给出了详细的设计过程，并对一些问题进行探讨。

关键字：Android ， 音乐播放器 ， Java ， MP3格式

Abstract

Design and Implementation of Music Player

Based on Android Platform

In recent years, smart phones have become more and more popular, and the functions of mobile phones have become more and more diversified. Mobile phones are no longer just communication tools for calling and texting. People are increasingly demanding the entertainment function of mobile phones, and one of the essential functions is to play music. This paper will study how to design and implement music player based on the Android platform.

The application layer of Android OS is programmed with the Java language. Java is an object-oriented, semi-compiled, semi-interpreted language that is easy to use, powerful, and cross-platform. In terms of data storage, Android OS has built-in embedded database SQLite.

In the design of this application, I chose the Android Studio integrated development environment, and realized some basic function for music (mainly for MP3 music files) playing, such as to play, to pause, to stop, to jump up and down, to display lyrics, to display image.

Considering the user's functional requirements for music playback on mobile phone and the interaction mode of touchscreen mobile phone, a friendly user interface is designed. This paper gives a detailed design process and discusses some problems.

Keywords: Android , music player , Java , MP3 format

目录

第1章绪论 6

1.1 研究背景 6

1.1.1 国内外现状 6

1.1.2 Android OS 7

## 指 标

## 疑似剽窃文字表述

1. 智能手机日渐普及，手机的功能越来越多样化，不再只是用来打电话、发短信的通讯工具，人们对手机的娱乐功能要求越来越高，而其中

## 2. 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第2部分

总字数：411

相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

## 第2章技术准备 9

## 2.1 技术概览 9

## 2.1.1 Android系统结构 9

## 2.1.2 Android开发环境 9

## 2.1.3 Android应用程序特点 10

## 2.2 应用程序框架 10

## 2.2.1 Android应用程序组件 10

## 2.2.2 Android Manifest 12

## 2.2.3 SQLite数据库 13

## 2.3 音频文件处理 1.....3

## 第3章程序设计 1.....5

## 3.1 分析 1.....5

## 3.1.1 可行性分析 1.....5

## 3.1.2 需求分析 1.....5

## 3.1.3 开发环境搭建 17

## 3.2 设计 17

## 3.2.1 交互事件 17

## 3.2.2 布局管理 26

## 3.2.3 数据库设计 27

## 3.3 编码 28

## 3.4 测试 39

## 第4章结论 47

## 参考文献 49

## 致谢 50

## 3. 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第3部分

总字数：2570

相似文献列表 文字复制比：6.5%(167) 疑似剽窃观点：(0)

1	201447201031封宇 1毕业设计 封宇 - 《高职高专院校联合比对库》 - 2017-05-19	4.9% ( 125 ) 是否引证：否
2	基于Android平台的无线点餐系统的设计与实现 崇爱磊 - 《大学生论文联合比对库》 - 2016-06-02	3.5% ( 89 ) 是否引证：否
3	基于QRCode二维码的快速登录方案 李明欣; - 《成都航空职业技术学院学报》 - 2016-12-19	1.6% ( 42 ) 是否引证：否
4	1_边春晖_普通话练习系统的设计与实现 边春晖 - 《大学生论文联合比对库》 - 2017-05-02	1.4% ( 36 ) 是否引证：否

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

## 第1章绪论

## 1.1 研究背景

科技改变生活，如今智能手机日渐普及，手机的功能越来越多样化，不再只是用来打电话、发短信的通讯工具，人们对手机的娱乐功能要求越来越高，而其中一个必不可少的功能就是播放音乐。

听音乐是一项重要的娱乐方式，播放音乐的任务以前主要由电脑承担，现在许多人会把它交给手机来处理，手机音乐播放器就显得尤为重要。许多手机都自带一个简单的音乐播放器，它没有像Windows的千千静听、foobar2000那样丰富的功能。

从磁带到MP3再到手机，是科技的进步，使内容从匮乏到丰富。如果获取内容不成问题，人们的需求会逐渐细化。一些人会注重音乐的管理，耳朵挑剔的人会对音质有要求，等等。设计美观、功能强大、音质优秀的离线音乐播放器会受到他们的青睐。

### 1.1.1 国内外现状

在音乐播放器这方面，国内由各大在线音乐平台主导，有QQ音乐、虾米音乐、网易云音乐等，上线了十年的多米音乐已经停止服务，版权竞争可谓惨烈。而专注于音乐体验的离线音乐播放器，除了早年流行的千千静听，国内几乎无人有兴趣开发。随着千千静听被百度收购，PC端离线音乐播放器基本宣告结束。

到了手机平台也依然如此，几乎没有一款离线音乐播放器能脱颖而出。虽然与互联网的发展和用户习惯有关，这也体现出国内软件行业发展很不健康的情况，无关本文主旨的游戏产业尤甚。

国外的流媒体音乐有Apple Music、Spotify、Pandora Radio等平台，在离线音乐播放器方面也有Winamp、AIMP、foobar2000等优秀作品。

单单就手机端的音乐播放器来说，在国内流行的，都是在线音乐，主打的是内容和社交，纯粹的离线音乐播放器几乎没有。而国外有Poweramp、Rockbox、Neutronmp等，foobar2000也推出了移动版。本文要探讨的就是Android离线音乐播放器的设计与实现。

### 1.1.2 Android OS

Android OS 基于Linux内核，是一个开放的平台。最初是由Danger公司的Andy Rubin领头开发，随后很快就卖身给了Google。Google与各大硬件制造商、软件开发商及电信营运商，组成开放手持设备联盟Open Handset Alliance，来协同研发Android OS和设备。Google以Apache开源许可证的授权方式，发布了Android OS的源代码。

Android凭借其开放性，受到各大手机生产商的重视，迅速占领智能手机市场。其不断优化用户体验，友好而美观的交互界面，逐渐得到消费者的青睐。经过与其他各类移动操作系统激烈的角逐，Android OS已经成为了市场占有率最大的移动平台。在不断的发展中，Android OS已经更新到9.0版本，代号Pistachio Ice Cream。

Android OS在智能手机中占有巨大份额，在其他智能硬件也有不少地盘，如智能手表、智能电视、物联网等领域，因此其发展前景非常可观，自然就需要大量的应用程序开发者。

### 1.1.3 Java语言

Android OS的上层应用程序是用Java语言(Kotlin语言也可以)开发的，运行在内建的Java虚拟机Dalvik/ART之上。

Java语言是由Sun公司(已被Oracle全盘收购)推出的一门编程语言，最初是为了在电视、冰箱等家用电器上进行交互。随着互联网的迅速发展，Java语言却首先在网络上得到广泛的应用。如今的Java语言遍地开花，在桌面应用程序、Web应用程序、分布式系统和嵌入式设备等平台都有应用。

Java语言是一种面向对象的半编译半解释型语言，有简单易用、面向对象、功能强大、多线程、分布式、跨平台、可移植与健壮性、安全性、动态性等优点。作为面向对象编程语言的典范，很好的诠释了面向对象编程的思想，利用Java语言，程序员可以用面向对象的思维方式来开发大型的工程、解决复杂的问题。

相较于C++语言，Java语言是纯粹的面向对象语言，去掉了多重继承、运算符重载、指针、结构体、联合体等复杂的概念，引入了自动内存管理、真正的数组、字符串变量的功能。

由于Oracle公司的不作为，Java语言最近略显颓势，但是Java的编程思想、Java Virtual Machine运行的模式深入人心，基于JVM的新兴编程语言不断被推出，如Scala、Kotlin，它们吸收了Java语言的优点，同时加入的现代编程语言的特点，如泛型、Lambda表达式。

## 1.2 我的工作

本文在了解Android应用程序的四大组件、各种数据存储方式、音乐播放的实现方式的基础上，遵照软件工程的设计流程，来设计并实现这款Android音乐播放器。

我查找了关于Android应用程序设计和音乐播放器开发的相关背景知识，了解国内外在这方面的研究现状。重点学习了Android OS的多媒体技术、SQLite数据库和XML语言界面设计。这些是我工作的基础。

设计并实现基于Android的音乐播放器，需要调查用户对音乐播放器的要求，确定音乐播放器的逻辑功能和框架。在此我并没有做用户需求的调查，因为这里并不是一个商业行为，而是一次Android开发的尝试。我附会了现流行的手机音乐播放器的设计习惯，依样画葫芦，加入了一些自己的想法，并按照自己的喜好来调整。

我采用了软件工程的结构化思想和标准软件设计流程。我认为这是有必要提前适应的，将来在公司的工作，一个大型的工程，肯定是多人协作来完成的，必须遵守科学的规则才行。

搭建Android Studio开发环境，安装Java SDK和Android SDK，安装Android模拟器。我边学边做，完成了这款Android音



乐播放器，实现了读取文件、播放控制、音乐分类、背景播放、歌词显示、图片显示、通知栏控制等功能模块。经过软件测试，基本达到预期的要求。

当然，本人作为Android开发的新手，在几个月内是不可能拿出与前面点到的Poweramp、Rockbox相提的那种程度的作品，此次的毕业设计旨在锻炼自己的编程能力，熟悉整个Android 应用程序开发的过程，争取在能力范围内做到最好，希望对将来的工作有所帮助。

指 标		
疑似剽窃文字表述		
<div> <div>1. Google。Google与各大硬件制造商、软件开发商及电信营运商，组成开放手持设备联盟Open Handset Alliance，来协同研发Android</div> <div>2. Google以Apache开源许可证的授权方式，发布了Android OS的源代码。</div> <div>Android</div> </div>		
4. 53120801_沈建_计算机科学与技术_基于Android平台的音乐播放器设计与实现.doc_第4部分		总字数：4027
相似文献列表 文字复制比：37.4%(1506) 疑似剽窃观点：(0)		
1	手机点名签到系统设计与开发 俞震东 - 《大学生论文联合比对库》 - 2016-03-30	10.1% ( 408 ) 是否引证：否
2	文献综述 - 《互联网文档资源 ( <a href="http://wenku.baidu.c">http://wenku.baidu.c</a> ) 》 - 2017	6.9% ( 279 ) 是否引证：否
3	杨锋毕业论文 杨锋毕业论文 - 《高职高专院校联合比对库》 - 2017-05-18	5.2% ( 211 ) 是否引证：否
4	杨锋-毕业论文 杨锋 - 《高职高专院校联合比对库》 - 2017-05-19	5.2% ( 211 ) 是否引证：否
5	基于安卓系统的科学计算器开发 茅天宇 - 《大学生论文联合比对库》 - 2017-05-12	5.2% ( 211 ) 是否引证：否
6	基于Andriod的计算器的设计与实现 江雪松 - 《大学生论文联合比对库》 - 2017-06-04	5.2% ( 211 ) 是否引证：否
7	Android平台上的计算器设计_图文 - 《互联网文档资源 ( <a href="http://wenku.baidu.c">http://wenku.baidu.c</a> ) 》 - 2017	5.2% ( 211 ) 是否引证：否
8	android实现计算器 - 《互联网文档资源 ( <a href="http://wenku.baidu.c">http://wenku.baidu.c</a> ) 》 - 2017	5.2% ( 211 ) 是否引证：否
9	基于Android平台手机图形编辑软件的设计与实现 熊文阔(导师：宋茂强) - 《北京邮电大学博士论文》 - 2011-01-01	5.2% ( 211 ) 是否引证：否
10	基于Android的“跑在工大”软件设计 方禹 - 《大学生论文联合比对库》 - 2014-04-03	4.6% ( 186 ) 是否引证：否
11	基于Android的“跑在工大”软件设计与实现 方禹 - 《大学生论文联合比对库》 - 2014-04-09	4.6% ( 186 ) 是否引证：否
12	基于Android平台的便民缴费软件的设计与实现 章庆汉 - 《大学生论文联合比对库》 - 2015-06-08	4.6% ( 186 ) 是否引证：否
13	Android开发者指南-播放媒体-Media Playback[原创译文]_呆呆大虾 - 《网络 ( <a href="http://blog.sina.com">http://blog.sina.com</a> ) 》 - 2013	4.4% ( 177 ) 是否引证：否
14	基于Android平台的语音控制系统的设计与实现 邓阳(导师：闻英友) - 《东北大学博士论文》 - 2014-06-01	4.1% ( 167 ) 是否引证：否
15	基于人体传感网的穿戴式远程健康监护系统的设计与实现 邱明林(导师：叶学松) - 《浙江大学博士论文》 - 2013-01-25	3.0% ( 120 ) 是否引证：否
16	Android MediaPlayer使用注意_博客SumTung - 《网络 ( <a href="http://blog.sina.com">http://blog.sina.com</a> ) 》 - 2013	3.0% ( 120 ) 是否引证：否
17	基于Android的智能手持终端在航空制造中的应用 郭娜;- 《现代工业经济和信息化》 - 2016-07-30	2.9% ( 118 ) 是否引证：否
18	j基于微信的图书馆图书采购需求获取应用 任灯明 - 《大学生论文联合比对库》 - 2015-05-07	2.9% ( 116 ) 是否引证：否
19	基于一对多税控处理模型的研究	2.6% ( 103 )

赵静伟(导师：于书举) - 《北京工业大学硕士论文》 - 2009-04-01		是否引证：否
20	51081118_秦刚_电子信息工程_基于ARM Cortex-A8平台的嵌入式系统设计 Cortex - 《大学生论文联合比对库》 - 2012-06-05	2.5% ( 102 ) 是否引证：否
21	51081118_秦刚_电子信息工程_基于ARM Cortex-A8平台的嵌入式系统设计 Cortex - 《大学生论文联合比对库》 - 2012-06-08	2.5% ( 102 ) 是否引证：否
22	浅谈SQLite数据库技术在Android平台的应用 唐磊; - 《电子世界》 - 2014-05-15	2.3% ( 91 ) 是否引证：否
23	基于嵌入式Linux的休渔期渔船管理系统设计 谭业超(导师：杨尚明) - 《烟台大学博士论文》 - 2013-04-01	2.2% ( 88 ) 是否引证：否
24	基于J2EE的校园移动GIS的设计与实现 马海兵;蔡永香;黄智伟;黄乐乐;陈浩;万莉华; - 《城市勘测》 - 2014-12-31	2.0% ( 81 ) 是否引证：否
25	基于安卓的某市教育考试院移动APP的研究与应用 曹家祺(导师：王宜怀) - 《苏州大学博士论文》 - 2014-10-01	1.9% ( 76 ) 是否引证：否
26	查重5--应用 - 《大学生论文联合比对库》 - 2013-04-02	1.7% ( 70 ) 是否引证：否
27	基于决策树的低功耗手机活动识别系统的设计与实现 刘恬恬(导师：饶泓) - 《南昌大学博士论文》 - 2016-05-23	1.7% ( 70 ) 是否引证：否
28	基于Android的移动公交信息服务系统的设计 刘蕾(导师：陈佳) - 《大连海事大学博士论文》 - 2014-12-01	1.5% ( 60 ) 是否引证：否
29	基于Android的益智题游戏的开发 李志杰 - 《大学生论文联合比对库》 - 2012-07-07	1.4% ( 55 ) 是否引证：否
30	基于Android智能手机系统平台的新闻接收客户端的设计与实现 阙锋; - 《计算机与现代化》 - 2012-04-15	1.2% ( 49 ) 是否引证：否
31	基于开源软件的水井调测仿真系统的设计和实现 王欣玉;张洪江; - 《硅谷》 - 2013-03-08	1.2% ( 47 ) 是否引证：否
32	基于无线传感器网络的智能家居监控平台设计 龚婉婉;刘士兴;冯婷婷; - 《信息安全》 - 2015-03-10	1.1% ( 46 ) 是否引证：否
33	基于Android移动终端平台的LBS应用开发 何凯 - 《大学生论文联合比对库》 - 2015-06-10	1.1% ( 43 ) 是否引证：否
34	基于Android的嵌入式Web服务器设计 刘伟江;李振汉;唐余亮;黄联芬; - 《电子设计工程》 - 2013-05-05	0.8% ( 33 ) 是否引证：否

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

## 第2章技术准备

### 2.1 技术概览

#### 2.1.1 Android 系统结构

##### 图2-1 Android系统结构

Android OS是基于Linux内核的开源移动平台，其底层建立在Linux内核上，采用软件叠层(Software Stack)的方式来构建上层。本质就是在Linux内核上增加了Java虚拟机，并在虚拟机上搭建了一个Java Application Framework，所有的应用程序都运行在Java Application Framework之上。Android Runtime由核心库和Dalvik/ART虚拟机组成。核心库提供了Java核心库的绝大部分功能，而虚拟机则负责运行应用程序。每个应用程序都运行在Dalvik/ART虚拟机单独的一个实例上，这个实例驻留在一个由Linux内核管理的进程中。

#### 2.1.2 Android开发环境

Android的上层应用程序是用Java语言开发的，早期主要借助于Eclipse集成开发环境，在Eclipse上安装Android ADT插件。现在Google推荐使用自家的基于IntelliJ IDEA的集成开发环境Android Studio。要使用Java语言开发Android应用程序，还应该安装Oracle提供的Java SDK(软件开发工具集)和Google提供的Android SDK。要使应用程序能运行在Android 5.1上，JDK要求在7版或更高，Android SDK要求API级别22或以上。

#### 2.1.3 Android应用程序特点

1. 功能强大。搭载Android OS的设备一般都配备运算能力较强的CPU、各种各样的传感器、较大的存储空间、支持移动网络和WiFi网络的基带。硬件强大，软件才有发挥空间。
2. 可复用性。Android应用程序界面由许多Activity组成，不同的Activity在程序中实现不同的功能模块，不同应用程序之间的Activity可以共享(可以通过安全选项进行配置)。
3. 开放度高。“开放”是Android OS的设计宗旨之一，一些核心功能都可以由第三方案程序取代，开发者可以自行设计诸如拨号、通讯录、短信、主屏幕此类的程序。
4. 跨平台性。Android应用程序运行在Java虚拟机上。Andy Rubin在Android OS设计之初，就是看中了Java语言跨平台

的特性。几乎任何Android应用程序都能适配所有的Android设备，不局限于手机和平板电脑，甚至能同时支持手环、电视和智能家居。

## 2.2 应用程序框架

### 2.2.1 Android应用程序组件

#### 1. Activity活动

Activity是一段实时可执行代码，需要运行时即被实体化。一个应用程序允许有多个Activity。一个Activity提供一个单独的屏幕，即一个人机交互界面，上面显示一些控件，同时监听用户操作并做出响应。Activity之间通过意图(Intent)进行通信。当Activity不活跃时，系统可以结束它来节省内存。

Activity starts

onCreate()

用户使用导航回到该Activity

onRestart()onStart()

Process is killedonResume()

Activity is running

该Activity到达前景

另一Activity到达该Activity前端

该Activity到达前景其他应用程序需要内存

onPause()

该Activity不再可见

onStop()

onDestroy()

Activity is shut down

图2-2 Activity生命周期

#### 2. Service服务

Service是一种长生命周期、无可视化界面、运行于背景的程序。

音乐背景播放是一个Service的典型例子，此程序需要一直执行，即使用户切换到其他Activity或其他应用程序，甚至可以根本没有用户界面。

#### 3. Broadcast Receiver广播接收器

系统在产生某个事件时发送广播。Broadcast Receiver 用于接收并响应来自其他应用程序或系统自身的广播。一个广播接收器响应一个事件。这些消息有时被称为意图(Intent)。

Intent是一个简单的消息传递框架，负责对应用程序中一次操作的动作、动作涉及数据、附加数据进行描述。使用Intent，可以在系统范围内向目标Activity或服务广播消息，以说明希望执行某个动作的意图。之后系统就会确定那些最适合执行动作的目标。

#### 4. Content Provider内容提供者

在Android OS中，没有一个公共的内存区域，供多个应用共享存储数据。Content Provider 是应用程序之间共享数据的接口。

### 2.2.2 Android Manifest

#### 1. 组件注册

每个Activity、Service、Content Provider都需要在AndroidManifest.xml配置文件中声明。未声明的Activity、Service、Content Provider将被系统忽略。而Broadcast Receiver的注册分静态注册(在AndroidManifest.xml文件中声明)和通过代码动态创建并以调用Context.registerReceiver()的方式注册至系统。在AndroidManifest.xml文件中配置的Broadcast Receiver会随系统的启动而一直处于活跃状态，只要接收到相应的广播就会触发(即使程序未运行)。

#### 2. 权限控制

每个Android应用程序都有一个名为AndroidManifest.xml的程序清单文件，其中定义了该应用程序(对于系统来说)一些非常重要的信息。Android OS需要这些信息才能正常运行该应用。其中描述了应用程序使用某些受保护的API或者与其它应用程序交互所需的权限，也描述了其它应用程序和该应用程序交互时应拥有的权限。

Android Manifest包含Manifest.permission和Manifest.permission\_group的常量。

Manifest.permission的常量主要有：CLEAR\_APP\_CACHE允许应用程序清除所有安装在设备上的应用程序的缓存、DELETE\_CACHE\_FILES允许应用程序删除缓存文件、DELETE\_PACKAGES允许应用程序删除程序包等。

Manifest.permission\_group的常量主要有：HARDWARE\_CONTROLS被用来提供直接访问硬件设备的权限、STORAGE是与SD卡访问相关的权限组、SYSTEM\_TOOLS是与系统API有关的权限组等。

### 2.2.3 SQLite数据库

SQLite是一种轻型的嵌入式数据库，是遵守ACID的关联式数据库管理系统，包含于一个相对小的C库中。SQLite的处理

速度很快，它的设计目标是嵌入式的，目前已经在很多嵌入式产品中被使用，占用资源非常的低，可能只需要几百KB的内存就够了。Android OS在系统库中已经内置了SQLite。

Android OS通过SQLiteOpenHelper类来实现对SQLite数据库的操作。SQLiteOpenHelper封装了数据库的创建和版本维护功能，程序员也可以通过继承此类、改写其接口的方法来实现对数据库的操作。Android OS提供了一个名为SQLiteDatabase的类，SQLiteOpenHelper类中的 getWritableDatabase()和getReadableDatabase()方法返回这个类的对象。SQLiteDatabase类封装了一些操作数据库的API，使用该类可以完成对数据进行添加(Create)、查询(Retrieve)、更新(Update)和删除>Delete)的操作。还有两个类定义了ContentValues和Cursor这两种数据结构。ContentValues类似于C++中的map(String key, Integer value)，用来存放每个表的数据段及其对应的值。在Android OS中查询数据是通过Cursor类来实现的，Cursor就是查询结果的记录。

2.3 音频文件处理

音频文件格式很多，除了WAV格式是没有压缩的，其余都属于压缩存储格式，例如MP3格式。Android OS的MediaPlayer类集成了MP3音频解码器，该类直接继承类java.lang.Object，借助MediaPlayer类的对象播放MP3文件只需调用播放控制方法。依次调用reset方法复位MediaPlayer对象、setDataSource方法获得音频文件、prepare方法准备播放、start方法开始播放。

Android多媒体框架包括对多种常见媒体类型的播放支持，这样程序员就可以轻松地将音频、视频和图像集成到应用程序中。程序员可以从存储在应用程序资源(res)中的媒体文件、文件系统中的独立文件或到达网络连接的数据流播放音频或视频，所有这些都使用MediaPlayer API。

下面介绍如何编写用户与系统交互的媒体播放应用程序，以获得优秀的性能和良好的用户体验。注意，只能将音频数据播放到标准输出设备，目前来说就是移动设备扬声器或蓝牙耳机，不能在通话中播放音频文件。

- 下面的类用于在Android框架中播放声音和视频:
- MediaPlayer 这个类是播放声音和视频的主要API.
- AudioManager 这个类管理设备上的音频源和音频输出。

指 标
疑似剽窃文字表述
<div><div>1. 本质就是在Linux内核上增加了Java虚拟机，并在虚拟机上搭建了一个Java Application Framework，所有的应用程序都</div><div>2. 虚拟机则负责运行应用程序。每个应用程序都运行在Dalvik/ART虚拟机单独的一个实例上，这个实例驻留在一个由Linux内核管理的进程中。</div><div>2.1.2 Android</div><div>3. 一个应用程序允许有多个Activity。一个Activity提供一个单独的屏幕，</div><div>4. 使用Intent，可以在系统范围内向目标Activity或服务广播消息，以说明希望执行某个动作的意图。之后系统就会确定那些最适合执行动作的目标。</div><div>4. Content Provider内容提供</div><div>5. Activity、Service、Content Provider将被系统忽略。</div><div>6. 应用程序交互时应拥有的权限。</div><div>Android Manifest包含Manifest.permission和Manifest.permission_group的常量。</div><div>Manifest.permission的常量主要有：CLEAR_APP_CACHE允许应用程序清除所有安装在设备上的应用程序的缓存、DELETE_CACHE_FILES允许应用程序删除缓存文件、DELETE_PACKAGES允许应用程序删除程序包等。</div><div>Manifest.permission_group的常量主要有：HARDWARE_CONTROLS被用来提供直接访问硬件设备的权限、STORAGE是与SD卡访问相关的权限组、SYSTEM_TOOLS是与系统API有关的权限组等。</div><div>7. SQLite数据库</div><div>SQLite是一种轻型的嵌入式数据库，是遵守ACID的关联式数据库管理系统，包含于一个相对小的C库中。</div><div>8. 它的设计目标是嵌入式的，目前已经在很多嵌入式产品中被使用，占用资源非常的低，可能只需要几百KB的内存就够了。</div><div>9. SQLiteOpenHelper类中的 getWritableDatabase()和getReadableDatabase()方法返回这个类的对象。</div><div>10. Android多媒体框架包括对多种常见媒体类型的播放支持，这样程序员就可以轻松地将音频、视频和图像集成</div><div>11. 注意，只能将音频数据播放到标准输出设备，目前来说就是移动设备扬声器或蓝牙耳机，不能在通话中播放音频文件。</div><div>下面的类用于在Android框架中播放声音和视频:</div><div>MediaPlayer 这个类是播放声音和视频的主要API.</div><div>AudioManager 这个类管理设备上的音频源和音频输出。</div></div>



相似文献列表 文字复制比：1.1%(95) 疑似剽窃观点：(0)		
1	信息学院+B1102+0911110225+龙增东 - 《大学生论文联合比对库》- 2015-05-25	0.7% ( 62 ) 是否引证：否
2	2013180095-闫哲-在线音乐网站的设计与实现 闫哲 - 《大学生论文联合比对库》- 2017-05-05	0.3% ( 30 ) 是否引证：否
原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容		

第3章程序设计

3.1 分析

3.1.1 可行性分析

在Windows 10系统的PC上，搭建集成开发环境，Android Studio配合Jdk和Android SDK，使用Java语言编码，在Android虚拟机上进行调试，在Android手机上进行测试。软硬件都满足开发需求。

Android手机的音乐播放器普遍具有以下基础功能：

1. 播放、暂停、停止；
2. 跳到队列中的上/下一曲；
3. 程序处于背景仍能播放；
4. 以进度条表示播放进度；

这类Android音乐播放器很常见，此课题可行。

3.1.2 需求分析

软件开发流程的首要步骤就是需求分析，这是一个较为困难的过程，如果未能准确了解用户需求，那么设计出来的成果可能不是用户想得到的；如果用户的需求是无法实现的或者是超出能力范围，最后又如何交付呢？对本人来说则更甚，因为在选择开发一个Android音乐播放器之前，我对Android开发一无所知，仅仅是Android手机的一个普通用户而已。当我试图去做一个简单的示例工程时，我才意识到这是一个多么复杂的事情。对于新手来说，解决开发环境的配置已经是噩梦了，Android Studio的下载，SDK的选择，JDK的配置，还有Gradle的配置。

Android OS的Hello World程序可能是最简单的，因为不需要程序员输入任何代码；也可能是最难的，因为初学者要建立并生成一个示例工程很麻烦。在做完这些之后，我再来想我能做出什么呢？我心里没底，作为一个喜欢听歌的人，我知道我需要什么；作为一个菜鸟开发者，我不知道自己能做到哪些？在紧张的几个月时间里，要做出一个Android音乐播放器，对其功能上的策划必须慎重。初步的计划是，基本的播放、暂停、跳转、背景执行、图片显示等。可选的是歌词，使用内置的音乐家标签进行分类等。

根据预期想要达到的目标，对产品的整体进行设想，提出音乐播放器的基本要求，描述其功能需求。

音乐播放器首先扫描本地文件，根据内置标签构建播放列表，并对多种音频文件格式的提供支持。

1. 从SD卡读取文件

本音乐播放器从Android手机的SD卡中扫描音频文件，需要用户提前将自己喜欢的音乐下载好，并存放在手机的SD卡上，以便使用此音乐播放器随时聆听。

2. 读取MP3标签

ID3 tag，一般是位于一个mp3文件的开头或者末尾的若干字节内，用以说明该音频的相关信息，如歌曲名称、艺术家、专辑名称、作词、作曲、年代、风格等信息。ID3 tag分为v1和v2这两个版本。这里只说v2版，tag信息位于mp3文件的开头，除了可以存储前述歌曲名称等信息之外，还可以存储专辑图片、歌词等占用空间较大的信息。

3. 对常见音频格式的支持

音频文件最基本的格式是WAVE，将模拟信号转换为数字信号，以二进制形式存储起来。这些二进制数字信号在播放时，转换为模拟的电信号，被送到扬声器从而发出声音。其他的音频格式是对WAVE文件进行有损或无损的压缩来实现的。所以播放这些音频文件时，还要先对压缩之后的数据进行解码。

WAV(波浪，Wave)是微软公司推出的一种无损音频格式，准确的说是一种封装技术，其本质上的数据格式为PCM。直接从音乐CD上抓取的音轨，通常就是做成这种格式的文件。

MP3(MPEG-1 Audio Layer 3)格式是最常见的一种有损音频压缩格式，几乎成为了音乐代名词。MP3格式是一种比较古老的技术，现在有被淘汰的趋势。

WMA(Windows Media Audio)同样是微软公司推出的，是一种有损音频压缩格式，压缩率和音质均超过了MP3格式，在采样频率较低时也可以有较好的音质。

OGG(Ogg Vorbis)是一种类似于MP3的音频压缩格式，不同的是，它是完全免费的、开源的，没有任何专利限制，因此近年来得到了一定程度的推广。

FLAC(Free Lossless Audio Codec)是一种很流行的无损音频压缩格式，它也是免费的、开源的。与有损压缩编码不同，无损压缩编码不会对原有的采样信息造成损失，还可以原原本本的还原成WAVE格式的文件。

AAC(高级音频编码技术，Advanced Audio Coding)是多家公司(AT&T、Sony、Fraunhofer IIS、杜比实验室等)为了取代MP3格式，而共同开发的音频压缩编码技术，压缩率和音质都很高。AAC音频文件有.aac .mp4 .m4a等三种扩展名。

#### 4. 播放控制

最基本的播放控制，播放、暂停、跳转，是音乐播放器首要的功能。在交互界面的设计上，播放/暂停按钮设计成一个复用按钮，要放在最醒目的位置，以使用户快速进行体验。上一曲、下一曲在预定的播放队列中上下跳转，按照以往约定俗成的习惯，上一曲和下一曲按钮则分别放在“播放/暂停”左右两边。这是主要的按钮。

次要的如单曲循环/列表循环的按钮，顺序播放/随机播放的按钮，用来在播放音乐时，调整播放的顺序。这两个按钮就可以稍微小一点，应该另起一行，放在左右两侧，或者一起放在左侧或右侧。

另外，还有音频进度条，以显示当前音乐播放的进度，拖动它还可以自由调控播放进度。

#### 5. 显示歌曲信息

除了播放控制之外，播放界面还应该显示出歌曲相关信息，如歌曲名称、艺术家、专辑图片等。

##### 3.1.3 开发环境搭建

笔记本电脑Windows 10操作系统，在Android软件开发的IDE选择上，我在IntelliJ IDEA与Android Studio 的2.x和3.x之间折腾了好长时间，不是这里有问题，就是那里有bug。我最终选择了Android Studio 2.x的最后一个版本 2.3.3，下载安装了Bundle版，其中绑定了API级别为25的Android SDK。由于我事先安装了JDK SE 1.8.0u162，我删掉了Android Studio自带的Open JDK。

Android模拟器，我选择了Genymotion 2.5.3，配合Oracle VM VirtualBox 4.3.40，下载了Nexus 4 with Android 5.1的模板，模拟出Android 5.1的运行环境。

测试环境，Android OS版本为5.1的Moto G 手机。

#### 3.2 设计

##### 3.2.1 交互事件

结合UML用例图进行图形化分析，如下图所示，总述几个基本的交互事件：

图3-1 用例图

又如下表3-1到表3-16所示，详细分析各个用例的事件流，理解整个音乐播放器结构。

表3-1 歌曲播放一类事件

**用例名称：**歌曲播放一类参与者：音乐播放器的使用者

**简要描述：**音乐播放器的使用者可选择预先存储在SD卡中并被扫描到的音乐播放器的任意音乐文件进行播放

**前提条件：**音乐播放器的主界面或队列或列表的Activity处于活跃状态，并且有音频文件被音乐播放器扫描到

**主要事件流：**1. 使用者进入到音乐播放器的主界面2. 使用者打开播放队列或音乐标签分组或歌曲列表3. 使用者选中想要播放的音频条目4. 音乐播放器从主界面转到播放界面5. 音乐播放器开始播放被选中的音乐条目6. 播放界面显示该音频的信息如专辑图片、歌曲名称、演唱者、歌曲时长，以及播放进度

**异常事件流：**1. 音乐播放器发出错误提示，音频播放失败，然后，音乐播放器会自动尝试相应播放队列中的下一曲2. 被使用者选中的条目被音乐播放器成功播放，但是未能正确显示歌曲的信息和播放进度3. 使用者点击某个条目之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity

**后置条件：**音频播放成功，播放界面正确显示出歌曲信息和播放进度

表3-2 暂停播放事件

**用例名称：**暂停播放参与者：音乐播放器的使用者

**简要描述：**音乐播放器的使用者对正在播放的音频进行暂停的操作

**前提条件：**音乐播放器的播放界面的Activity处于活跃状态，并且有音频文件正在被播放

**主要事件流：**1. 使用者进入到音乐播放器的播放界面2. 使用者点击一次“播放/暂停”音频播放状态切换按钮3. 音乐播放器暂停播放正在被播放的音频文件4. 播放界面的“播放/暂停”音频播放状态切换按钮，由“暂停”样式变为“播放”样式

**异常事件流：**1. 在当前音频正处于播放状态时，点击了一次“播放/暂停”音频播放状态切换按钮之后，音频仍然正常播放2. 在当前音频正处于播放状态时，点击了一次“播放/暂停”音频播放状态切换按钮之后，它并没有由“暂停”样式变为“播放”样式3. 在当前音频正处于播放状态时，点击了一次“播放/暂停”音频播放状态切换按钮之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity

**后置条件：**音频暂停成功，播放界面的“播放/暂停”音频播放状态切换按钮由“暂停”样式变为“播放”样式

表3-3 歌曲播放二类事件

**用例名称：**歌曲播放二类参与者：音乐播放器的使用者

**简要描述：**音频正处于还没有开始播放的状态或者已经暂停的状态，音乐播放器的使用者点击“播放/暂停”音频播放状态切换按钮，使音频文件开始播放或者继续播放

**前提条件：**音乐播放器的播放界面的Activity处于活跃状态，并且有音频文件正准备被播放或者已经被暂停播放

**主要事件流：**1. 使用者进入到音乐播放器的播放界面2. 使用者点击一次播放界面的“播放/暂停”音频播放状态切换按钮3. 音乐播放器开始播放或者继续在原来位置播放该音频文件4. 播放界面的“播放/暂停”音频播放状态切换按钮，从“播放”样式变为“暂停”样式

**异常事件流：**1. 使用者点击一次“播放/暂停”音频播放状态切换按钮，音乐播放器发出错误提示，音频播放失败，然后

，音乐播放器自动尝试播放队列中的下一曲2. 在音频正处于暂停播放状态时，使用者点击了一次“播放/暂停”音频播放状态切换按钮，该音频却没有从暂停的位置继续播放，而是从头开始播放3. 在音频正处于未播放状态时，使用者点击了一次“播放/暂停”音频播放状态切换按钮之后，它并没有由“播放”式样变为“暂停”式样4. 使用者点击了“播放/暂停”音频播放状态切换按钮，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity

后置条件：音频播放成功，播放界面的“播放/暂停”音频播放状态切换按钮由“播放”式样变为“暂停”式样

表3-4 切换上/下一曲事件

用例名称：切换上/下一曲参与者：音乐播放器的使用者

简要描述：在音乐播放器的播放界面，使用者对播放队列中的音频进行上一曲或者下一曲的动态切换

前提条件：音乐播放器的播放界面的Activity处于活跃状态，音频正处于播放或准备播放或暂停播放某个音频的状态

主要事件流：1. 使用者进入到音乐播放器的播放界面2. 使用者点击一次播放界面的“上一曲”或者“下一曲”音频切换按钮3. 音频被切换到队列中的“上一曲”或者“下一曲”4. 播放界面的歌曲信息如专辑图片、歌曲名称、演唱者、歌曲时长，发生相应的变化

异常事件流：1. 使用者点击了一次“上一曲”或者“下一曲”音频切换按钮之后，播放界面仍然处于播放或准备播放或暂停播放当前音频的状态2. 使用者点击了一次“上一曲”或者“下一曲”音频切换按钮之后，音频跳转到了并非是队列中的上一曲或者下一曲的其他音频3. 在使用者将歌曲成功切换之后，相应的歌曲信息没有切换，或者专辑图片、歌曲名称、演唱者、歌曲时长，某一条或几条没有正确显示4. 使用者点击了“上一曲”或者“下一曲”音频切换按钮之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity

后置条件：音频切换上/下一曲成功，歌曲信息如专辑图片、歌曲名称、演唱者、歌曲时长正确的被切换

表3-5 单曲/列表循环事件

用例名称：单曲/列表循环参与者：音乐播放器的使用者

简要描述：在音乐播放器的播放界面，使用者对播放队列中的音频进行单曲循环或列表循环或不循环的播放模式切换

前提条件：音乐播放器的播放界面的Activity处于活跃状态，音频正处于播放或准备播放或暂停播放的状态

主要事件流：1. 使用者进入到音乐播放器的播放界面2. 使用者点击播放界面的“单曲/列表循环”播放模式切换按钮之后，该按钮的式样不停在“单曲循环”、“列表循环”、“不循环”之间来回变化3. 当前音频播放结束之后，音乐播放器如期按照使用者切换之后的播放模式在播放队列中进行跳转，且与“单曲/列表循环”播放模式切换按钮的式样始终保持一致4. 点击“上一曲”或者“下一曲”音频切换按钮之后，音乐播放器如期按照使用者切换之后的播放模式在播放队列中进行跳转，且与“单曲/列表循环”播放模式切换按钮的式样始终保持一致

异常事件流：1. 使用者点击了“单曲/列表循环”播放模式切换按钮之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 使用者点击了“单曲/列表循环”播放模式切换按钮之后，该按钮的式样并没有在“单曲循环”、“列表循环”、“不循环”之间来回变化3. 使用者点击了“单曲/列表循环”播放模式切换按钮之后，当前音频播放结束之后，或者点击一次“上一曲”或“下一曲”音频切换按钮之后，音乐播放器的音频跳转模式并没有相应改变4. 使用者点击了“单曲/列表循环”播放模式切换按钮之后，当前音频播放结束之后，或者点击一次“上一曲”或“下一曲”音频切换按钮之后，音乐播放器的音频跳转模式与“单曲/列表循环”播放模式切换按钮所显示的不同

后置条件：音频切换“单曲/列表循环”播放模式成功，该按钮的式样在“单曲循环”、“列表循环”、“不循环”之间来回变化，并且与音乐播放器实际的播放模式相符合

表3-6 随机/顺序播放事件

用例名称：随机/顺序播放参与者：音乐播放器的使用者

简要描述：在音乐播放器的播放界面，使用者对播放队列中的音频进行随机播放或者顺序播放的跳转模式转换

前提条件：音乐播放器的播放界面的Activity处于活跃状态，音频正处于播放或准备播放或暂停播放的状态

主要事件流：1. 使用者进入到音乐播放器的播放界面2. 使用者点击播放界面的“随机/顺序”跳转模式切换按钮之后，该按钮的式样不停在“随机播放”与“非随机播放”之间来回变化3. 当前音频播放结束之后，音乐播放器如期按照使用者切换之后的跳转模式在播放队列中进行跳转，且与“随机/顺序”跳转模式切换按钮的式样始终保持一致4. 点击“上一曲”或者“下一曲”音频切换按钮之后，音乐播放器如期按照使用者切换之后的跳转模式在播放队列中进行跳转，且与“随机/顺序”跳转模式切换按钮的式样始终保持一致

异常事件流：1. 使用者点击了“随机/顺序”跳转模式切换按钮之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 使用者点击了“随机/顺序”跳转模式切换按钮之后，该按钮的式样并没有在“随机播放”与“非随机播放”之间来回变化3. 使用者点击了“随机/顺序”跳转模式切换按钮之后，当前音频播放结束之后，或者点击一次“上一曲”或“下一曲”音频切换按钮之后，音乐播放器的音频跳转模式并没有相应改变4. 使用者点击了“随机/顺序”跳转模式切换按钮之后，当前音频播放结束之后，或者点击一次“上一曲”或“下一曲”音频切换按钮之后，音乐播放器的音频跳转模式与“随机/顺序”跳转模式切换按钮所显示的不同

后置条件：音频切换“随机/顺序”跳转模式成功，该按钮的式样在“随机播放”与“非随机播放”之间来回变化，并且与音乐播放器实际的跳转模式相符合

表3-7 拖动播放进度条事件



用例名称：拖动播放进度条参与者：音乐播放器的使用者

简要描述：在音乐播放器的播放界面，使用者可以非常直观的看到音频播放的进度和具体的时间点，并且可以自由的拖动播放进度条，来调整该音频播放的进度

前提条件：音乐播放器的播放界面的Activity处于活跃状态，音频正处于播放或准备播放或暂停播放的状态

主要事件流：1. 使用者进入到音乐播放器的播放界面2. 使用者前后拖动播放界面的音频进度条之后，播放界面的时间信息相应发生改变3. 使用者前后拖动播放界面的音频进度条之后，如果当前音频正在播放状态，音乐播放器就跳到进度条指定的时间位置继续播放；如果当前音频正处于准备播放或者暂停播放的状态，音乐播放器就准备在所跳到的进度条指定的时间位置播放4. 音乐播放器播放音频的时间点与进度条所示的时间位置时刻保持一致

异常事件流：1. 使用者或前或后拖动了播放界面的音频进度条之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 使用者或前或后拖动了播放界面的音频进度条之后，音频未能如期实现时间点跳跃，或者点击一次“播放/暂停”音频播放状态切换按钮后，仍然在原来的时间点开始或者继续播放3. 使用者或前或后拖动了播放界面的音频进度条之后，音频实现了播放位置的跳跃，但是与预期的时间点不符合4. 使用者或前或后拖动了播放界面的音频进度条之后，音频实现了播放位置的跳跃，但是文字所显示的时间点未能及时更新

后置条件：音频播放的进度跳跃成功，文字所显示的时间信息成功改变，且进度条位置、音频播放的实际时间点、文字所显示的时间，三者始终保持一致

表3-8 查看所有歌曲事件

用例名称：查看所有歌曲参与者：音乐播放器的使用者

简要描述：在音乐播放器的主界面，使用者可以非常直观的查看到所有音乐的名单，其中列出了歌曲名称、演唱者、内置专辑图片和对该音频进行操作的菜单按钮

前提条件：音乐播放器的主界面的Activity处于活跃状态

主要事件流：1. 使用者进入到音乐播放器的主界面2. 使用者点击音乐播放器主界面的“歌曲”标签，转到“歌曲”页面3. 音乐播放器列出所有的歌曲信息，包括歌曲名称、演唱者、内置专辑图片，如果音频文件没有该歌曲的某一条信息，就显示空白4. 使用者上下滑动屏幕来浏览所有的歌曲信息

异常事件流：1. 点击音乐播放器主界面的“歌曲”标签之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 点击音乐播放器主界面的“歌曲”标签之后，转到了“歌曲”页面，但是音乐播放器没有列出所有的歌曲信息3. 点击音乐播放器主界面的“歌曲”标签之后，转到了“歌曲”页面，音乐播放器列出了一些歌曲信息，但是所列出的歌曲信息不全，或者与预期本该列出的歌曲信息不符合，或者张冠李戴产生混乱4. 点击音乐播放器主界面的“歌曲”标签之后，转到了“歌曲”页面，音乐播放器列出了所有的歌曲信息，但是不能滑动屏幕来上下浏览

后置条件：所有歌曲的相关信息成功的准确无误的列出，并且可以滑动屏幕自由浏览

表3-9 查看所有专辑事件

用例名称：查看所有专辑参与者：音乐播放器的使用者

简要描述：在音乐播放器的主界面，使用者可以非常直观的查看到所有专辑的名单，其中列出了专辑图片、专辑名称、艺术家

前提条件：音乐播放器的主界面的Activity处于活跃状态

主要事件流：1. 使用者进入到音乐播放器的主界面2. 使用者点击音乐播放器主界面的“专辑”标签，转到“专辑”页面3. 音乐播放器列出所有的专辑信息，包括专辑图片、专辑名称、专辑艺术家，如果音频文件没有该专辑的某一条信息，就显示空白4. 使用者上下滑动屏幕来浏览所有的专辑信息

异常事件流：1. 点击音乐播放器主界面的“专辑”标签之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 点击音乐播放器主界面的“专辑”标签之后，转到了“专辑”页面，但是音乐播放器没有列出所有的专辑信息3. 点击音乐播放器主界面的“专辑”标签之后，转到了“专辑”页面，音乐播放器列出了一些专辑信息，但是所列出的专辑信息不全，或者与预期本该列出的专辑信息不符合，或者张冠李戴产生混乱4. 点击音乐播放器主界面的“专辑”标签之后，转到了“专辑”页面，音乐播放器列出了所有的专辑信息，但是不能滑动屏幕来上下浏览

后置条件：所有专辑的相关信息成功的准确无误的列出，并且可以滑动屏幕自由浏览

表3-10 查看专辑曲目事件

用例名称：查看专辑曲目参与者：音乐播放器的使用者

简要描述：在音乐播放器主界面的“专辑”标签，使用者点击所查看到的所有专辑中的任意一项，可以转到该专辑的页面，其中将显示专辑图片、专辑名称，并列出该专辑中所有曲目

前提条件：音乐播放器的主界面的Activity处于活跃状态

主要事件流：1. 使用者进入到音乐播放器的主界面，点击“专辑”标签，查看所有的专辑的列表2. 使用者点击所查看的专辑列表中的任意一项3. 音乐播放器转到该专辑的页面，列出该专辑的信息，包括专辑图片、专辑名称、所有专辑曲目，如果没有图片，就显示空白4. 使用者上下滑动屏幕来浏览所有的曲目信息

异常事件流：1. 使用者点击音乐播放器主界面的“专辑”标签，并选中所显示专辑列表中的任意一项之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 使用者点击音乐播放器主界面的“专辑”标签，并选中所显示专辑列表中的



任意一项之后，音乐播放器转到了该专辑的页面，但是该专辑的相关信息没有被列出3. 音乐播放器列出了一些该专辑的信息，但是所列出的信息不全，或者与预期本该列出的专辑信息不符合，或者张冠李戴产生混乱4. 音乐播放器列出了该专辑所有的信息，但是不能滑动屏幕来上下浏览

后置条件：所有该专辑的相关信息成功的准确无误的列出，并且可以滑动屏幕自由浏览

表3-11 查看所有艺术家事件

用例名称：查看所有艺术家参与者：音乐播放器的使用者

简要描述：在音乐播放器的主界面，使用者可以非常直观的查看到所有的艺术家的名单，其中列出了艺术家姓名、艺术家相片（如果有）与该艺术家的专辑数量、歌曲数量

前提条件：音乐播放器的主界面的Activity处于活跃状态

主要事件流：1.

6. 53120801_沈建_计算机科学与技术_基于Android平台的音乐播放器设计与实现.doc_第6部分		总字数：11235
相似文献列表 文字复制比：64.4%(7236) 疑似剽窃观点：(0)		
1	Android播放媒体详解 - jimo - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	55.7% ( 6261 ) 是否引证：否
2	Android API Guides--Media Playback - 毅之忆的博客 - CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	51.6% ( 5797 ) 是否引证：否
3	Android API Guide for Media and Camera (一) —— 媒体与相机 (媒体播放器) -- CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	46.4% ( 5208 ) 是否引证：否
4	Android 媒体播放---media playback - 《互联网文档资源 ( <a href="http://wenku.baidu.c">http://wenku.baidu.c</a> ) 》 - 2016	45.0% ( 5057 ) 是否引证：否
5	Android 媒体播放---media playback - 《互联网文档资源 ( <a href="http://wenku.baidu.c">http://wenku.baidu.c</a> ) 》 - 2017	45.0% ( 5057 ) 是否引证：否
6	media and camera 框架之一： media playback - 奋起追赶 - CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	44.8% ( 5032 ) 是否引证：否
7	Android开发指南(34) —— Multimedia and Camera - Media Playback - junjiekings的专栏 - CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	42.6% ( 4789 ) 是否引证：否
8	Android 媒体播放 ( Media Playback ) - 竹叶青的专栏 - CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	42.6% ( 4787 ) 是否引证：否
9	Android -- 多媒体播放之Media Playback基础介绍 - 可爱、可信、可为 - CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	42.5% ( 4770 ) 是否引证：否
10	android Media back - citycity222的专栏 - 博客频道 - CSDN.NET - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2013	42.2% ( 4745 ) 是否引证：否
11	Android MediaPlayer Playback---多媒体开发应用程序接口 - &#183;没有风雨躲得过，没有坎坷不必走&#183; - 博客频道 - CSDN.NET - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2013	41.6% ( 4673 ) 是否引证：否
12	Android MediaPlayer Playback---多媒体开发应用程序接口 - &#183;没有风雨躲得过，没有坎坷不必走&#183; - CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	41.6% ( 4673 ) 是否引证：否
13	Android开发者指南-播放媒体-Media Playback[原创译文]_呆呆大虾 - 《网络 ( <a href="http://blog.sina.com">http://blog.sina.com</a> ) 》 - 2013	41.6% ( 4669 ) 是否引证：否
14	Media Playback - 博客频道 - CSDN.NET - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	41.1% ( 4620 ) 是否引证：否
15	Android MediaPlayer使用注意_博客SumTung - 《网络 ( <a href="http://blog.sina.com">http://blog.sina.com</a> ) 》 - 2013	40.8% ( 4584 ) 是否引证：否
16	android mediaplayer使用注意 - 博客频道 - CSDN.NET - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	40.2% ( 4521 ) 是否引证：否
17	MediaPlayer 的使用介绍 - songshizhuyuan的专栏 - CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	38.2% ( 4288 ) 是否引证：否
18	Android官方Media Playback中文教程 - HYM BLOG - CSDN博客 - 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	28.2% ( 3164 ) 是否引证：否
19	MediaPlayer和Service用法 - iblade的博客 - CSDN博客	26.3% ( 2957 )

	- 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	是否引证：否
20	Android媒体播放---Media playback ( 二 ) - FireOfStar的专栏 - 博客频道 - CSDN.NET	20.2% ( 2273 )
	- 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2013	是否引证：否
21	Android工作总结之如何做一个优秀的MediaPlayer音频播放器 - 十蛋的博客 - CSDN博客	13.7% ( 1534 )
	- 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	是否引证：否
22	API Guides ( 七 ) —— Property Animation To Media Playback - lneartao的博客 - CSDN博客	13.5% ( 1518 )
	- 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	是否引证：否
23	Android多媒体编程——MediaPlayer播放音乐 - 博客频道 - CSDN.NET	13.3% ( 1492 )
	- 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2017	是否引证：否
24	android 多媒体和相机详解四 - nkmmk的专栏 - 博客频道 - CSDN.NET	12.6% ( 1421 )
	- 《网络 ( <a href="http://blog.csdn.net">http://blog.csdn.net</a> ) 》 - 2012	是否引证：否
25	赵志浩-10024221-Android系统下的全景视频播放器研究 王以忠	10.5% ( 1182 )
	赵志浩 - 《大学生论文联合比对库》 - 2014-06-09	是否引证：否
26	信息学院+B1102+0911110225+龙增东	0.3% ( 33 )
	- 《大学生论文联合比对库》 - 2015-05-25	是否引证：否

原文内容 **红色文字**表示存在文字复制现象的内容; **绿色文字**表示其中标明了引用的内容

使用者进入到音乐播放器的主界面2. 使用者点击音乐播放器主界面的“艺术家”标签，转到“艺术家”页面3. 音乐播放器列出所有的艺术家信息，包括艺术家姓名、艺术家相片（如果有）与该艺术家的专辑数量、歌曲数量，如果没有找到该艺术家的相片，就显示默认图片4. 使用者上下滑动屏幕来浏览所有的艺术家信息

异常事件流：1. 点击音乐播放器主界面的“艺术家”标签后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 点击音乐播放器主界面的“艺术家”标签之后，转到了“艺术家”页面，但是音乐播放器没有列出所有的艺术家信息3. 点击音乐播放器主界面的“艺术家”标签之后，转到了“艺术家”页面，音乐播放器列出了一些艺术家信息，但是所列出的艺术家信息不全，或者与预期本该列出的艺术家信息不符合，或者张冠李戴产生混乱4. 点击音乐播放器主界面的“艺术家”标签之后，转到了“艺术家”页面，音乐播放器列出了所有的艺术家信息，但是不能滑动屏幕来上下浏览

后置条件：所有艺术家的相关信息成功的准确无误的列出，并且可以滑动屏幕自由浏览

表3-12 查看艺术家作品事件

用例名称：查看艺术家作品参与者：音乐播放器的使用者

简要描述：在音乐播放器主界面的“艺术家”标签，使用者点击所查看到的所有艺术家中的任意一项，可以转到该艺术家的页面，其中将显示该艺术家的姓名、所有专辑、所有歌曲

前提条件：音乐播放器的主界面的Activity处于活跃状态

主要事件流：1. 使用者进入到音乐播放器的主界面，点击“艺术家”标签，查看所有的艺术家的列表2. 使用者点击所查看的艺术家列表中的任意一项3. 音乐播放器转到该艺术家的页面，列出该艺术家的信息，包括艺术家相片、所有专辑、所有歌曲，如果没有图片，就显示空白4. 使用者上下滑动屏幕来浏览该艺术家所有的信息

异常事件流：1. 使用者点击音乐播放器主界面的“艺术家”标签，并选中所显示艺术家列表中的任意一项之后，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 使用者点击音乐播放器主界面的“艺术家”标签，并选中所显示艺术家列表中的任意一项之后，音乐播放器转到了该艺术家的页面，但是该艺术家的相关信息没有被列出3. 音乐播放器列出了一些该艺术家的信息，但是所列出的信息不全，或者与预期本该列出的艺术家信息不符合，或者张冠李戴产生混乱4. 音乐播放器列出了该艺术家所有的信息，但是不能滑动屏幕来上下浏览

后置条件：所有该艺术家的相关信息成功的准确无误的列出，并且可以滑动屏幕自由浏览

表3-13 音乐搜索事件

用例名称：音乐搜索参与者：音乐播放器的使用者

简要描述：以关键字搜索音乐库中的任意歌曲、专辑、艺术家

前提条件：音乐播放器的主界面或者播放界面Activity处于活跃状态

主要事件流：1. 使用者进入到音乐播放器的主界面或者播放界面2. 使用者点击主界面或者播放界面右上角的搜索按钮3. 转到音乐播放器的搜索界面4. 使用者在搜索框输入关键字，生成相应的搜索结果，包括所有相关的歌曲、专辑、艺术家（如果有）5. 使用者点击搜索结果，进行相应的操作

异常事件流：1. 使用者点击主界面或者播放界面右上角的搜索按钮，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 转到了搜索界面，输入关键字之后无效，或者音乐播放器立即停止工作并重新启动Activity3. 使用者在搜索框输入关键字进行搜索之后，生成的结果不全面4. 点击搜索结果，无法进行相关操作

后置条件：搜索界面转到成功，输入关键字之后，生成所有结果，且结果有效

表3-14 主界面菜单事件

用例名称：主界面菜单参与者：音乐播放器的使用者

简要描述：主界面菜单给使用者提供几个设置选项，如排序方式、均衡器等

前提条件：音乐播放器的主界面Activity处于活跃状态

主要事件流：1. 使用者进入到音乐播放器的主界面2. 使用者点击主界面右上角的菜单按钮3. 呼出主菜单的功能选项：视图模式、排序方式、均衡器4. 使用者点击其中任意选项，产生相应效果

异常事件流：1. 使用者点击主界面右上角的菜单按钮，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 呼出了菜单，点击选项无效，或者音乐播放器立即停止工作并重新启动Activity

后置条件：主界面菜单呼出成功，点击选项产生相应效果

表3-15 播放界面菜单事件

用例名称：播放界面菜单参与者：音乐播放器的使用者

简要描述：播放界面菜单包含主界面菜单选项，并且给使用者提供几个转到选项，如转到专辑、转到艺术家、显示歌词等

前提条件：音乐播放器的播放界面Activity处于活跃状态

主要事件流：1. 使用者进入到音乐播放器的播放界面2. 使用者点击播放界面右上角的菜单按钮3. 呼出主菜单的功能选项：转到专辑、转到艺术家、显示歌词4. 使用者点击其中任意选项，产生相应效果或转到相应页面

异常事件流：1. 使用者点击播放界面右上角的菜单按钮，没有任何的响应，或者音乐播放器立即停止工作并重新启动Activity2. 呼出了菜单，点击选项无效，或者音乐播放器立即停止工作并重新启动Activity

后置条件：播放界面菜单呼出成功，点击选项产生相应效果

表3-16 侧滑导航菜单事件

用例名称：侧滑导航菜单参与者：音乐播放器的使用者

简要描述：侧滑菜单给使用者提供几个导航选项，如音乐库、播放列表、播放队列等

前提条件：音乐播放器的主界面Activity处于活跃状态

主要事件流：1. 使用者进入到音乐播放器的主界面2. 使用者点击主界面左上角的菜单按钮，或者在屏幕左侧向右滑动3. 呼出侧滑菜单的导航选项：音乐库、播放列表、播放队列4. 使用者点击其中任意选项，转到相应页面

异常事件流：1. 使用者点击主界面左上角的菜单按钮，或者在屏幕左侧向右滑动，没有任何的响应或音乐播放器立即停止工作并重新启动Activity2. 呼出了侧滑菜单，点击选项无效，或者音乐播放器立即停止工作并重新启动Activity

后置条件：侧滑导航菜单呼出成功，点击选项转到相应页面

3.2.2 布局管理

Android系统采用Xml文件声明的方式，设计UI界面、添加UI控件，把程序逻辑与界面元素分隔开来。在Android Studio中，有两种方式编辑xml布局文件，一种是布局编辑器，直观的通过拖放控件的方式布局，一种是直接编辑文本添加控件声明。XML文件包含于Layout文件夹。

有如下几种布局：

帧布局(FrameLayout)是最简单的一种布局模型，主要用于图片切换，它将每个新添加的控件都放在布左上角，并覆盖在前一个控件的上层；

线性布局(LinearLayout)是最常用的一种布局方式，它将控件排列成一行多列或者一列多行的形式，依照orientation的值水平或者垂直排列，同时可以设置控件的相关参数控制各个控件在布局中的相对大小；

相对布局(RelativeLayout)的控件之间位置相关或者和他们的父控件位置相关，参照控件可以是父控件，也可以是其他子控件，但被参照的控件必须要在参照它的控件之前被定义；

表格布局(TableLayout)按照行列排序的形式管理控件，每一行是一个TableRow或者是一个View。在TableRow中添加一个个控件形成列；

坐标布局(AbsoluteLayout)直接指定控件在布局中的横、纵坐标值，否则就会像帧布局一样被排在左上角。这种布局不能自适应屏幕的尺寸，需要为不同的分辨率分别指定坐标值，应用的很少。

3.2.3 数据库设计

列表编号

列表名称

播放列表

1

组成

歌曲名称文件大小

发行时间作曲 N

歌曲

作词

流派

歌词

N N

音轨号组成属于

1 1

专辑艺术家属于

N 1

专辑发行时间

艺术家姓名

艺术家相片专辑名称专辑图片

图3-2 数据库ER图

### 3.3 编码

首先打开配置好的Android Studio，新建立一个Android工程，如下图示：

图3-3 Android Studio截图

图3-4 Android Studio截图

等待Gradle Build Tool自动进行Sync，然后再进行Build.

图3-5 Android Studio截图

打开build.gradle (Module:app)文件进行相关配置。

图3-6 Android Studio截图

然后就可以开始编码了，经过不断的编码、调试、Build，最后做出成品。

以对MediaPlayer的处理为例，进行代码分析。

#### 1. 清单声明

在开始使用MediaPlayer开发应用程序之前，确保你的清单具有允许使用相关特性的适当声明。

Internet权限 - 如果你正在使用MediaPlayer来流基于网络的内容，你的应用程序必须请求网络访问。

`<uses-permission android:name="android.permission.INTERNET" />`

唤醒锁许可 - 如果你的播放器应用程序需要防止屏幕变暗或处理器休眠，或使用

MediaPlayer.setScreenOnWhilePlaying()或MediaPlayer.setWakeMode()方法，你必须请求许可。

`<uses-permission android:name="android.permission.WAKE_LOCK" />`

#### 2. 使用MediaPlayer

媒体框架中最重要的组件之一是MediaPlayer类。这个类的一个对象可以获取、解码和播放带有最小设置的音频和视频。

它支持几种不同的媒体来源，例如：

本地资源

内部uri，例如你可能从一个内容提供者获得

外部url(流)

下面是一个如何播放本地原始资源中的音频的示例(保存在应用程序的res/raw/目录中)：

`MediaPlayer mediaPlayer = MediaPlayer.create(context, R.raw.sound_file_1);`

`mediaPlayer.start();` // 不需要调用prepare(); create()已经完成了

在这种情况下，“原始”资源是系统不尝试以任何特定方式解析的文件。但是，该资源的内容不应该是原始音频。它应该是一种被支持的格式正确编码和格式化的媒体文件。

下面是如何从系统中本地可用的URI播放(例如，通过内容提供者获得)：

`Uri myUri = ....;` // 在这里初始化Uri

`MediaPlayer mediaPlayer = new MediaPlayer();`

`mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);`

`mediaPlayer.setDataSource(getApplicationContext(), myUri);`

`mediaPlayer.prepare();`

`mediaPlayer.start();`

通过HTTP流从远程URL播放如下：

`String url = "http://.....";` // 在这里填写你的URL

`MediaPlayer mediaPlayer = new MediaPlayer();`

`mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);`

`mediaPlayer.setDataSource(url);`

`mediaPlayer.prepare();` // 进行buffering等

`mediaPlayer.start();`

注意：如果你正在传递一个URL来流一个在线媒体文件，该文件必须能够进行渐进式下载。在使用setDataSource()时，你必须捕获或传递IllegalArgumentException和IOException，因为你所引用的文件可能不存在。

#### 3. 异步准备

在原则上，使用MediaPlayer可以很简单。然而，重要的是要记住，要正确地集成一个典型的Android应用程序，还需要



做更多的事情。例如，prepare()的调用可能需要很长的时间来执行，因为它可能涉及到获取和解码媒体数据。因此，正如任何可能需要很长时间才能执行的方法一样，你不应该从应用程序的UI线程调用它。这样做会导致UI挂起直到方法返回，这是非常糟糕的用户体验，会导致ANR(应用程序没有响应)错误。即使你希望你的资源能够快速加载，记住任何需要超过十分之一的秒的时间在UI中响应，都会引起明显的停顿，并给用户留下你的应用程序很慢的印象。

为了避免挂起UI线程，生成另一个线程来准备MediaPlayer，并在完成时通知主线程。虽然你可以自己编写线程逻辑，但是在使用MediaPlayer时，这个模式非常常见，因为框架提供了一种方便的方法，可以使用prepareAsync()方法来完成这个任务。该方法开始在背景准备媒体，并立即返回。当媒体做好准备时，调用MediaPlayer.OnPreparedListener的onPrepared()方法，它是通过setOnPreparedListener()配置的。

#### 4. 管理状态

另一个你应该记住的MediaPlayer的方面是它是基于状态的。也就是说，MediaPlayer有一个内部状态，在编写代码时，你必须始终注意到它，因为某些操作只有在特定状态下才有效。如果在错误的状态下执行操作，系统可能会抛出异常或导致其他不良行为。

MediaPlayer类中的文档显示了一个完整的状态图，它阐明了哪些方法将MediaPlayer从一个状态转换到另一个状态。例如，当你创建一个新的MediaPlayer时，它处于空闲状态。在这一点上，你应该通过调用setDataSource()来初始化它，将它变为初始化状态。在此之后，你必须使用prepare()或prepareAsync()方法来准备它。当MediaPlayer完成准备时，它进入准备状态，这意味着你可以调用start()来让它播放媒体。在这一点上，正如图示，你可以通过调用start()、pause()、seekTo()等方法在开始、暂停和PlaybackCompleted状态之间转换。但是，当你调用stop()时，请注意你不能再次调用start()，直到你再次准备MediaPlayer。

在编写与MediaPlayer对象交互的代码时，始终要记住状态图，因为在错误的状态调用它的方法是bug的常见原因。

#### 5. 释放MediaPlayer

MediaPlayer可以消耗宝贵的系统资源。因此，你应该始终采取额外的预防措施，以确保你没有在必要的时候挂在MediaPlayer实例上。当你完成它时，你应该始终调用release()来确保分配给它的任何系统资源都被适当地释放。例如，如果你使用的是一个MediaPlayer和活动接收onStop()调用请求，你必须释放该MediaPlayer，因为当你的活动不与用户进行交互时(除非你是在后台播放媒体，这是在下一节中讨论)，保留它毫无意义。当然，当你的活动恢复或重新启动时，你需要创建一个新的MediaPlayer并在恢复播放之前再次准备它。

下面是你应该如何释放和取消你的MediaPlayer:

```
mediaPlayer.release();
```

```
mediaPlayer = null;
```

举例来说，如果你在停止活动时忘记释放MediaPlayer，但在活动再次启动时创建一个新的，请考虑可能发生的问题。正如你可能知道的，当用户更改屏幕的方向(或以另一种方式更改设备配置)，系统通过重新启动活动(默认情况下)来处理它，所以如果用户旋转设备，在竖向和横向之间来回切换，你可能会很快消耗掉所有系统资源，因为在每次方向变化时，你创建一个新的MediaPlayer，并且永远不会释放。(有关运行时重启的更多信息，请参见处理运行时更改。)

你可能想知道发生了什么，如果你想继续播放“后台媒体”，即使用户离开了你的活动，就像内置的音乐应用程序的行为一样。在这种情况下，你需要的是一个由服务控制的MediaPlayer，如下一节所讨论的。

#### 6. MediaPlayer用于服务

如果你希望你的媒体在背景播放，即使你的应用程序不在屏幕显示——也就是说，在用户与其他应用程序交互时，你希望它继续播放——那么你必须启动一个服务并从那里控制MediaPlayer实例。你需要把MediaPlayer嵌入MediaBrowserServiceCompat服务并让它与MediaBrowserCompat在另一个活动交互。

你应该注意这个客户机/服务器设置。这里有一些期望，关于在背景服务中运行的播放器如何与系统的其余部分进行交互。如果你的应用程序没有满足这些期望，用户可能会有糟糕的体验。请阅读构建一个音频应用程序的全部细节。

本节描述了管理一个在服务中实现的MediaPlayer的特殊指令。

#### 7. 异步运行

首先，就像一个活动一样，服务中的所有工作都默认在一个单独的线程中完成，事实上，如果你在同一个应用程序中运行一个活动和一个服务，默认情况下它们使用相同的线程(“主线程”)。因此，服务需要快速地处理传入的意图，并且在响应它们时从不执行冗长的计算。如果预期有任何繁重的工作或阻塞调用，则必须异步执行这些任务:从另一个线程中实现自己，或者使用框架的许多设备进行异步处理。

例如，当在主线程中使用MediaPlayer时，应该调用prepareAsync()而不是prepare()，并实现一个MediaPlayer.OnPreparedListener，以便在准备完成时通知你，你可以开始播放了。例如:

```
public class MyService extends Service implements
    MediaPlayer.OnPreparedListener {
    private static final String ACTION_PLAY = "com.example.action.PLAY";
    MediaPlayer mMediaPlayer = null;
    public int onStartCommand(Intent intent, int flags, int startId) {
```

```

...
if (intent.getAction().equals(ACTION_PLAY)) {
    mMediaPlayer = ... // 在这里初始化
    mMediaPlayer.setOnPreparedListener(this);
    mMediaPlayer.prepareAsync(); // prepare async 不锁定主线程
}
}
/* MediaPlayer就绪时被调用 */
public void onPrepared(MediaPlayer player) {
    player.start();
}
}

```

## 8. 处理异步错误

在同步操作中，通常会用异常或错误代码来表示错误，但是无论何时使用异步资源，都应该确保正确地通知应用程序错误。在MediaPlayer的情况下，你可以通过实现MediaPlayer.OnErrorListener来达到这一点，在你的MediaPlayer实例中设置：

```

public class MyService extends Service implements
    MediaPlayer.OnErrorListener {
    MediaPlayer mMediaPlayer;
    public void initMediaPlayer() {
        // ...在此初始化MediaPlayer...
        mMediaPlayer.setOnErrorListener(this);
    }
    @Override
    public boolean onError(MediaPlayer mp, int what, int extra) {
        // ... 做出相应的响应...
        // MediaPlayer已经转到错误状态，必须重置！
    }
}

```

重要的是要记住，当出现错误时，MediaPlayer会切换到错误状态(在MediaPlayer类的文档中查看完整状态图)，你必须在再次使用它之前重新设置它。

## 9. 使用唤醒锁

在设计在背景播放媒体的应用程序时，当你的服务运行时，设备可能会休眠。由于Android系统试图在设备休眠时节省电量，系统试图关闭手机的任何不需要的功能，包括CPU和WiFi硬件。但是，如果你的服务是播放或流媒体音乐，你希望防止系统干扰你的播放。

为了确保你的服务在这些条件下继续运行，你必须使用“唤醒锁”。唤醒锁是一种向系统发出信号的方式，你的应用程序正在使用某些特性，即使手机处于空闲状态，也应该保持可用。

注意:你应该经常保守地使用唤醒锁，并且只在必要时使用，因为它们会显著降低设备的电池寿命。

为了确保在你的MediaPlayer播放时，CPU持续运行，在初始化你的MediaPlayer时调用setWakeMode()方法。一旦你这样做了，MediaPlayer会在播放时保持指定的锁，当暂停或停止时释放锁：

```

mMediaPlayer = new MediaPlayer();
// ... 在此做其他初始化 ...
mMediaPlayer.setWakeMode(getApplicationContext(), PowerManager.PARTIAL_WAKE_LOCK);

```

但是，在这个示例中获取的唤醒锁只保证CPU保持醒着。如果你通过Wi-Fi网络流出媒体，你可能也想要保持一个Wi-Fi锁，你必须手动获取和释放。因此，当你开始使用远程URL准备MediaPlayer时，你应该创建并获取Wi-Fi锁。例如：

```

WifiLock wifiLock =
    ((WifiManager) getSystemService(Context.WIFI_SERVICE))
        .createWifiLock(WifiManager.WIFI_MODE_FULL, "mylock");
wifiLock.acquire();

```

当你暂停或停止你的媒体，或当你不再需要网络时，你应该释放锁：

```

wifiLock.release();

```

## 10. 执行清理

正如前面提到的，一个MediaPlayer对象可以消耗大量的系统资源，所以你应该只在需要时保留它，并在用完时调用release()。很重要的一点是，要显式地调用这个清理方法，而不是依赖于系统垃圾收集，因为垃圾回收器可能需要过一段时间

才能回收MediaPlayer，它只对内存需求敏感，而不是其他媒体相关资源的缺少。因此，在使用服务时，应该始终覆盖onDestroy()方法，以确保释放MediaPlayer:

```
public class MyService extends Service {  
    MediaPlayer mMediaPlayer;  
    // ...  
    @Override  
    public void onDestroy() {  
        if (mMediaPlayer != null) mMediaPlayer.release();  
    }  
}
```

你也应该总是寻找其他机会释放你的MediaPlayer，除了在被关闭时释放它。例如，如果你希望在一段时间内不能播放媒体(例如，在丢失了音频焦点之后)，你应该明确地释放你现有的MediaPlayer并在以后再创建它。另一方面，如果你只希望在很短的时间内停止播放，那么你可能应该保留你的MediaPlayer，以避免再次创建和准备它的开销。

#### 11. 从内容提供者获取媒体

在媒体播放器应用程序中可能有用的另一个功能是能够检索用户在设备上的音乐。可以通过查询外部媒体的内容提供者来做到这一点:

```
ContentResolver contentResolver = getContentResolver();  
Uri uri = android.provider.MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;  
Cursor cursor = contentResolver.query(uri, null, null, null, null);  
if (cursor == null) {  
    // 查询失败，处理错误  
} else if (!cursor.moveToFirst()) {  
    // 该设备上没有媒体  
} else {  
    int titleColumn =  
        cursor.getColumnIndex(android.provider.MediaStore.Audio.Media.TITLE);  
    int idColumn =  
        cursor.getColumnIndex(android.provider.MediaStore.Audio.Media._ID);  
    do {  
        long thisId = cursor.getLong(idColumn);  
        String thisTitle = cursor.getString(titleColumn);  
        // ...程序入口...  
    } while (cursor.moveToNext());  
}
```

要在MediaPlayer中利用这一点，你可以这样做:

```
long id = /* 从某处取得 */;  
Uri contentUri = ContentUris.withAppendedId(  
    android.provider.MediaStore.Audio.Media.EXTERNAL_CONTENT_URI, id);  
mMediaPlayer = new MediaPlayer();  
mMediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);  
mMediaPlayer.setDataSource(getApplicationContext(), contentUri);  
// ...prepare和start...
```

Java代码示意(如下图)

图3-7 Android Studio截图

Android Manifest程序清单(如下图)

图3-8 Android Studio截图

XML Layout布局(如下图)

图3-9 Android Studio截图

在Android模拟器上调试(如下图)

图3-10 Android模拟器截图

#### 3.4 测试

Build工程，生成apk文件，在手机上安装并打开，应用程序正常运行。依照表3-1至表3-16所述，对此音乐播放器逐条测试，各个事件均正常。截图如下：

### 1. 清单声明

在开始使用MediaPlayer开发应用程序之前，确保你的清单具有允许使用相关特性的适当声明。

Internet权限 - 如果你正在使用MediaPlayer来流基于网络的内容，你的应用程序必须请求网络访问。

```
<uses-permission android:name="android.permission.INTERNET" />
```

唤醒锁许可 - 如果你的播放器应用程序需要防止屏幕变暗或处理器休眠，或使用

MediaPlayer.setScreenOnWhilePlaying()或MediaPlayer.setWakeMode()方法，你必须请求许可。

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

### 2. 使用MediaPlayer

媒体框架中最重要的组件之一是MediaPlayer类。这个类的一个对象可以获取、解码和播放带有最小设置的音频和视频。

但是，该资源的内容不应该是原始音频。它应该是一种被支持的格式正确编码和格式化的媒体文件。

3. 在使用setDataSource()时，你必须捕获或传递IllegalArgumentException和IOException，因为你所引用的文件可能不存在。

### 3. 异步准备

在原则上，使用MediaPlayer可以很简单。然而，重要的是要记住，要正确地集成一个典型的Android应用程序，还需要做更多的事情。例如，prepare()的调用可能需要很长的时间来执行，因为它可能涉及到获取和解码媒体数据。因此，正如任何可能需要很长时间才能执行的方法一样，你不应该从应用程序的UI线程调用它。

4. 即使你希望你的资源能够快速加载，记住任何需要超过十分之一秒的时间在UI中响应，都会引起明显的停顿，并给用户留下你的应用程序很慢的印象。

为了避免挂起UI线程，生成另一个线程来准备MediaPlayer，并在完成时通知主线程。虽然你可以自己编写线程逻辑，但是在使用MediaPlayer时，这个模式非常常见，因为框架提供了一种方便的方法，可以使用prepareAsync()方法来完成这个任务。该方法开始在背景准备媒体，并立即返回。当媒体做好准备时，调用MediaPlayer.OnPreparedListener的onPrepared()方法，它是通过setOnPreparedListener()配置的。

### 4. 管理状态

另一个你应该记住的MediaPlayer的方面是它是基于状态的。也就是说，MediaPlayer有一个内部状态，在编写代码时，你必须始终注意到它，因为某些操作只有在特定状态下才有效。如果在错误的状态下执行操作，系统可能会抛出异常或导致其他不良行为。

MediaPlayer类中的文档显示了一个完整的状态图，它阐明了哪些方法将MediaPlayer从一个状态转换到另一个状态。例如，当你创建一个新的MediaPlayer时，它处于空闲状态。在这一点上，你应该通过调用setDataSource()来初始化它，将它变为初始化状态。在此之后，你必须使用prepare()或prepareAsync()方法来准备它。当MediaPlayer完成准备时，它进入准备状态，这意味着你可以调用start()来让它播放媒体。在这一点上，正如图所示，你可以通过调用start()、pause()、seekTo()等方法在开始、暂停和PlaybackCompleted状态之间转换。但是，当你调用stop()时，请注意你不能再次调用start()，直到你再次准备MediaPlayer。

在编写与MediaPlayer对象交互的代码时，始终要记住状态图，因为在错误的状态调用它的方法是bug的常见原因。

### 5. 释放MediaPlayer

MediaPlayer可以消耗宝贵的系统资源。因此，你应该始终采取额外的预防措施，以确保你没有在必要的时候挂在MediaPlayer实例上。当你完成它时，你应该始终调用release()来确保分配给它的任何系统资源都被适当地释放。

5. 当然，当你的活动恢复或重新启动时，你需要创建一个新的MediaPlayer并在恢复播放之前再次准备它。

下面是你应该如何释放和取消你的MediaPlayer:

```
mediaPlayer.release();
```

```
mediaPlayer = null;
```

举例来说，如果你在停止活动时忘记释放MediaPlayer，但在活动再次启动时创建一个新的，请考虑可能发生的问题。

6. 你可能想知道发生了什么，如果你想继续播放“后台媒体”，即使用户离开了你的活动，就像内置的音乐应用程序的行为一样。在这种情况下，你需要的是一个由服务控制的MediaPlayer，如下一节所讨论的。

### 6. MediaPlayer用于服务

如果你希望你的媒体在背景播放，即使你的应用程序不在屏幕显示——也就是说，在用户与其他应用程序交互时，你希望它继续播放——那么你必须启动一个服务并从那里控制MediaPlayer实例。你需要把MediaPlayer嵌入MediaBrowserServiceCompat服务并让它与MediaBrowserCompat在另一个活动交互。

你应该注意这个客户机/



7. 期望, 关于在背景服务中运行的播放器如何与系统的其余部分进行交互。如果你的应用程序没有满足这些期望, 用户可能会有糟糕的体验。
8. 因此, 服务需要快速地处理传入的意图, 并且在响应它们时从不执行冗长的计算。如果预期有任何繁重的工作或阻塞调用, 则必须异步执行这些任务: 从另一个线程中实现自己, 或者使用框架的许多设备进行异步处理。  
例如, 当在主线程中使用MediaPlayer时, 应该调用prepareAsync()而不是prepare(), 并实现一个MediaPlayer.OnPreparedListener, 以便在准备完成时通知你, 你可以开始播放了。
9. 使用唤醒锁  
在设计在背景播放媒体的应用程序时, 当你的服务运行时, 设备可能会休眠。由于Android系统试图在设备休眠时节省电量, 系统试图关闭手机的任何不需要的功能, 包括CPU和WiFi硬件。但是, 如果你的服务是播放或流媒体音乐, 你希望防止系统干扰你的播放。  
为了确保你的服务在这些条件下继续运行, 你必须使用“唤醒锁”。唤醒锁是一种向系统发出信号的方式, 你的应用程序正在使用某些特性, 即使手机处于空闲状态, 也应该保持可用。  
注意: 你应该经常保守地使用唤醒锁, 并且只在必要时使用, 因为它们会显著降低设备的电池寿命。  
为了确保在你的MediaPlayer播放时, CPU持续运行, 在初始化你的MediaPlayer时调用setWakeMode()方法。
10. 如果你通过Wi-Fi网络流出媒体, 你可能也想要保持一个Wi-Fi锁, 你必须手动获取和释放。因此, 当你开始使用远程URL准备MediaPlayer时, 你应该创建并获取Wi-Fi锁。
11. 很重要的一点是, 要显式地调用这个清理方法, 而不是依赖于系统垃圾收集, 因为垃圾回收器可能需要过一段时间才能回收MediaPlayer, 它只对内存需求敏感, 而不是其他媒体相关资源的缺少。
12. 另一方面, 如果你只希望在很短的时间内停止播放, 那么你可能应该保留你的MediaPlayer, 以避免再次创建和准备它的开销。
13. 媒体  
在媒体播放器应用程序中可能有用的另一个功能是能够检索用户在设备上的音乐。

## 7. 53120801\_沈建\_计算机科学与技术\_基于Android平台的音乐播放器设计与实现.doc\_第7部分

总字数: 1500

相似文献列表 文字复制比: 6.1%(92) 疑似剽窃观点: (0)

1	基于Android的考研“小助手” 周忠玉;陈汝特;张建标;王丽丽; - 《电脑知识与技术》- 2016-11-07 1	5% ( 75 ) 是否引证: 否
2	基于Android手机平台的农资安全监管系统的设计与实现 彭新航(导师: 王开义) - 《黑龙江八一农垦大学博士论文》- 2014-06-01	5% ( 75 ) 是否引证: 否
3	基于优化DTW算法的在线手写签名认证系统研究与设计 罗勇军(导师: 丁磊) - 《广东工业大学博士论文》- 2014-05-01	5% ( 75 ) 是否引证: 否
4	基于android平台的流量监控系统的设计与实现 冯卫广(导师: 刘琼;赵飞宇) - 《华南理工大学博士论文》- 2014-05-05	5% ( 75 ) 是否引证: 否
5	基于Web of Things技术的智能家居终端应用的设计与实现 徐楠(导师: 裴晓峰) - 《北京邮电大学博士论文》- 2013-03-05	4.2% ( 63 ) 是否引证: 否
6	Android输入法总体设计 张涛 - 《大学生论文联合比对库》- 2016-05-04	4% ( 60 ) 是否引证: 否

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

### 第4章结论

经过此次Android音乐播放器的设计与实现, 我对Android的系统结构有了初步的认识, 熟悉了Android应用程序的开发流程, 对多媒体技术和SQLite有了大致的了解。

Android OS在Linux Kernel的基础之上添加了系统库和Java虚拟机(Android Runtime), 并在此之上建立了一个Java应用程序框架, 所有的应用程序都运行在Application Framework之上。

Surface Manager、Media Framework、SQLite、OpenGL ES、FreeType、WebKit、SGL、SSL、libc等系统库提供了Android OS的一些重要功能, 如界面管理、多媒体框架、数据库、图形渲染、字体引擎、浏览器引擎、网络通信等。

Android Runtime由核心库和Dalvik/ART虚拟机组成。其中核心库提供了Java核心库的绝大部分功能, 而ART Virtual Machine (Dalvik自Android 5.0起被淘汰)则负责运行应用程序。每个应用程序都运行在ART虚拟机单独的一个实例上, 这个实例驻留在一个由Linux Kernel管理的进程中。

而我们开发Android应用程序, 通常不需要去了解底层的東西, 只需要理解这个Android应用程序框架, 利用系统提供的API来完成程序设计。

Android应用程序的四个组件, Activity(活动)、Service(服务)、Broadcast Receiver(广播接收器)、Content Provider(内容提供者), 和各项组件之间相互通讯的Intent(意图), 构成Android应用程序的基本框架。

一个完整Android应用程序, 与以往习惯做的控制台程序完全不同。在动手编码之前, 需要事先对整个程序框架有清晰的

设想，程序的逻辑功能、具体的交互动作、数据存储方式等。

一个音乐播放器的界面，主要由主界面、播放界面、播放列表、侧滑菜单、歌曲搜索组成。有播放/暂停、上/下一曲、循环模式、跳转顺序、播放列表、音乐搜索、歌词显示等功能。由于需要处理大量的音乐文件的多条信息，所以使用SQLite数据库来存储数据。

音频文件格式多种多样，除了WAVE格式是没有压缩的，其他如MP3、AAC等都是压缩编码格式。Android OS的MediaPlayer类集成了MP3音频解码器，该类直接继承类java.lang.Object，借助MediaPlayer类的对象播放MP3文件，只需调用播放控制方法。依次调用reset方法复位MediaPlayer对象、setDataSource方法获得音频文件、prepare方法准备播放、start方法开始播放。

搭建开发环境：安装Android Studio 2.3.3，下载Android SDK (API 25)，安装JDK SE 1.8.0u162，安装Oracle VM VirtualBox 4.3.40和Genymotion 2.5.3，选择Google Nexus 4 with Android 5.1模板，创建Android虚拟机。

创建一个新的工程，等待Sync和Build，配置Gradle的相关参数，重新Sync和Build，然后编码、调试，设计Layout界面，在Android Manifest中注册各个组件、声明访问权限。逐一实现读取文件、播放控制、音乐列表、歌词显示、图片显示、音乐搜索等功能模块，完成音乐播放器整体框架，然后逐渐补充其他功能，最后进行软件测试。

#### 参考文献

- [1] 梁勇. Java语言程序设计[M]. 戴开宇,译. 北京:机械工业出版社, 2015-07-01(1).
- [2] 黄岚,王岩,王康平. Java程序设计[M]. 北京:机械工业出版社, 2013-08(1).
- [3] 明日科技. Java从入门到精通[M]. 北京:清华大学出版社, 2016-10-01(4).
- [4] 施威铭. Android APP开发入门:使用Android Studio环境[M]. 北京:机械工业出版社, 2016-07-01(1).
- [5] 欧阳燊. Android Studio开发实战:从零基础到App上线[M]. 北京:清华大学出版社, 2017-06-01(1).
- [6] 张泳,葛丽娜. 深入剖析Android开发:小应用里的大智慧[M]. 北京:人民邮电出版社, 2014-07-01(1).
- [7] 张勇. Android移动开发技术[M]. 北京:清华大学出版社, 2017-06(1).
- [8] 张海藩. 软件工程导论[M]. 北京:清华大学出版社, 2008-2(5).
- [9] MICK. 图灵程序设计丛书:SQL基础教程[M]. 孙淼等译. 北京:人民邮电出版社, 2013-8.
- [10] 邵超等. 数据库实用教程:SQL Server 2008[M]. 北京:清华大学出版社, 2009-8.
- [11] 学院博士、硕士与往届本科毕业生论文与毕业设计.
- [12] 中国知网数据库相关资料.
- [13] 学院资料室、教材、指导教师及网上查阅.

说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例

2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例

3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例

4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比

5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的

6.红色文字表示文字复制部分;绿色文字表示引用部分

7.本报告单仅对您所选择比对资源范围内检测结果负责



 [amlc@cnki.net](mailto:amlc@cnki.net)

 <http://check.cnki.net/>

 <http://e.weibo.com/u/3194559873/>