重庆大学

课 程 设 计 说 明 书

题目： Java 程序设计之课程设计

院　　系：　　计算机学院

专业班级：　　计算机科学与技术 4 班

学　　生：　　代金龙(20225231)

程　飞(20225265)

王小东(2022　　)

指导教师：　　李芝兴

2022　年 1 月 6 日

目 录

课程设计指导教师评定成绩表

姓名：代金龙　　学号： 20225231

项目	分值	优秀 (100>x≥90)	良好 (90>x≥80)	中等 (80>x≥70)	及格 (70>x≥60)	不及格 (x<60)	评分
		参考标准	参考标准	参考标准	参考标准	参考标准	
学习态度	15	学习态度认真，科学作风严谨，严格保证设计时间并按任务书中规定的进度开展各项工作	学习态度比较认真，科学作风良好，能按期圆满完成任务书规定的任务	学习态度尚好，遵守组织纪律，基本保证设计时间，按期完成各项工作	学习态度尚可，能遵守组织纪律，能按期完成任务	学习马糊，纪律涣散，工作作风不严谨，不能保证设计时间和进度	
技术水平与实际能力	25	设计合理、理论分析与计算正确，实验数据准确，有很强的实际动手能力、经济分析能力和计算机应用能力，文献查阅能力强、引用合理、调查调研非常合理、可信	设计合理、理论分析与计算正确，实验数据比较准确，有较强的实际动手能力、经济分析能力和计算机应用能力，文献引用、调查调研比较合理、可信	设计合理，理论分析与计算基本正确，实验数据比较准确，有一定的实际动手能力，主要文献引用、调查调研比较可信	设计基本合理，理论分析与计算无大错，实验数据无大错	设计不合理，理论分析与计算有原则错误，实验数据不可靠，实际动手能力差，文献引用、调查调研有较大的问题	
创新	10	有重大改进或者独特见解，有一定实用价值	有较大改进或者新颖的见解，实用性尚可	有一定改进或者新的见解	有一定见解	观念陈旧	
论文(计算书、图纸)撰写质量	50	结构严谨，逻辑性强，层次清晰，语言准确，文字流畅，彻底符合规范化要求，书写工整或者用计算机打印成文；图纸非常工整、清晰	结构合理，符合逻辑，文章层次分明，语言准确，文字流畅，符合规范化要求，书写工整或者用计算机打印成文；图纸工整、清晰	结构合理，层次较为分明，文理通顺，基本达到规范化要求，书写比较工整；图纸比较工整、清晰	结构基本合理，逻辑基本清晰，文字尚通顺，勉强达到规范化要求；图纸比较工整	内容空泛，结构混乱，文字表达不清，错别字较多，达不到规范化要求；图纸不工整或者不清晰	

指导教师评定成绩：

指导教师签名：　　年　　月　　日

课程设计指导教师评定成绩表

姓名：程飞　　学号： 20225265

项目	分值	优秀 (100>x≥90)	良好 (90>x≥80)	中等 (80>x≥70)	及格 (70>x≥60)	不及格 (x<60)	评分
		参考标准	参考标准	参考标准	参考标准	参考标准	
学习态度	15	学习态度认真，科学作风严谨，严格保证设计时间并按任务书中规定的进度开展各项工作	学习态度比较认真，科学作风良好，能按期圆满完成任务书规定的任务	学习态度尚好，遵守组织纪律，基本保证设计时间，按期完成各项工作	学习态度尚可，能遵守组织纪律，能按期完成任务	学习马糊，纪律涣散，工作作风不严谨，不能保证设计时间和进度	
技术水平与实际能力	25	设计合理、理论分析与计算正确，实验数据准确，有很强的实际动手能力、经济分析能力和计算机应用能力，文献查阅能力强、引用合理、调查调研非常合理、可信	设计合理、理论分析与计算正确，实验数据比较准确，有较强的实际动手能力、经济分析能力和计算机应用能力，文献引用、调查调研比较合理、可信	设计合理，理论分析与计算基本正确，实验数据比较准确，有一定的实际动手能力，主要文献引用、调查调研比较可信	设计基本合理，理论分析与计算无大错，实验数据无大错	设计不合理，理论分析与计算有原则错误，实验数据不可靠，实际动手能力差，文献引用、调查调研有较大的问题	
创新	10	有重大改进或者独特见解，有一定实用价值	有较大改进或者新颖的见解，实用性尚可	有一定改进或者新的见解	有一定见解	观念陈旧	
论文(计算书、图纸)撰写质量	50	结构严谨，逻辑性强，层次清晰，语言准确，文字流畅，彻底符合规范化要求，书写工整或者用计算机打印成文；图纸非常工整、清晰	结构合理，符合逻辑，文章层次分明，语言准确，文字流畅，符合规范化要求，书写工整或者用计算机打印成文；图纸工整、清晰	结构合理，层次较为分明，文理通顺，基本达到规范化要求，书写比较工整；图纸比较工整、清晰	结构基本合理，逻辑基本清晰，文字尚通顺，勉强达到规范化要求；图纸比较工整	内容空泛，结构混乱，文字表达不清，错别字较多，达不到规范化要求；图纸不工整或者不清晰	

指导教师评定成绩：

指导教师签名：　　年　　月　　日

课程设计指导教师评定成绩表

姓名：王小东　学号： 2022

项目	分值	优秀 (100>x≥90)	良好 (90>x≥80)	中等 (80>x≥70)	及格 (70>x≥60)	不及格 (x<60)	评分
		参考标准	参考标准	参考标准	参考标准	参考标准	
学习态度	15	学习态度认真，科学作风严谨，严格保证设计时间并按任务书中规定的进度开展各项工作	学习态度比较认真，科学作风良好，能按期圆满完成任务书规定的任务	学习态度尚好，遵守组织纪律，基本保证设计时间，按期完成各项工作	学习态度尚可，能遵守组织纪律，能按期完成任务	学习马糊，纪律涣散，工作作风不严谨，不能保证设计时间和进度	
技术水平与实际能力	25	设计合理、理论分析与计算正确，实验数据准确，有很强的实际动手能力、经济分析能力和计算机应用能力，文献查阅能力强、引用合理、调查调研非常合理、可信	设计合理、理论分析与计算正确，实验数据比较准确，有较强的实际动手能力、经济分析能力和计算机应用能力，文献引用、调查调研比较合理、可信	设计合理，理论分析与计算基本正确，实验数据比较准确，有一定的实际动手能力，主要文献引用、调查调研比较可信	设计基本合理，理论分析与计算无大错，实验数据无大错	设计不合理，理论分析与计算有原则错误，实验数据不可靠，实际动手能力差，文献引用、调查调研有较大的问题	
创新	10	有重大改进或者独特见解，有一定实用价值	有较大改进或者新颖的见解，实用性尚可	有一定改进或者新的见解	有一定见解	观念陈旧	
论文(计算书、图纸)撰写质量	50	结构严谨，逻辑性强，层次清晰，语言准确，文字流畅，彻底符合规范化要求，书写工整或者用计算机打印成文；图纸非常工整、清晰	结构合理，符合逻辑，文章层次分明，语言准确，文字流畅，符合规范化要求，书写工整或者用计算机打印成文；图纸工整、清晰	结构合理，层次较为分明，文理通顺，基本达到规范化要求，书写比较工整；图纸比较工整、清晰	结构基本合理，逻辑基本清晰，文字尚通顺，勉强达到规范化要求；图纸比较工整	内容空泛，结构混乱，文字表达不清，错别字较多，达不到规范化要求；图纸不工整或者不清晰	

指导教师评定成绩：

指导教师签名：　　年　月　日

课程设计任务说明书

课程设计题目	Java 程序设计之课程设计				
学院	计算机学院	专业	计算机科学与技术	年级	2022

实验教学的目的、任务与要求

Java 程序设计是计算机网络工程专业一门重要的专业必修课。 为了进一步巩固课堂上所学到的知识， 深刻把握 Java 技术的重要概念及其面向对象的特性， 锻炼学生熟练的应用面向对象的思想和设计方法解决实际问题的能力，开设 Java 程序设计的课程设计。

课程设计的任务是完成课程设计内容，写出课程设计报告。

要求学生掌握：

1) 掌握 Java 的语言规范，面向对象的核心概念和特性；
2) 掌握 Java 的编程技术，包括异常处理，图形界面设计，多线程，网络通信程序等；
3) 掌握 Java 应用软件的开辟环境和开辟过程；
4) 掌握基于 Jsp 网站的开辟环境和开辟过程；
5) 掌握面向对象的思想和程序设计方法。

学生应完成的工作：

分组 3~4 人/组

内容： java 版视频播放器

语言： Java

具体设计内容及要求：

1) 对视频文件的正确播放；
2) 能够实现对视频播放的控制，如暂停，播放，快进，快退，上一个，下一个等功能；
3) 能够进行文件视频的选择，全屏，音量的控制，拖动，播放模式的控制等；
4) 其它的一些提示信息或者附加功能。

目前资料采集情况(含指定参考资料)：

1) 《Java 程序设计之网络编程》， 李芝兴编，清华大学出版社，出版时间 2022 年 3 月
2) (美) Cay S.Horstmann,Gary Cornell 编，《Java2 核心技术第 6 版：基础知识》，机械工业出版社，出版时间 2003 年 10 月
3) (美) Bruce Eckel 编，《Java 编程思想第 2 版》，机械工业出版社，出版时间 2002 年 9 月
4) 《JMF 入门指南》，网络下载。

课程设计的工作计划：

1．需求分析(说明系统目的，要求，操作流程，开辟工具与开辟平台)
2．总体设计(描述系统功能，系统架构，模块化分)
3．详细设计(建立系统的数据结构，协议结构，数据流程图)
4．系统实现编码(用所选开辟工具完成应用系统)

任务下达日期 2022 年 12 月 24 日　　完成日期 2022 年 1 月 6 日

指导教师 _____(签名)　　学　生 _____(签名)

选题

利用 java JMF 编制一个能播放 mpeg，mpg，mov 等格式的视频播放器；要求能够进行播放文件的选择，文件列表框中陈列选择的视频文件，能够控制视频的播放，退出，住手，快进，快退，下一个，上一个，音量的控制，静音控制，全屏选择，任意改变播放视频界面大小等功能。

3.需求分析

3.1 任务目的：

1. 实现视频文件的正确播放；
2. 能够实现对视频播放的控制，如暂停，播放，快进，快退，上一个，下一个等功能；
3. 能够进行文件视频的选择，全屏，音量的控制，拖动，播放模式的控制等；
4. 任意改变播放视频界面大小等功能。

要求学生掌握：

1) 掌握 Java 的语言规范，面向对象的核心概念和特性。

2) 掌握 Java 的编程技术， 包括异常处理， 图形界面设计， 多线程， 网络通信程序等。

3) 掌握 Java 应用软件的开辟环境和开辟过程

4) 掌握基于 Jsp 网站的开辟环境和开辟过程

5) 掌握面向对象的思想和程序设计方法。

3.2 程序的设计、调试、运行的软件环境：

操作系统： Windows XP (SP2)

数据库及数据库管理软件： SQL Server 2000

JDK 环境： Java SE Development Kit (JDK) Version 6

开辟工具： Eclipse3.2

运行平台： Windows、Linux 各个版本、 MAC 等任何平台

运行环境： Java SE Runtime Environment (JRE) Version

4.程序设计过程

4.1 实验原理

JMF 提供了一个平台无关的框架来呈现时基媒体(time-based media)。JavaMediaPlayerAPI 的设计目标是支持多种标准的媒体格式，包括 MPEG-1,MPEG-2,QuickTime,AVI,WAV,AU 和 MIDI.使用 JMF,可以同步呈现不同来源的时基媒体。

现有的一些媒体播放器都严重依赖原生码来执行解压缩、渲染 等 计算密集型任务。而 JMF API 则隐藏了具体实现，只提供抽象的编程接口。

举例来说，一个用 JMF 制作的播放器，具体运行的过程中可能会调用到操作系统的本地方法，但开辟者写代码时可以无视本地方法的存在。

JMF Player API ：

- 接入不同的协议和分发(传输)机制
- 接入不同的媒体数据类型

定义事件模型，用于 Player 和应用程序间的异步通信

4.1.1 数据源(Data Source)

一个 DataSource 封装了媒体的地址、协议和分发的软件。一个JavaMediaPlayer 包含一个 DataSource。一旦创建，这个 DataSource 不能被用于其他媒体的传输。 一个 Player 的数据源可以用 MediaLocator 或者 URL 来标示。

MdiaLocator(媒体定位器)是一个 JMF 类，用来描述 Player 播放的媒体。 MediaLocator 与 URL 类似， 并且可以从 URL 来创建。 他们的区别在于， 在 Java 中，URL 惟独其协议是已知协议时才干被创建，而 MediaLocator 则没有这个限制。

Java 媒体播放器可以播放来自多种数据源的媒体数据， 包括本地、 网络文件和实时广播等。JMF 支持两种不同的数据源。

- Pull Data-Source （数据源被动的被获取）－由客户端发起数据传输并控制数据流，已知的协议包括 HTTP(超文本传输协议)和本地文件。
- Push Data-Source （数据源主动推送）－由服务器发起数据传输并控制推送数据流。此类数据源包括广播媒体，多点传送的媒体和 VOD(视频点播)。

客户端程序所能够控制的度量，取决于媒体源的类型。举个例子说， MPEG 文件可以被重定位 (reposition) ,那末播放 MPEG 的客户端程序就可以允许用户重播或者跳进至一个新时间点；而由服务器段控制的广播媒体则不能被重定位；此外 VOD 协议则支持有限的用户操作，比如一个 VOD 客户端程序可以允许用户跳进至新位置，但不能快进和快倒。

4.1.2 Players

一个 Java Media Player 是一个对象。她基于时间来处理数据流，从 DataSource 读取数据并在切当的时间点渲染媒体。一个 Java Media Player 必须实现 Player interface.

- Clock 定义了基本的计时和同步操作，　她被 Player 用来控制媒体的呈现。
- Controller 继承 Clock 对外提供提供如下方法：

1. 管理系统资源
2. 预载数据
3. 提供监听机制(Observable) ,对外发送媒体事件通知

- Duration 提供了检测媒体时长的途径。
- Player 支持标准的用户控制，并放宽了来自于 Clock 的一些操作限制。

多个 Player 共享一个公共的计时和同步模型。一个 Player 的媒体时间表示了媒体流的当前位置。每一个 Player 有一个 TimeBase。TimeBase 定义了 Player 的时间流逝。当一个 Player 被执行 start，他的媒体时间会被映射到 time-base 时间。如多个媒体要同步，那么他们必须使用同一个 TimeBase.

一个 Player 的用户界面可以包含一个可视组件和一个控制面板组件(control- panel component）。我们用的时候可以选择实现一个自定义的用户界面， 或者使用 Player 的默认的控制面板组件。

普通来说，一个 Player 在能够呈现媒体之前，必须先执行一序列的操作。而这些操作有可能会耗费一定的时间， 所以 JMF 定义了一些操作状态， 并且提供了状态转换的操作机制。

4.1.3 Media Events

JMF 事件报告机制允许我们的程序响应媒体驱动的错误，比如数据丢失或者资源不可用。事件系统同时也提供了重要的通知协议；当我们的程序调用一个 Player 的异步方法时，只有当收到响应的事件消息时，才干确认操作是否完成。

有两种 JMF 对象会抛出事件，他们是：　GainControl 对象 和 Controller 对象。　对于事件, GainControl 和 Controller 遵循 Java Beans 形式。

GainControl 对象只抛出一种类型的事件－ GainChangeEvent。我们通过实现 GainChangeListener interface 来响应 gain(增益？)的变化。

Controller 则会抛出多种派生自 ControllerEvent 的事件。我们通过实现 ControllerListener interface 来接收诸如 Player 的 Controller 抛出的事件消息。

下图显示了 Controller 抛出的各种事件类型：

ControllerEvents 可分为三类：改变通知、关闭事件和转换事件

- 变化通知(Change notification events)诸如 RateChangeEvent 和 DurationUpdateEvent。他们表示 Player 的一些属性数值发生了变化。这种事件通常是对一些方法调用的回馈。例如， 一个 Player 的 setRate 方法被调用，他会抛出一个 RateChangeEvent.
- 转换事件(TransitionEvents) 让我们的程序能够响应 Player 的状态变化。 当 Player 从一个状态转换到另一个状态时，就会抛出一个转换事件。 (在 1.4 中，我们会提供更多的关于 Player 状态的信息)
- 当 player 关闭时，则会抛出关闭事件(ControllerClosedEvents)。当一个 Player 再也不可用时， 抛出 ControllerClosedEvent. ControllerErrorEvent(控制器错误事 件)则是关闭事件的一个特例。我们写程序时，通过监听控制器错误事件，可以对 Player 故障作出响应，从而增进用户体验。

4.1.4 Player States

JavaMediaPlayer 有 6 种状态。Clock 接口定义了两种主要的状态：Stopped 和 Started。在普通操作中， Plyaer 在到达 Started 状态前，会逐个通过上图中的每一个状态。

- Unrealized (未实现) 状态表示 Player 已经被实例化，但还不知道媒体的任何信息。当 Player 第一次被创建，他的状态就是 Urealized。
- Player 的 realize 方法被调用后， 会从 Unrealized 转入 Realizing(实现中) 状态。这时的 Player 应该正在检测资源需求。在 relization 过程中， Player 会获取只需加在一次的资源。这些资源包括非独享的渲染资源。(独享资源指的受限的资源。例如只能被一个播放器使用的个别硬件资源， 此类资源会在预取 (Prefetching) 的过程中加载。) Realizing 中的 Player 往往通过网络下载东西。
- Player 结束 Realizing 状态后，会转入 Realized (已实现)状态。这个状态下，Player 会知道他需要哪些资源，还知道媒体的类型信息。因为 Realized Player 知道怎样渲染数据，所以他能够提供可视组件和控件。此时 ,Player 与其他系统对象的连接已经就位，但此时还不会占用任何会阻挠其他 Player 启动的资源。
- Prefetch 被调用后， Player 会从 Realized 状态转入 Prefetching (预取中)状态。此时播放器在为呈现媒体作准备， 包括调用媒体数据、 获取独享资源 和其他一些准备工作。在媒体呈现过程中，一些操作可能会导致 Prefetching 状态 重现，例如重新定位播放位置、播放器请求额外的缓冲区等。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
https://d.book118.com/248115074017006065