华 北 科 技 学 院

Java课程设计

说明书

学号：

班级: 姓名:

设计题目: 电脑端记事本开发设计

设计地点: 基础六实验室

设计时间: 2025-6-30 至 2025-7-11

成绩评定：

1、工作量： A（ ），B（ ），C（ ），D（ ）,F( )

2、难易度： A（ ），B（ ），C（ ），D（ ）,F( )

3、答辩情况：

基本操作： A（ ），B（ ），C（ ），D（ ）,F( )

代码理解： A（ ），B（ ），C（ ），D（ ）,F( )

4、报告规范度： A（ ），B（ ），C（ ），D（ ）,F( )

5、学习态度： A（ ），B（ ），C（ ），D（ ）,F( )

总评成绩:

指导教师:

**目录**

[1问题描述及需求分析 1](#_Toc202955539)

[1.1问题描述 1](#_Toc202955540)

[1.2功能需求分析 1](#_Toc202955541)

[2系统设计 2](#_Toc202955542)

[2.1功能模块设计 2](#_Toc202955543)

[2.2模块简介 3](#_Toc202955544)

[2.3数据库设计 3](#_Toc202955545)

[3详细设计 3](#_Toc202955546)

[3.1类设计 3](#_Toc202955547)

[3.2模块设计与实现 4](#_Toc202955548)

[3.2.1行号设置 4](#_Toc202955549)

[3.2.2文件操作 5](#_Toc202955550)

[3.2.3编辑操作 6](#_Toc202955551)

[3.2.4字体与颜色设置 7](#_Toc202955552)

[3.2.5状态与帮助 9](#_Toc202955553)

[3.3运行调试与分析 9](#_Toc202955554)

[3.3.1记事本界面 9](#_Toc202955555)

[3.3.2文件操作设计 10](#_Toc202955556)

[3.3.3编辑操作设计 11](#_Toc202955557)

[3.3.4字体与颜色设计 11](#_Toc202955558)

[3.3.5状态与帮助设计 13](#_Toc202955559)

[4总结体会 14](#_Toc202955560)

[参考文献 15](#_Toc202955561)

[AIGC检测 15](#_Toc202955562)

# 1问题描述及需求分析

## 1.1问题描述

本课程设计是Java语言学习中很重要的实践性环节之一，是在学习完Java课程后进行的一次全面的综合练习，旨在巩固和加深对Java语言基本知识的理解和掌握，掌握Java语言编程和程序调试的基本技能，利用Java语言进行基本的软件设计，掌握书写程序设计说明文档的能力，提高运用Java语言解决实际问题的能力。

针对以上要求，电脑端记事本是每台电脑的标配，有相当大的实用性，方便人们平时的记事之用，尤其是在快速笔记这方面，更是有非常大的作用，基本能满足人们的记事需求，有很大的开发及继续完善开发的意义。

基于记事本的诸多优点，本课程设计针对电脑端的记事本进行开发设计，并在原有基础上进行完善，使它的功能更完善、更人性化及更实用化。

## 1.2功能需求分析

随着人们生活信息化的提高，记事本只拘泥于笔和纸的时代已经一去不复返了，越来越多的电子版记事本进入了人们的生活。但如今的电脑端记事本软件感觉功能不够丰富，缺少一些个性化功能，导致用户体验不是很好，故本课程设计将开发一个加强版的电脑端记事本，来满足用户的需求。

具体功能如下：

（1）“文件”主菜单中有“新建”、“打开”、“保存”、“另存为”、“页面设置”、“打印”、“退出”这几个子功能。

（2）“编辑”主菜单中有“撤销”、“剪切”、“复制”、“粘贴”、“删除”、“查找”、“查找下一个”、“替换”、“转到”、“全选”、“日期/时间”这几个子功能。

（3）“格式”主菜单中有“自动换行”、“字体”、“背景颜色”、“字体颜色”这四个子功能。

（4）“查看”主菜单中有“状态栏”子功能。

（5）“帮助”主菜单中有“查看帮助”、“关于记事本”这两个子功能。

# 2系统设计

## 2.1功能模块设计

依据程序的功能需求分析，得到该系统的功能模块图如图1所示。

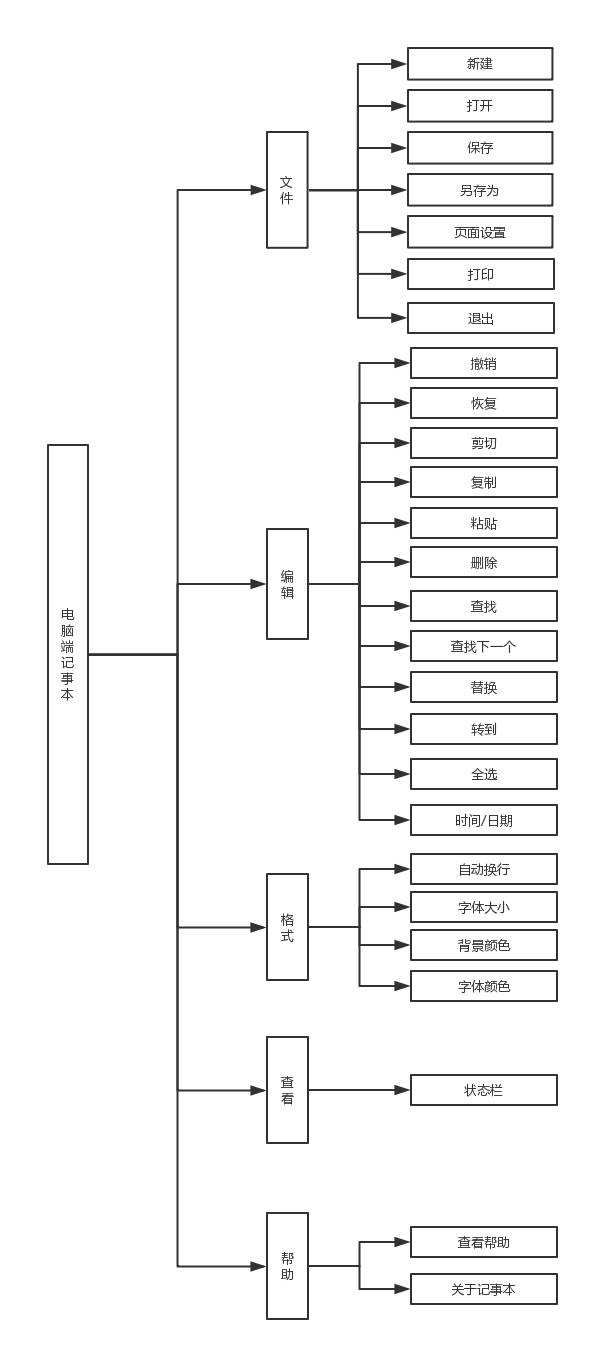


图1 系统功能模块图

## 2.2模块简介

在实现系统自带笔记本的功能同时，再添加一些个性化功能，例如为记事本添加上行号（这大大提高了我们程序员看代码的方便性），在状态栏添加上当前时间以及字数统计，让用户能够对自己所写的字数一目了然，大大增强了用户体验。

此外，此记事本支持用户自定义背景颜色以及字体颜色，增强了趣味性，用户可以根据自己的喜好选择符合自己的主题。

即实现的功能有：

（1）“文件”主菜单中有“新建”、“打开”、“保存”、“另存为”、“页面设置”、“打印”、“退出”这几个子功能。

（2）“编辑”主菜单中有“撤销”、“剪切”、“复制”、“粘贴”、“删除”、“查找”、“查找下一个”、“替换”、“转到”、“全选”、“日期/时间”这几个子功能。

（3）“格式”主菜单中有“自动换行”、“字体”、“背景颜色”、“字体颜色”这四个子功能。

（4）“查看”主菜单中有“状态栏”子功能。

（5）“帮助”主菜单中有“查看帮助”、“关于记事本”这两个子功能。

## 2.3数据库设计

记事本无数据库设计

# 3详细设计

## 3.1类设计

依据程序的功能划分，系统实现设计以下五个类。

类名：NotepadMainFrame

作用：主类，继承JFrame类来承载组件、跨平台兼容，实现接口ActionListener，用于监听对记事本的操作。

关键代码：public class NotepadMainFrame extends JFrame implements ActionListener{}

类名：MQFontChooser

作用：内部类，继承JDialog类，自定义字体选择器，修改字体。

关键代码：public class MQFontChooser extends JDialog {}

类名：TestLine

作用：内部类，继承JComponent类，显示行号，绘制行号设计，提高可阅读性。

关键代码：public class TestLine extends javax.swing.JComponent {}

类名：SystemParam

作用：内部类，实现使用当前系统换行符，多系统使用。

关键代码：

public class SystemParam { public static final String *LINE\_SEPARATOR*=System.*getProperty*("line.separator");}

类名：Clock

作用：内部类，继承Thread类进行多线程操作，实现状态栏动态显示。

关键代码：public class Clock extends Thread{}

## 3.2模块设计与实现

### 3.2.1行号设置

行号设置实现的是：计算当前可见区域需要显示的行号，只绘制当前可见的行号，行号文字采用右对齐显示，内置间距和位置调整通过字体测量获取精确的尺寸信息，优化绘制过程，提高性能，自动适应不同长度的行号。核心代码详解如下：

protected void paintComponent(Graphics g) {  
 int nlineHeight = getLineHeight();  
 int startOffset = getStartOffset();  
 Rectangle drawHere = g.getClipBounds();//重绘区域  
 g.setColor(getBackground());  
 g.fillRect(drawHere.x, drawHere.y, drawHere.width, drawHere.height);  
 g.setColor(getForeground());  
 int startLineNum = (drawHere.y / nlineHeight) + 1;//第一行起点  
 int endLineNum = startLineNum + (drawHere.height / nlineHeight);  
 int start =startLineNum \* nlineHeight - startOffset;//y为基线位置，减去四个像素  
 for (int i = startLineNum; i <= endLineNum; ++i) {//循环绘制  
 String lineNum = String.valueOf(i);  
 int width = fontMetrics.stringWidth(lineNum);//控制右间距  
 g.drawString(lineNum + " ", MARGIN + currentRowWidth - width -1, start);//总宽度-右间距=紧贴右  
 start += nlineHeight;  
 }  
 setPreferredSize(endLineNum);  
}

Graphics类提供绘图上下文，通过getClipBounds获取重绘区域范围。Rectangle类表示绘图区域边界。颜色设置使用setColor方法分别设置背景色和前景色。字体度量通过fontMetrics计算字符串宽度。

行号计算基于行高nlineHeight和起始偏移量startOffset。循环绘制使用startLineNum和endLineNum确定绘制范围。字符串转换使用String.valueOf将行号转为字符串。尺寸控制通过setPreferredSize方法设置组件首选大小。

坐标计算考虑基线位置调整和右边距MARGIN。数值运算确定每行绘制位置。currentRowWidth变量记录当前行宽度。绘制方法drawString实现文本输出。

### 3.2.2文件操作

文件操作实现的是：包括打开文件（openFile）、保存文件（save）、另存为（saveAs）、新建文件（newFile）、退出（exit）、页面设置（打开页面设置对话框）、打印（Print）等。核心代码详解如下：

//获取行距  
FontMetrics metrics = g2d.getFontMetrics();  
int lineHeight = metrics.getHeight();  
String[] lines = textArea.getText().split("\n");  
int linesPerPage = (int)(pageFormat.getImageableHeight() / lineHeight);//一页的行数  
int totalPages = (int)Math.ceil((double) lines.length / linesPerPage);//打印所需要的页数  
if (pageIndex >= totalPages) return Printable.NO\_SUCH\_PAGE;  
int start = pageIndex \* linesPerPage;  
int end = Math.min(start + linesPerPage, lines.length);//结束行索引，避免越界。若剩余行数不足一页（如最后一页），则取实际行数作为结束位置

### 3.2.3编辑操作

编辑操作实现的是：撤销（undo）和恢复（redo）：使用undoMgr进行撤销和恢复操作。剪切（cut）、复制（copy）、粘贴（paste）、删除（删除选中的文本）、查找、查找下一个、替换（通过mySearch方法处理）、转到指定行（turnTo）、全选（selectAll）等。核心代码详解如下：

int searchStart;  
if (isUp) {  
 searchStart = (textArea.getSelectedText() == null) ? caretPos - 1 : caretPos - keyword.length() - 1;//有选择从选择前查找  
 if (searchStart < 0) searchStart = source.length() - 1;//转至末尾  
 index = source.lastIndexOf(target, searchStart);//IndexOf的逆置  
 if (index == -1 && searchStart != source.length() - 1) {  
 index = source.lastIndexOf(target); // 循环查找  
 }  
} else {  
 searchStart = (textArea.getSelectedText() == null) ? caretPos : caretPos - keyword.length() + 1;  
 if (searchStart >= source.length()) searchStart = 0;  
 index = source.indexOf(target, searchStart);  
 if (index == -1 && searchStart != 0) {  
 index = source.indexOf(target); // 循环查找  
 }  
}

Graphics2D类的FontMetrics用于获取字体度量信息，包括行高。String类的split方法按换行符分割文本内容。Math类的ceil方法计算总页数，确保向上取整。Printable接口的NO\_SUCH\_PAGE常量表示无效页码。基本数学运算计算起始行和结束行索引，使用Math.min方法防止数组越界。变量linesPerPage表示每页可打印行数，totalPages表示总页数。通过计算确定当前页的起始行和结束行，处理最后一页行数不足的情况。

### 3.2.4字体与颜色设置

字体与颜色设置：字体设置：通过自定义字体选择器（MQFontChooser）选择字体，并更新文本区域的字体。背景颜色和字体颜色：使用JColorChooser选择颜色，并设置文本区域的背景色和前景色。关键代码详解如下：

private void addListener() {  
 sizeText.addFocusListener(new FocusListener() {  
 public void focusLost(FocusEvent e) {  
 setPreview();  
 }  
 public void focusGained(FocusEvent e) {  
 sizeText.selectAll();  
 }  
 });  
 // 字体列表发生选择事件的监听器  
 fontList.addListSelectionListener(new ListSelectionListener() {  
 public void valueChanged(ListSelectionEvent e) {  
 if (!e.getValueIsAdjusting()) {  
 fontText.setText(String.*valueOf*(fontList.getSelectedValue()));  
 // 设置预览  
 setPreview();  
 }  
 }  
 });  
 styleList.addListSelectionListener(new ListSelectionListener() {  
 public void valueChanged(ListSelectionEvent e) {  
 if (!e.getValueIsAdjusting()) {  
 styleText.setText(String.*valueOf*(styleList.getSelectedValue()));  
 // 设置预览  
 setPreview();  
 }  
 }  
 });  
 sizeList.addListSelectionListener(new ListSelectionListener() {  
 public void valueChanged(ListSelectionEvent e) {  
 if (!e.getValueIsAdjusting()) {  
 if(!sizeText.isFocusOwner()){  
 sizeText.setText(String.*valueOf*(sizeList.getSelectedValue()));  
 }  
 // 设置预览  
 setPreview();  
 }  
 }  
 });

FocusListener接口用于处理文本框的焦点变化事件，实现focusLost和focusGained方法分别响应失去焦点和获得焦点事件。ListSelectionListener接口监听列表选择变化，通过valueChanged方法处理选择事件，使用getValueIsAdjusting方法避免重复触发。

JTextField组件通过selectAll方法实现获得焦点时全选文本内容。JList组件通过getSelectedValue方法获取当前选中项的值。事件处理机制通过判断isFocusOwner来避免重复设置文本框内容。

setPreview方法被多个监听器调用实现统一预览更新逻辑。字符串转换使用String.valueOf方法将列表选中值转为文本。条件判断逻辑确保只在选择完成时触发操作。

### 3.2.5状态与帮助

状态与帮助：切换状态栏的可见性：根据菜单项的选中状态设置工具状态栏的可见性。帮助和关于：弹出消息对话框，显示帮助信息和软件说明。关键代码详解如下：

//记录行数和列数，实时刷新  
textArea.addCaretListener(e -> {  
 JTextArea editArea = (JTextArea)e.getSource();  
 try {  
 int caretpos = editArea.getCaretPosition();//获取光标位置  
 linenum = editArea.getLineOfOffset(caretpos);//获取光标行数  
 columnnum = caretpos - textArea.getLineStartOffset(linenum);//获取光标列数  
 label2.setText(" 第 " + (linenum+1) + " 行, 第 " + (columnnum+1)+" 列 ");  
 length=NotepadMainFrame.this.textArea.getText().length();  
 label3.setText(" 一共 " +length+" 字 ");  
 }  
 catch(Exception ex) { }});

## 3.3运行调试与分析

### 3.3.1记事本界面

记事本界面用于显示记事本内容与行号，程序成功运行时，即可进入主界面，如图2所示。



图2 记事本主界面

### 3.3.2文件操作设计

单击菜单处文件或者快捷键Alt+F，查看文件菜单的操作模块。执行另存为操作如图3所示。

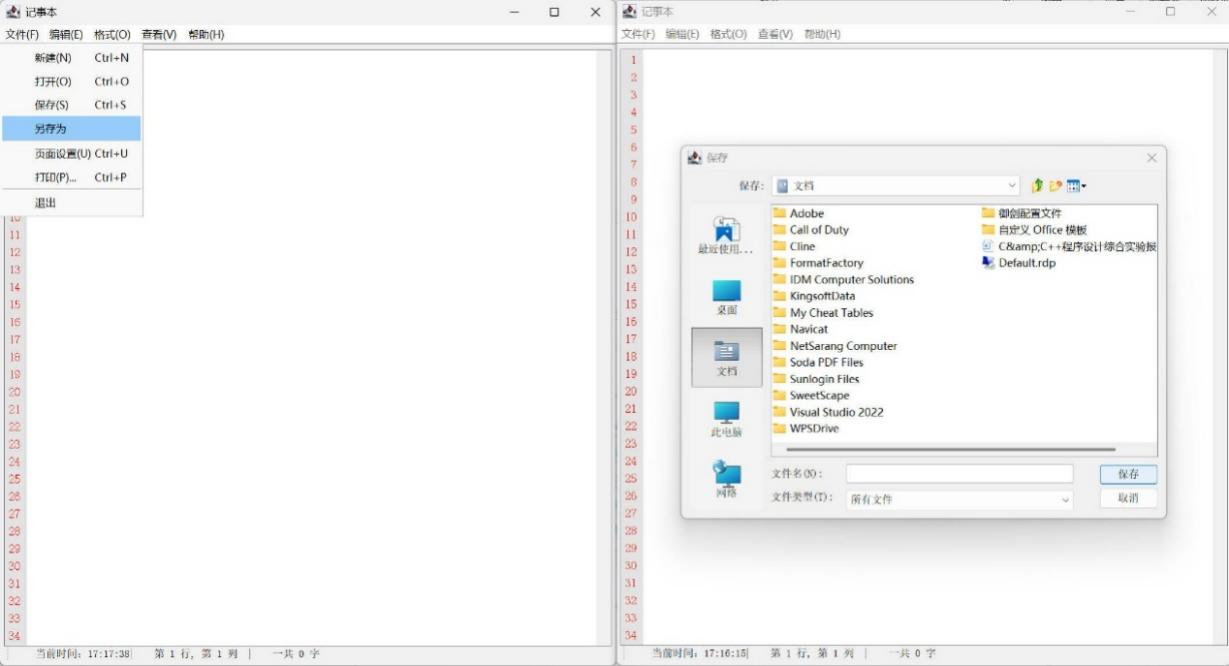


图3 另存为操作

### 3.3.3编辑操作设计

单击菜单处编辑或者快捷键Alt+E，。查看编辑菜单的操作模块。执行查找如图4所示。

图形用户界面, 应用程序, Word

AI 生成的内容可能不正确。

图4 查询结果

### 3.3.4字体与颜色设计

单击菜单处格式或者快捷键Alt+O，。查看格式菜单的操作模块。执行字体颜色如图5所示。执行字体大小格式如图6所示。执行背景颜色如图7所示。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图5 字体颜色

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图6 字体大小格式

图表

AI 生成的内容可能不正确。

图7 背景颜色

### 3.3.5状态与帮助设计

可拖动状态栏如图8所示。单击菜单处帮助或者快捷键Alt+H，。查看帮助菜单的操作模块。执行帮助与关于如图9所示。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

图8 状态栏

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图9 帮助与关于

# 4总结体会

系统设计实现的功能：文件操作模块支持新建文件功能，用户可以创建空白文档开始编辑；支持打开文件功能，能够读取现有文本文件内容；支持保存文件功能，将编辑内容写入原文件；支持另存为功能，允许用户指定新的文件路径和名称；提供页面设置功能，可调整打印页面的边距等参数；实现打印功能，精确计算分页和行数确保输出质量；包含退出功能，在关闭程序时确保数据安全。编辑功能模块实现撤销操作，可以回退上一步编辑；实现重做操作，恢复被撤销的修改；提供查找功能，支持在文本中定位特定内容；提供替换功能，可批量修改文本内容；包含剪切功能，移除选中文本并存入剪贴板；包含复制功能，复制选中文本到剪贴板；包含粘贴功能，将剪贴板内容插入当前位置；实现删除功能，移除选中文本；支持转到指定行功能，快速跳转到目标行；提供全选功能，可一次性选中全部文本。格式设置模块字体选择器允许自定义字体类型；支持调整字体大小设置；可修改字体颜色，改变文字显示效果；可调整背景颜色，改变编辑区底色；提供自动换行选项，控制文本显示方式。辅助功能模块行号显示组件动态计算并显示当前行号；行号采用右对齐方式呈现；只渲染当前可见区域行号以优化性能；状态栏实时显示光标位置信息；状态栏动态更新文档字数统计；状态栏通过多线程保持实时更新；帮助系统提供使用说明文档；关于对话框显示软件版本信息。

系统不仅可作为Java教学案例演示GUI开发核心概念。实用扩展整合云存储或插件系统提升用户粘性。然而，在系统的开发和应用过程中也发现了一些问题和不足，如用户界面友好性不足，查找对话框交互复杂需简化操作流程，需要在未来的研究和开发中进一步改进。

系统的实现，在Swing GUI开发方面，我掌握了自定义组件技术，通过重写paintComponent方法优化了行号栏的渲染性能。面向对象设计能力得到加强，我设计了NotepadMainFrame主框架和MQFontChooser字体对话框，实践了模块化设计思想。在异常处理方面，我使用try-catch机制解决了文件操作中的边界问题。对Java跨平台特性有了更深理解，认识到JRE依赖的局限性。采用Git进行版本控制，有效管理代码迭代。调试技能提升，通过断点测试优化了行号计算逻辑。创造性使用FontMetrics解决行号右对齐问题，并通过多线程设计解决了状态栏卡顿问题。通过这次课程设计，我了解到了UI组件的基本使用方式，收获良多。

# 参考文献

[1]. Java从入门到精通（第三版），明日科技，清华大学出版社，2014.8

[2]. 软件工程导论，张海藩，编著，清华大学出版社

[3]. 疯狂java讲义 [J]，李刚，机械工业出版社, 2008. 17-48

[4].Java编程思想［M］.［美］埃克尔著，陈昊鹏译. 机械工业出版社,2004. 5-6