**南京邮电大学计算机学院**

**2016／2017学年第2学期**

**《JAVA程序设计》课程大作业报告**

**课题代号（单选）□课题1** √**课题2 □课题3**

**课题名称**： 简单编辑器程序

**小组成员人数（单选）□1人 □2人 □3人** √**4人 □5人**

**小组成员名单(学号和姓名）**：

1. **B15040125 姓名 袁富志**
2. **B15040102 姓名 韦露淋**
3. **B15040103 姓名 邓冠楠**
4. **B15040106 姓名 林雪迎**
5. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_姓名\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**课题实现采用的开发工具（如为其他，请填写具体工具名称）**

**□NetBeans** √**Eclipse □JDK命令行 □其他\_\_\_\_\_\_\_\_\_\_\_\_\_**

**课题实现采用的操作系统（如为其他，请填写操作系统名称）**

√**Microsoft Windows □Unix □ Linux □Mac OS □其他\_\_\_\_\_\_\_\_\_\_\_**

**课题完成时间**：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**对应提交的电子文档文件夹名称(准确填写)**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**===================以下部分为教师填写区，请勿填写=======================**

**评阅编号:（ ）－（ ）**

**成绩评定：**

**软件运行\_\_\_\_\_\_软件基本功能\_\_\_\_\_\_\_软件提高功能\_\_\_\_\_\_ 软件部分成绩**\_\_\_\_\_\_\_\_**文档结构\_\_\_\_\_\_文档理论\_\_\_\_\_\_\_\_\_\_\_文档撰写\_\_\_\_\_\_\_\_\_\_ 文档部分成绩\_\_­\_\_\_\_\_\_**

**总成绩\_\_\_\_\_\_\_\_\_\_\_\_\_**

**备注\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**简单编辑器程序设计**

# 1设计目的

# 《Java程序设计》是一门实践性很强的计算机科学与技术专业基础课程，通过本次的程序设计，一方面可以全面的复习本学期所学习的知识，另一方面可以通过实践加深学生对面向对象程序设计的理论、方法和基础知识的理解，掌握使用Java语言进行面向对象设计的基本方法，提高运用面向对象知识分析实际问题、解决实际问题的能力，提高学生的应用能力。

# 2设计任务

本次程序设计主要是设计一款类似于记事本的文本编辑器，其主要实现以下功能：

①基本要求：设计类似于Windows Notepad（记事本）文本编辑器的程序，可以创建，编辑和保存文本格式的文件。

②提高要求：在文本编辑时实现复制/粘贴功能；在文本中搜索/替换字符（字符串）；编辑Java或C/C++ 源程序时语法关键字加亮；可打开并对二进制格式的文件进行编辑和保存；在文本中以字符串形式插入当前的日期/时间。

# 3 设计内容

3.1需求分析

需求分析的任务是确定功能必须完成的工作，也就是对目标系统提出完整、准确、清晰、具体的要求。通过需求分析，我们可以清楚的了解到我们所需要的各种功能。例如，这款简单文本编辑器提供给用户基本的纯文本编辑功能，能够将用户录入的文本存储到本地磁盘中。能够读取磁盘中现有的纯文本文件，以及方便用户进行需要的编辑功能。文件操作能够实现新建、保存、打开文档等，编辑操作能过实现除了基本的文本的剪贴、复制、粘贴功能外，还可以实现在文本中查找字符串，替换单个或者全部字符串。格式操作能过实现字体设置、背景颜色等，帮助操作能够实现关于主题的查看等功能。

明白了我们需要实现的基本功能外，我们就可以设计相应的算法来一步步完成既定的功能，下面是我们的设计概要：

3.2概要设计

3.2.1程序基本功能概括

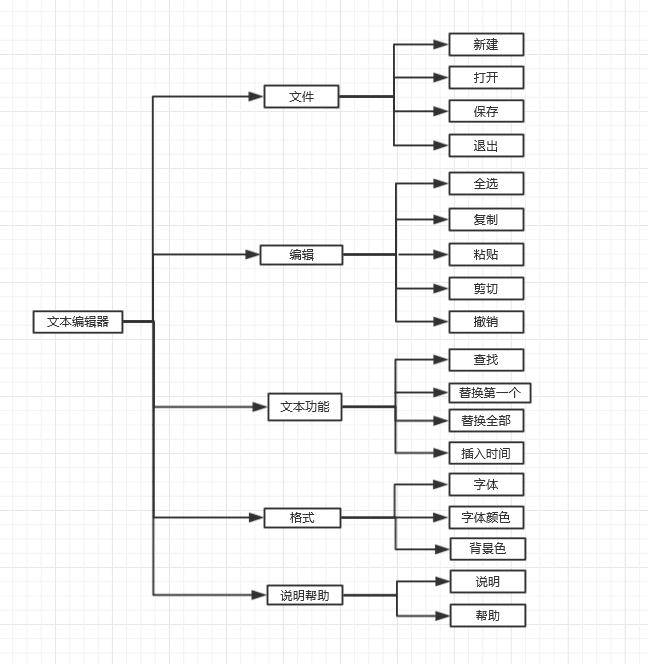


图3.2.1 功能架构图

3.2.2程序主要组件概括

1.基本的Frame框架；

2.菜单；

3.打开文件对话框；

4.保存文件对话框；

5.字体大小及颜色对话框；

6.查找及替换对话框；

7.简单的帮助与说明框架。

3.3详细设计

3.3.1主界面设计

Frame框架：这里设置了frame的标题，大小等基本的特征。

public static void main(String[] args)

{

TestDemo frame = new TestDemo();

frame.setTitle("文本编辑器");

frame.setSize(500, 500);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

frame.setLocationRelativeTo(null);

}

菜单目录：

private JMenuBar mymenubar = new JMenuBar();

private JMenu[] mymenu = { new JMenu("文件"),

new JMenu("编辑"),

new JMenu("格式"),

new JMenu("文本功能"),

new JMenu("说明&&帮助")

};

private JMenuItem[] menuitem ={

new JMenuItem("新建"),//0

new JMenuItem("打开"),//1

new JMenuItem("保存"),//2

new JMenuItem("退出"),//3

new JMenuItem("全选"),//4

new JMenuItem("复制"),//5

new JMenuItem("剪切"),//6

new JMenuItem("粘贴"),//7

new JMenuItem("撤销"),//8

new JMenuItem("字体"),//9

new JMenuItem("字体颜色"),//10

new JMenuItem("背景色"),//11

new JMenuItem("说明"),//12

new JMenuItem("帮助"),//13

new JMenuItem("查找"),//14

new JMenuItem("替换第一个"),//15

new JMenuItem("替换所有"),//16

new JMenuItem("插入当前时间"),//17

new JMenuItem("返回撤销")//18

};

利用布局管理器将菜单始终置于对话框的上端：

this.setLayout(new BorderLayout());

this.add(mymenubar,BorderLayout.NORTH);

为文本添加侦听器：

mytext.getDocument().getDocument().addUndoableEditListener(undomg); mytext.settextfont(font);

添加子菜单：

this.add(mytext,BorderLayout.CENTER);

for(i=0;i<mymenu.length;i++)

{

mymenubar.add(mymenu[i]);

}

for(i=0;i<4;i++)

{

mymenu[0].add(menuitem[i]);

}

for(i=4;i<9;i++)

{

mymenu[1].add(menuitem[i]);

}

mymenu[1].add(menuitem[18]);

for(i=9;i<12;i++)

{

mymenu[2].add(menuitem[i]);

}

for(i=12;i<14;i++)

{

mymenu[4].add(menuitem[i]);

}

for(i=14;i<18;i++)

{

mymenu[3].add(menuitem[i]);

}

为功能添加快捷键

menuitem[0].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_W,ActionEvent.CTRL\_MASK)); menuitem[1].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_O,ActionEvent.CTRL\_MASK)); menuitem[2].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_S,ActionEvent.CTRL\_MASK)); menuitem[3].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_D,ActionEvent.CTRL\_MASK)); menuitem[4].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_A,ActionEvent.CTRL\_MASK)); menuitem[5].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_C,ActionEvent.CTRL\_MASK)); menuitem[6].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_X,ActionEvent.CTRL\_MASK)); menuitem[7].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_V,ActionEvent.CTRL\_MASK)); menuitem[8].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_Z,ActionEvent.CTRL\_MASK)); menuitem[9].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_L,ActionEvent.CTRL\_MASK)); menuitem[10].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_1,ActionEvent.CTRL\_MASK)); menuitem[11].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_2,ActionEvent.CTRL\_MASK)); menuitem[12].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_M,ActionEvent.CTRL\_MASK)); menuitem[13].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_H,ActionEvent.CTRL\_MASK)); menuitem[14].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_F,ActionEvent.CTRL\_MASK)); menuitem[15].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_R,ActionEvent.CTRL\_MASK)); menuitem[16].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_T,ActionEvent.CTRL\_MASK)); menuitem[17].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_3,ActionEvent.CTRL\_MASK)); menuitem[18].setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_4,ActionEvent.CTRL\_MASK));

3.3.2文件打开与保存

定义一个文本框类，用来实现对文本的操作：

static class MyTextPanel extends JPanel

{

private JTextArea textarea = new JTextArea();

private JScrollPane scroll = new JScrollPane(textarea);

public MyTextPanel()

{

textarea.setBackground(Color.WHITE);

textarea.setLineWrap(true);

textarea.setWrapStyleWord(true);

textarea.setEditable(true);

this.setLayout(new BorderLayout());

this.add(scroll,BorderLayout.CENTER);

}

public void setTitle(String title)

{

textarea.setText(title);

}

public void clear()

{

textarea.setText("");

}

public void settextfont(Font font)

{

textarea.setFont(font);

}

public void settextcolor(Color color)

{

textarea.setForeground(color);

}

public void settextbackcolor(Color color)

{

textarea.setBackground(color);

}

public void appendtext(String string)

{

textarea.append(string);

}

public String gettext()

{

return textarea.getText();

}

public void selectall()

{

textarea.selectAll();

}

public String getselecedarea()

{

return textarea.getSelectedText();

}

public void replaceRange(String e,int i1,int i2)

{

textarea.replaceRange(e, i1, i2);

}

public int getSelectionStart()

{

return textarea.getSelectionStart();

}

public int getSelectionEnd()

{

return textarea.getSelectionEnd();

}

public void select(int start,int end)

{

textarea.select(start, end);

}

public JTextArea getDocument()

{

return textarea;

}

}

private MyTextPanel mytext = new MyTextPanel();

定义文件选择对话框，用来选择文件：

private JFileChooser jFileChooser1 = new JFileChooser(new File("."));

打开文件函数：

private void fileopen()

{

if(jFileChooser1.showOpenDialog(this)==JFileChooser.APPROVE\_OPTION)

{

open(jFileChooser1.getSelectedFile());

}

}

private void open(File file)

{

try{

BufferedInputStream input = new BufferedInputStream(new FileInputStream(file));

byte [] in =new byte[input.available()];

input.read(in,0,in.length);

mytext.clear();

mytext.appendtext(new String(in,0,in.length));

input.close();

}catch(IOException ex){

System.out.println("Sorry,Error opening"+file.getName());

}

}

保存文件函数：

private void filesave()

{

if(jFileChooser1.showSaveDialog(this)==JFileChooser.APPROVE\_OPTION)

{

save(jFileChooser1.getSelectedFile());

}

}

private void save(File file)

{

try{

BufferedOutputStream output = new BufferedOutputStream(new FileOutputStream(file));

byte [] out = (mytext.gettext()).getBytes();

output.write(out,0,out.length);

output.close();

}catch(IOException ex){

System.out.println("Sorry,Error saving"+file.getName());

}

}

3.3.3字体大小，字体颜色，背景的设置

定义相关组件的基本变量：

private String[] fontsize = new String [100];

private GraphicsEnvironment e = GraphicsEnvironment.getLocalGraphicsEnvironment();

private String[] fontnames = e.getAvailableFontFamilyNames();

private Font font = new Font("宋体",Font.BOLD,20);

private int fontstyle = Font.PLAIN;

private int fontofsize = 30;

private String fontofname = "宋体";

private JDialog dialog = new JDialog();

private JButton button = new JButton("（确定）");

private JCheckBox jchBold = new JCheckBox("Bold");

private JCheckBox jchItalic = new JCheckBox("Italic");

private JList list1 = new JList(fontsize);

private JList list2 = new JList(fontnames);

字体的设置：

System.out.println(fontnames.length);

for( i=0;i<fontsize.length;i++)

{

fontsize[i] = Integer.toString(i);

}

JPanel panel1 = new JPanel();

JPanel panel2 = new JPanel();

JPanel panel3 = new JPanel();

JPanel panel4 = new JPanel();

list1.setForeground(Color.black);

list1.setBackground(Color.WHITE);

list1.setSelectionBackground(Color.black);

list1.setSelectionForeground(Color.PINK);

list1.setVisibleRowCount(4);

list1.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

list2.setForeground(Color.black);

list2.setBackground(Color.WHITE);

list2.setSelectionBackground(Color.black);

list2.setSelectionForeground(Color.PINK);

list2.setVisibleRowCount(4);

list2.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

panel1.setLayout(new GridLayout(2,1));

panel1.add(jchBold);

panel1.add(jchItalic);

panel2.setLayout(new BorderLayout());

panel2.add(list1,BorderLayout.CENTER);

JScrollPane scrollofsize = new JScrollPane(list1);

panel2.add(scrollofsize,BorderLayout.CENTER);

panel3.setLayout(new FlowLayout(FlowLayout.CENTER,120,60));

panel3.add(button);

panel4.setLayout(new BorderLayout());

panel4.add(list2,BorderLayout.CENTER);

JScrollPane scrollofname = new JScrollPane(list2);

panel4.add(scrollofname,BorderLayout.CENTER);

dialog.setLayout(new GridLayout(1,4,20,20));

dialog.add(panel1);

dialog.add(panel2);

dialog.add(panel4);

dialog.add(panel3);

dialog.setSize(600,150);

jchBold.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

setNewFont();

}

});

jchItalic.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

setNewFont();

}

});

button.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

font = new Font(fontofname,fontstyle,fontofsize);

mytext.settextfont(font);

}

});

字体及字号选择：

list1.addListSelectionListener(new ListSelectionListener(){

public void valueChanged(ListSelectionEvent e) {

// TODO Auto-generated method stub

int indice = list1.getSelectedIndex();

fontofsize = indice;

}

});

list2.addListSelectionListener(new ListSelectionListener(){

public void valueChanged(ListSelectionEvent e) {

int indice = list2.getSelectedIndex();

fontofname = fontnames[indice];

}

});

打开字体选择对话框：

menuitem[9].addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0) {

dialog.setVisible(true);

}

})

打开字体颜色对话框：

menuitem[10].addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0) {

Color selectedColor = JColorChooser.showDialog(null, "choose color", mytext.getForeground());

if(selectedColor!=null)

mytext.settextcolor(selectedColor);

}

});

打开背景颜色对话框：

menuitem[11].addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0) {

Color selectedColor = JColorChooser.showDialog(null, "choose color", mytext.getForeground());

if(selectedColor!=null)

mytext.settextbackcolor(selectedColor);

}

});

3.3.4剪切，复制，粘贴设置

定义相关的组件：

private UndoManager undomg = new UndoManager();

用于查找的组件：

private JDialog finddialog = new JDialog();

private JTextField findtextfield = new JTextField();

private JButton findbutton = new JButton("（查找）");

private int findstart = 0;

private JCheckBox fitBigSmall = new JCheckBox();

boolean fitorunfit = false;

用于替换第一个字符的组件：

private JDialog replaceFirstdialog = new JDialog();

private JTextField changedtextfield = new JTextField();

private JTextField destinationtextfield = new JTextField();

private JButton replacebutton = new JButton("（替换）");

private int replacestart = 0;

用于替换所有查询字符的组件：

private JDialog replacedialog = new JDialog();

private JTextField changedtextfield\_all = new JTextField();

private JTextField destinationtextfield\_all = new JTextField();

private JButton replacebutton\_all = new JButton("（替换）");

private int replacestart\_all = 0;

用于剪切的组件：

Toolkit toolkit=Toolkit.getDefaultToolkit();//获取默认工具包。

Clipboard clipBoard = toolkit.getSystemClipboard(); //获取系统剪切板

查找字符的函数：

if(fitorunfit==false)//大小写不匹配

{

String findstring = findtextfield.getText();//寻找的字符串

String allstring = mytext.gettext();

System.out.println(allstring.indexOf(findstring, findstart));

mytext.select(allstring.indexOf(findstring, findstart), allstring.indexOf(findstring, findstart)+findstring.length());//选中目标

findstart = allstring.indexOf(findstring, findstart)+1; / /findstart做相应的调整

}

else//大小写匹配

{

String findstring = findtextfield.getText();//

String allstring = mytext.gettext();

String smallfindstring = findstring.toLowerCase();

String smallallstring = allstring.toLowerCase();

System.out.println(smallallstring.indexOf(smallfindstring, findstart));

mytext.select(smallallstring.indexOf(smallfindstring, findstart), smallallstring.indexOf(smallfindstring, findstart)+smallfindstring.length());

findstart = smallallstring.indexOf(smallfindstring, findstart)+1;

}

}

});

替换单个字符的函数：

String changedstring = changedtextfield.getText();

String destinationstring = destinationtextfield.getText();

String allstring = mytext.gettext();

if(allstring.indexOf(changedstring, replacestart)!=-1)//找到

{

mytext.select(allstring.indexOf(changedstring, replacestart), allstring.indexOf(changedstring, replacestart)+changedstring.length());

mytext.replaceRange(destinationstring, allstring.indexOf(changedstring, replacestart), allstring.indexOf(changedstring, replacestart)+changedstring.length());

}

else

{

JOptionPane.showMessageDialog(null, "没有找到","警告",JOptionPane.WARNING\_MESSAGE);

}

replacestart = 0;

}

});

替换所有的函数：

String changedstring = changedtextfield\_all.getText();

String destinationstring = destinationtextfield\_all.getText();

String allstring = mytext.gettext();

boolean temp = true;

while(allstring.indexOf(changedstring, replacestart\_all)!=-1)

{ temp = false;

allstring = mytext.gettext();

mytext.select(allstring.indexOf(changedstring, replacestart\_all), allstring.indexOf(changedstring, replacestart\_all)+changedstring.length());

int i = allstring.indexOf(changedstring, replacestart\_all);

mytext.replaceRange(destinationstring, i, i+changedstring.length());

replacestart\_all = i+1;

}

if(temp == true)

{

JOptionPane.showMessageDialog(null, "没有找到","警告",JOptionPane.WARNING\_MESSAGE);

}

replacestart\_all = 0;

}

});

插入当前时间：

Date date=new Date();

//设置要获取到什么样的时间

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

//获取String类型的时间

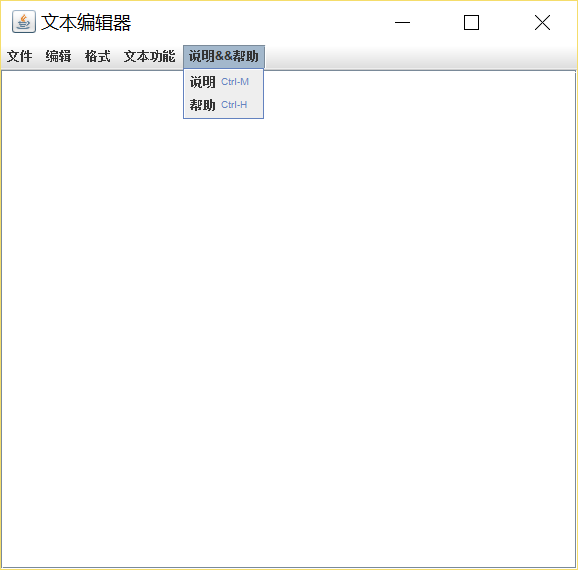
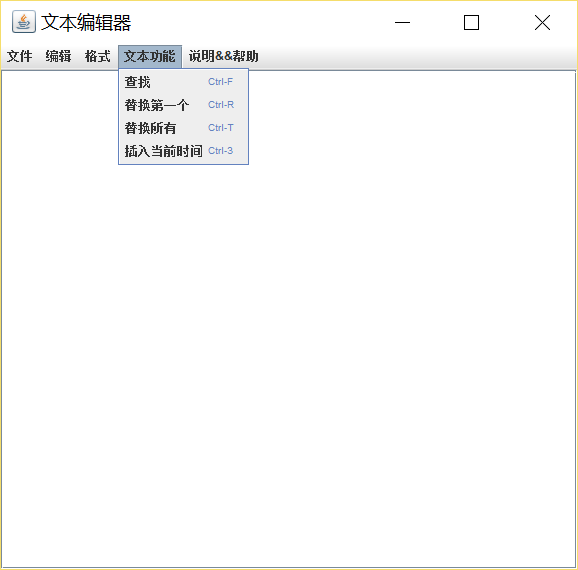
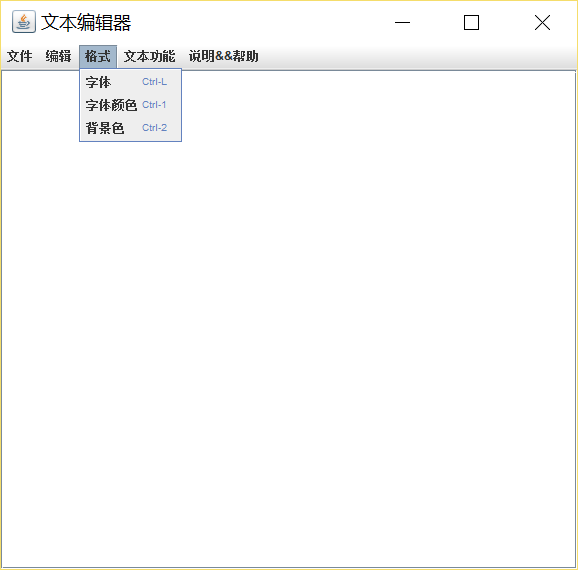
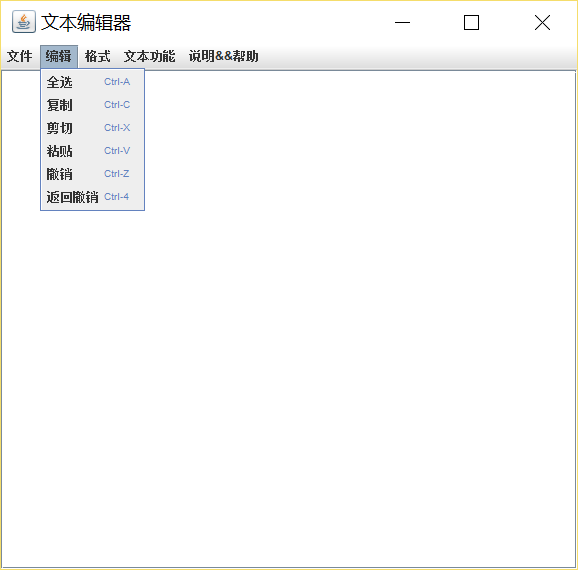
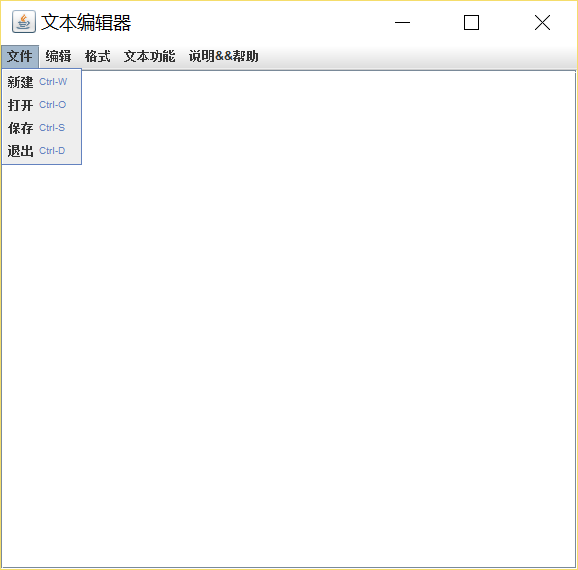
String createdate = sdf.format(date);

mytext.appendtext(createdate);

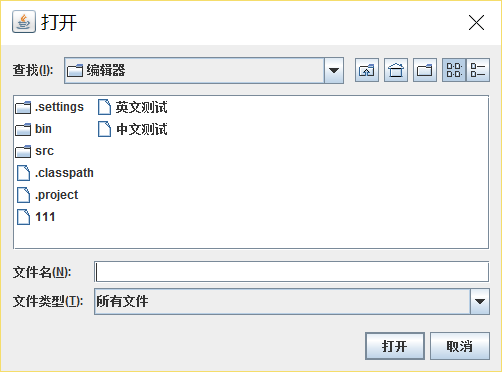
3.4设计成果

3.4.1运行界面

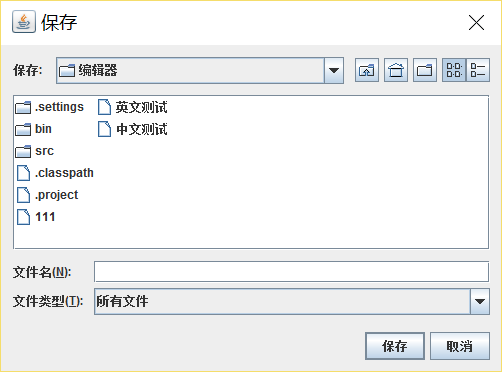
主界面展示：



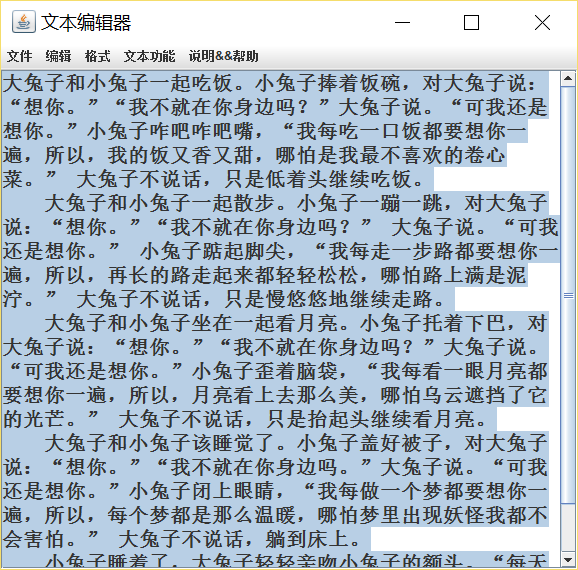
打开文件操作：



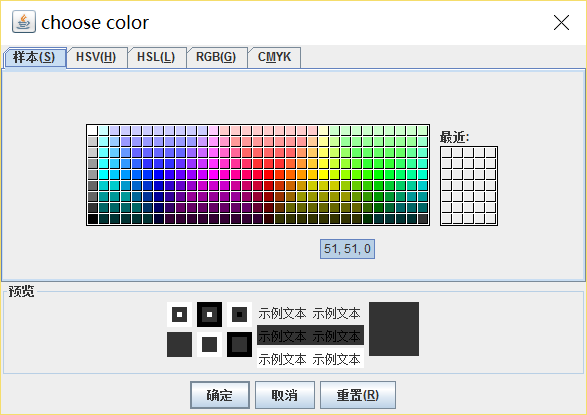
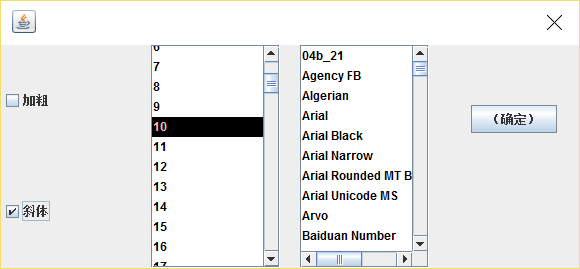
保存文件操作：



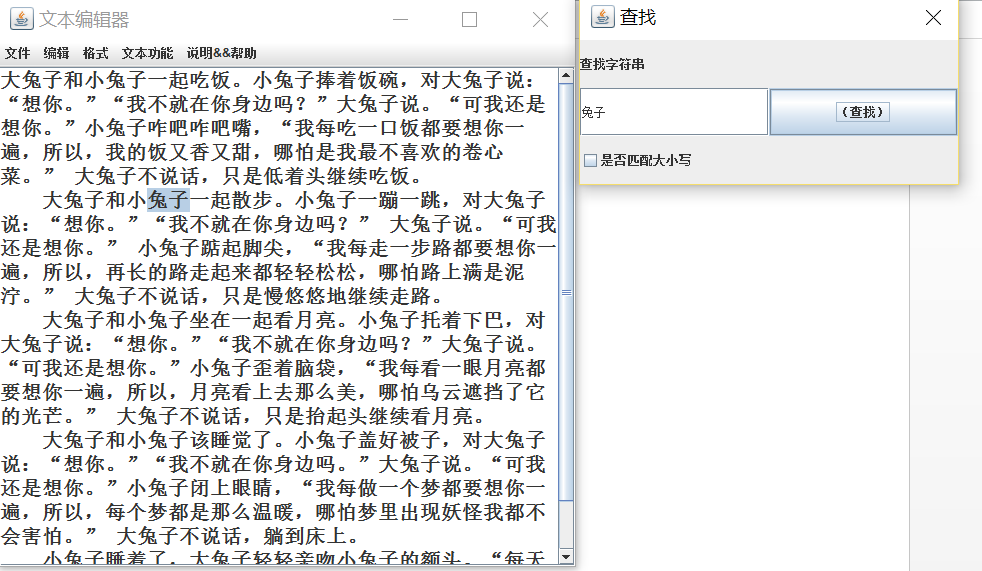
全选操作：



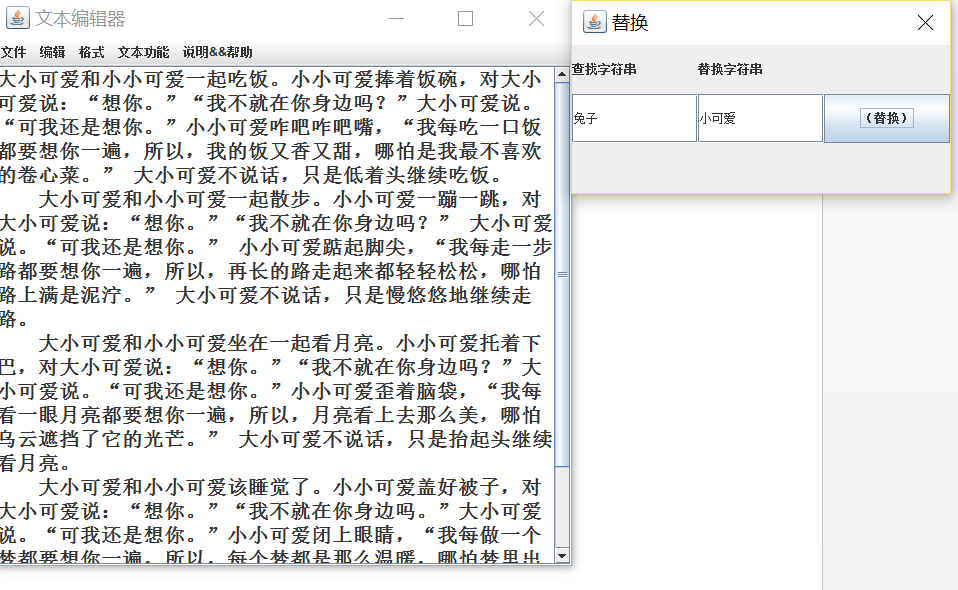
字体与颜色：



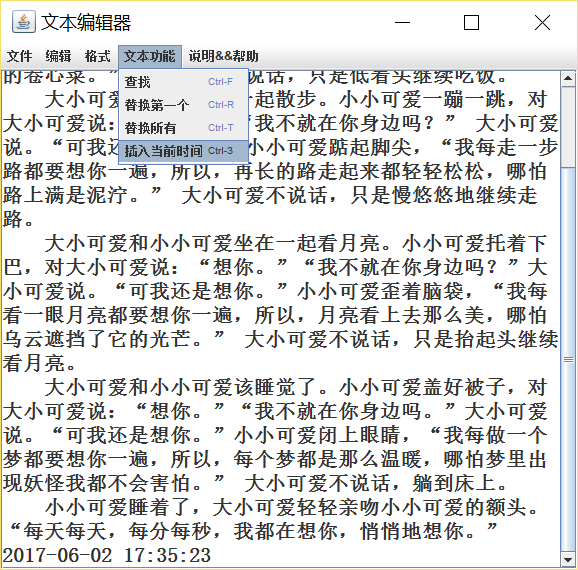
查找操作：



替换操作：



插入当前时间：



3.4.2运行结果

将程序类文件转化成EXE文件，打开后在编辑器中输入一段源代码，进行查找，保存，编译，执行。以上所有测试的结果都跟windows操作系统的记事本进行了比较，功能实现方面都基本正确无误。

程序实现了打开、新建、保存一个文本文件；对选中的文本进行各种编辑操作（设置字体、字号、字型、对齐方式、背景、前景色、复制、粘贴、剪切、查找、替换等）；在文本中能够插入对象等基本功能。

**4 总结与展望**

在本次的程序设计中，我们组设计的文本编辑器基本实现了打开, 新建,保存,复制,粘贴,剪贴，以及查找等功能。完成这个编辑器，我们首先在主类中添加一系列组件,如Button,JtextArea,Menu等,再对相应的组件添加监听,然后在各自对应的方法中来实现各自的功能。

通过本次的程序设计，让我们对Java面向对象编程有了更进一步的认识，大大拓展了我们思维方式，尤其是在设计中用到的一些新技术是对我们的一个挑战。由于有一些要用到的只是我们在课本中还没有学到，所以我们要自己去查阅一些资料来完成所需要的功能。

与此同时，通过本次的程序设计让我们对软件开发的有关知识有了一个更深，更全面的认识，也使我们很好的体验了从提出问题到问题定义，总体设计，详细设计，最后到编码调试程序等一系列过程中的设计过程。

由于时间和水平有限，本次设计的文本编辑器虽然实现了一些基本的操作，但是还存在一些不完善的地方。我们最后还是没有能够实现在编辑Java或C/C++ 源程序时语法关键字加亮。尽管我们没能够实现语法高亮，但是我们我们也找出了问题所在，由于我们使用的文本框是JtextArea,只能够对已存在的文本实现关键字加亮，无法对输入的文本实现高亮，想要实现要把JTextArea换成JTextPane,但是一旦更换，我们之前基于JTextArea实现的一些函数就都要重新定义实现，会带来很大的问题，由于时间问题我们无法来实现这个功能。

**附录一：**

参考材料及文献：

【1】(美)阿诺德，Ken Arnold，等. Java程序设计语言[M]. 北京：人民邮电出版社，2006；

【2】 耿祥义,张跃平. JAVA实用教程[M]. 清华大学出版社，2003.5；

【3】JFrame,JPanel,JLabel详解：<http://blog.csdn.net/qq_33860226/article/details/51489786>；

【4】Java文件操作解析; <http://blog.csdn.net/jiangxinyu/article/details/7885518/>;

【5】JavaGUI菜单设计：<http://blog.csdn.net/zhouzixin053/article/details/21476271>；

**附录二：成员面单及联系方式**

|  |  |  |  |
| --- | --- | --- | --- |
| 小组成员姓名和联系方法 | | | |
| 姓名 | 成员 | 联系方式 | 邮箱 |
| 袁富志 | 组长 | 18260090763 | 1552800131@qq.com |
| 韦露淋 | 组员 | 18362939963 | 1730980459@qq.com |
| 邓冠楠 | 组员 | 18251925922 | 532060610@qq.com |
| 林雪迎 | 组员 | 18260083996 | 3161392202@qq.com |

**附录三：意见及建议**

希望老师以后讲课的过程中尽量多演示一些基础的实例，因为很多同学都是零基础学习Java的，虽然有一定的C++功底，但是只是学习那些理论的知识，平时上课会显得很枯燥，希望老师以后的讲课过程中，可以多运行一些实用性的实例。