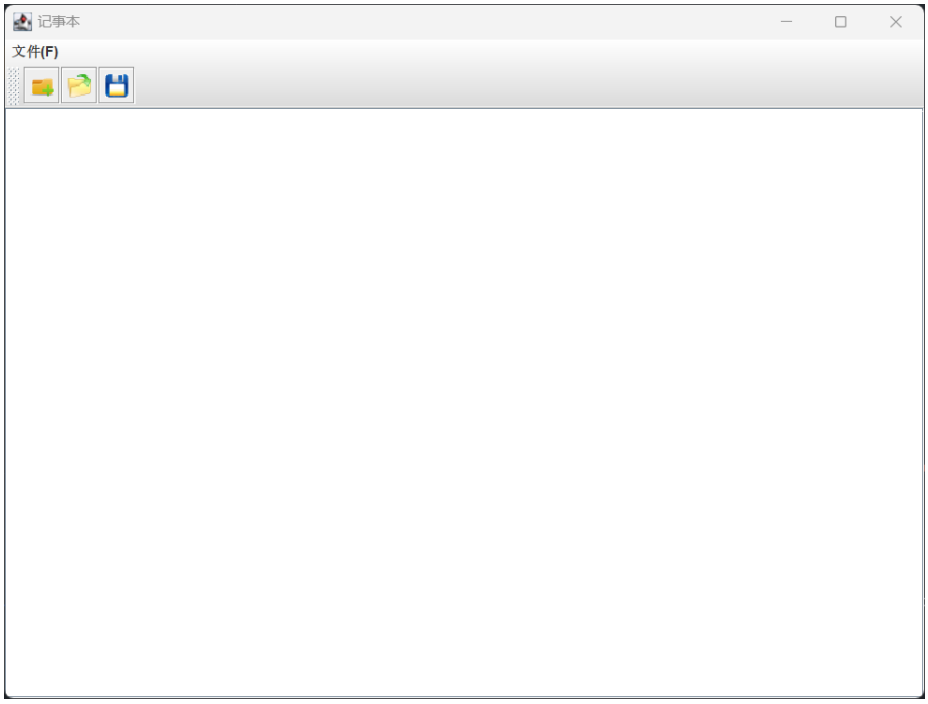
说明文档



主界面：使用了 JmenuBar 创建菜单，JtoolBar 创建工具栏

```
JToolBar toolBar = new JToolBar();
JButton newButton = null;
JButton openButton = null;
JButton saveButton = null;
try {
    Image newImg = ImageIO.read(new File( pathname: "E:\\java\\Text\\src\\img\\new.png")).getScaledInstance( width: 20,
    newButton = new JButton(new ImageIcon(newImg));
    toolBar.add(newButton);

    Image openImg = ImageIO.read(new File( pathname: "E:\\java\\Text\\src\\img\\open.png")).getScaledInstance( width: 20,
    openButton = new JButton(new ImageIcon(openImg));
    toolBar.add(openButton);

    Image saveImg = ImageIO.read(new File( pathname: "E:\\java\\Text\\src\\img\\save.png")).getScaledInstance( width: 20,
    saveButton = new JButton(new ImageIcon(saveImg));
    toolBar.add(saveButton);
} catch (IOException e) {
    e.printStackTrace();
}
add(toolBar, BorderLayout.NORTH);
```

工具栏部分代码

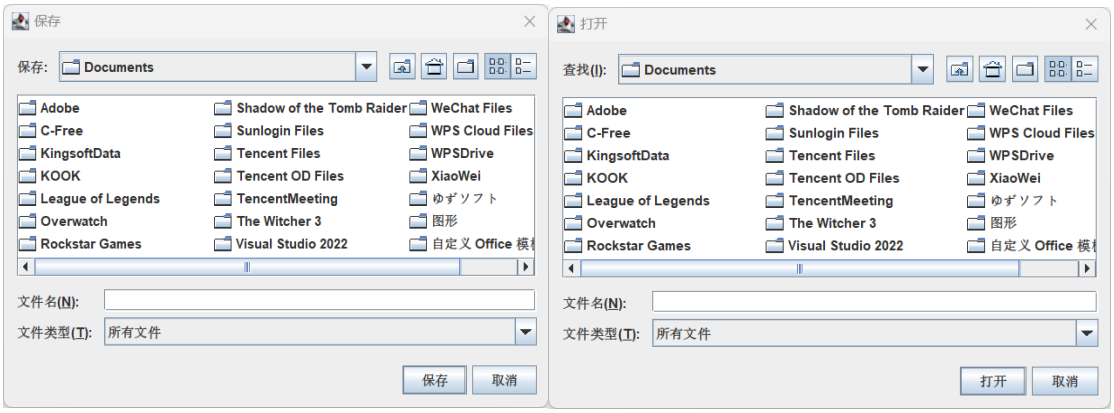
字体选择框：通过 JDialog

```java
public FontChooser(JFrame parent, Font currentFont) {
    super(parent, title: "字体选择", modal: true);
    setLayout(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();

    // 字体名称标签和下拉框
    JLabel fontNameLabel = new JLabel( text: "字体名称:");
    fontNameComboBox = new JComboBox<>(GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames(
    fontNameComboBox.setSelectedItem(currentFont.getName());
    gbc.insets = new Insets( top: 4, left: 4, bottom: 4, right: 4); // 设置组件之间的间距

    gbc.gridx = 0;
    gbc.gridy = 0;
    gbc.gridwidth = 1;
    gbc.anchor = GridBagConstraints.EAST;
    add(fontNameLabel, gbc);

    gbc.gridx = 1;
    gbc.gridy = 0;
    gbc.gridwidth = 2;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    add(fontNameComboBox, gbc);

    // 字体大小标签和下拉框
```

相关代码

```java
    // 字体大小标签和下拉框
    JLabel fontSizeLabel = new JLabel( text: "字体大小:");
    Integer[] sizes = {8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40};
    fontSizeComboBox = new JComboBox<>(sizes);
    fontSizeComboBox.setSelectedItem(currentFont.getSize());

    gbc.gridx = 0;
    gbc.gridy = 1;
    gbc.gridwidth = 1;
    gbc.anchor = GridBagConstraints.EAST;
    add(fontSizeLabel, gbc);

    gbc.gridx = 1;
    gbc.gridy = 1;
    gbc.gridwidth = 2;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    add(fontSizeComboBox, gbc);

    // 确定按钮
    okButton = new JButton( text: "确定");
    okButton.addActionListener(e -> {
        String fontName = (String) fontNameComboBox.getSelectedItem();
        int fontSize = (Integer) fontSizeComboBox.getSelectedItem();
        selectedFont = new Font(fontName, Font.PLAIN, fontSize);
        setVisible(false);
    });
    gbc.gridx = 1;
    gbc.gridy = 2;
    gbc.gridwidth = 1;
```

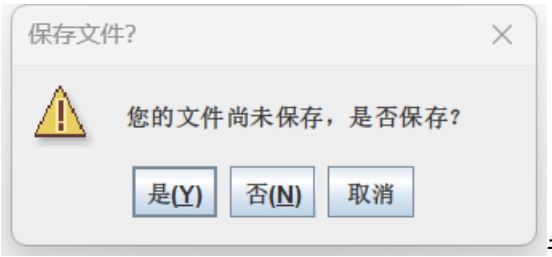
保存和打开文件界面：



相关代码有事件监听器和相应方法的具体实现：

```
    // 事件监听
    newItem.addActionListener(e -> textArea.setText(""));
    if (newButton != null) {
        newButton.addActionListener(e -> textArea.setText(""));
    }

    openItem.addActionListener(e -> openFile());
    if (openButton != null) {
        openButton.addActionListener(e -> openFile());
    }

    saveItem.addActionListener(e -> saveFile());
    if (saveButton != null) {
        saveButton.addActionListener(e -> saveFile());
    }

    exitItem.addActionListener(e -> System.exit( status: 0));
}
```

```
2 个用法
private void openFile() {
    if (fileChooser.showOpenDialog( parent: this) == JFileChooser.APPROVE_OPTION) {
        File file = fileChooser.getSelectedFile();
        try {
            BufferedReader reader = new BufferedReader(new FileReader(file));
            textArea.read(reader, desc: null);
            reader.close();
        } catch (IOException ex) {
            JOptionPane.showMessageDialog( parentComponent: this, message: "文件打开失败");
        }
    }
}

3 个用法
private void saveFile() {
    if (fileChooser.showSaveDialog( parent: this) == JFileChooser.APPROVE_OPTION) {
        File file = fileChooser.getSelectedFile();
        try {
            BufferedWriter writer = new BufferedWriter(new FileWriter(file));
            textArea.write(writer);
            writer.close();
        } catch (IOException ex) {
            JOptionPane.showMessageDialog( parentComponent: this, message: "文件保存失败");
        }
    }
```



关闭提示窗口：

1.为 JTextArea 添加一个键盘监听器来监测按键事件，从而设置文本已修改的标志。

2.在 JFrame 的关闭操作中加入检查，如果文本已修改，则提示用户。