Java 语言程序设计 作业 4-Java GUI

封钰震 (1951362)

2021 年 11 月 15 日

1. 编程环境

硬件环境

- 型号名称：MacBook Pro
- 处理器名称：Dual-Core Intel Core i5
- 内存：8 GB

软件环境

- 操作系统：macOS 10.15.7

运行环境

- JDK 14.0.2

2. 设计思想

窗口布局

首先从`JFrame`继承一个名为`TextComponentFrame`类作为我们的窗口容器。在这个`TextComponentFrame`中包括 2 个`JPanel`面板容器（分别名为`filePanel`和`stylePanel`）和 1 个`JScrollPane`滚动面板容器（名为`scrollPane`）。

`filePanel`面板容器分别容纳与加载和保存文件有关的组件（用于第一问），包括输入文件名的文本域`fileNameField`、“加载文件”按钮`loadFileButton`、“保存文件”按钮`saveFileButton`。

stylePanel面板容器分别容纳与更改、加载和保存样式有关的组件（用于第二、三问），包括用于选择样式的组合框、滑动条，输入样式文件名的文本域styleFileNameField，“更改样式”按钮changeStyleButton，“加载样式”按钮loadStyleButton，“保存样式”按钮saveStyleButton。

scrollPane滚动面板容器包含了一个文本区contentArea。

事件触发

对于窗口中的每一个按钮，调用addActionListener方法，设置一个监听器。每一个监听器都是一个内部类，实现了ActionListener接口。当按钮被按下时，监听器的actionPerformed方法就会被调用。

对于“加载文件”按钮loadFileButton，被点击后，若文件存在，则会设置文本区contentArea的内容为文件内容；若文件不存在，则会弹出弹窗询问用户是否需要新建文件，若用户点击“是”，则新建文件。

对于“保存文件”按钮saveFileButton，被点击后，则保存contentArea的内容至fileNameField路径文件下。

对于“更改样式”按钮changeStyleButton，被点击后，则设置窗口样式为用户选择的样式。

对于“加载样式”按钮loadStyleButton，被点击后，若文件存在，则会加载文件中的样式属性，并设置为窗口样式；若文件不存在，则会弹出弹窗提示用户。

对于“保存样式”按钮saveStyleButton，被点击后，则保存当前用户设置的样式至styleFileNameField路径文件下。

样式文件

样式文件的示例如下：

```
#Style Settings
#Sun Nov 14 23:33:00 CST 2021
loc=\u4E0A\u4E0B\u5E03\u5C40
size=36
fontStyle=0
font=Segoe UI
```

其中，第 1-2 行为注释，第 3 行为组件布局（Unicode 编码），第 4 行为字体大小，第 5 行为字型，第 6 行为字体。

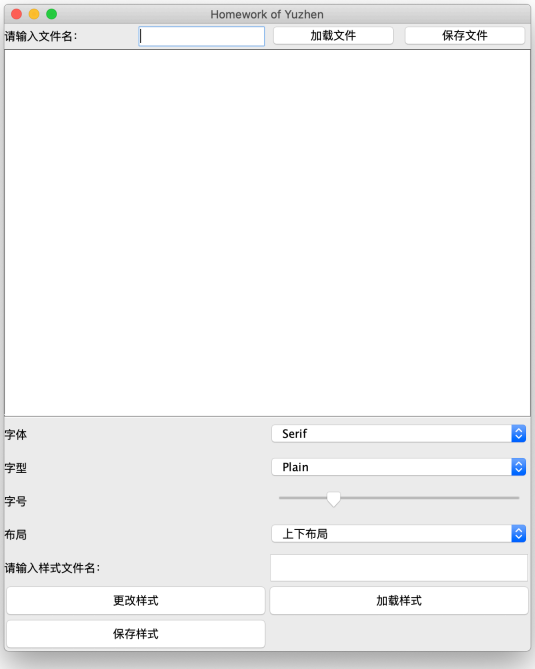

图 1: 窗口

3. 执行过程

完整的窗口如图1所示。

第一问

首先，在文本域中输入一个不存在的文件路径，点击“加载文件”。此时，会弹窗提示“文件不存在，是否要新建文件?”，如图2所示。点击“是”后，创建文件。在文件中输入内容，点击“保存文件”，则弹窗提示“文件保存成功!”，如图3所示。输入一个存在的文件路径，重新“加载文件”，弹窗提示“文件加载成功!”，如图4所示。然后看到文本区中加载了文件中的内容，如图5所示。

第二问

可以通过`stylePanel`面板容器中的组件调整样式，如图6和图7所示。

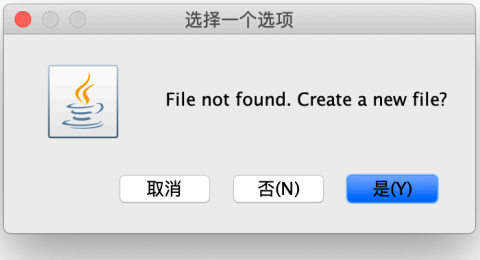

图 2: 文件不存在

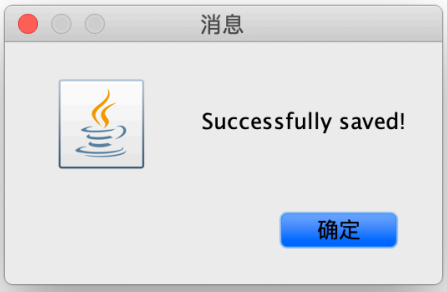

图 3: 文件保存成功

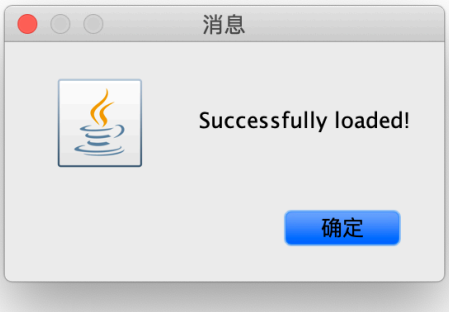

图 4: 文件加载成功

图 5: 文件存在

第三问

若要保存当前样式，则输入配置文件保存路径，点击“保存样式”，则将样式保存到目标路径中。对于图7中的样式，配置文件即为：

```
#Style Settings
#Mon Nov 15 10:59:12 CST 2021
loc=\u5DE6\u53F3\u5E03\u5C40
size=58
fontStyle=1
font=Comic Sans MS
```

若要加载配置文件中的样式，则输入其路径，点击“加载样式”即可。

4. 窗口类源代码

```
class TextComponentFrame extends JFrame
{
```

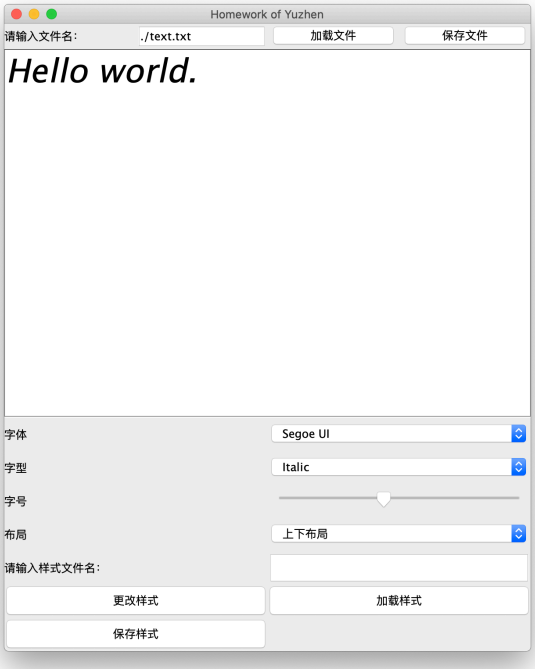

图 6: 风格切换 1

图 7: 风格切换 2

```
    private static final int WIDTH = 600;
    private static final int HEIGHT = 800;

    private JPanel filePanel;
    private JTextField fileNameField;
    private JButton loadFileButton;
    private JButton saveFileButton;

    private JScrollPane scrollPane;
    private JTextArea contentArea;

    private JPanel stylePanel;
    private JComboBox<String> fontOpt;
    private JComboBox<String> fontStyleOpt;
    private JSlider sizeOpt;
    private JComboBox<String> locOpt;
    private JTextField styleFileNameField;
    private JButton changeStyleButton;
    private JButton loadStyleButton;
    private JButton saveStyleButton;

    private void setStyle(String font, int fontStyle, int size,
       String loc)
    {
        contentArea.setFont(new Font(font, fontStyle, size));
        switch (loc) {
            case "左右布局": {
                getLayout().removeLayoutComponent(filePanel);
                getLayout().removeLayoutComponent(stylePanel);
                setSize(HEIGHT, WIDTH);
                filePanel.setLayout(new GridLayout(13, 1));
                stylePanel.setLayout(new GridLayout(13, 1));
                add(filePanel, BorderLayout.WEST);
                add(stylePanel, BorderLayout.EAST);
                add(scrollPane, BorderLayout.CENTER);
```

```
                break;
            }
            case "上下布局": {
                getLayout().removeLayoutComponent(filePanel);
                getLayout().removeLayoutComponent(stylePanel);
                setSize(WIDTH, HEIGHT);
                filePanel.setLayout(new GridLayout(1, 4));
                stylePanel.setLayout(new GridLayout(7, 2));
                add(filePanel, BorderLayout.NORTH);
                add(stylePanel, BorderLayout.SOUTH);
                add(scrollPane, BorderLayout.CENTER);
                break;
            }
        }
    }

    public TextComponentFrame()
    {
        contentArea = new JTextArea();
        contentArea.setLineWrap(true);
        scrollPane = new JScrollPane(contentArea);

        filePanel = new JPanel();
        filePanel.setLayout(new GridLayout(1, 4));
        fileNameField = new JTextField();
        loadFileButton = new JButton("加载文件");
        saveFileButton = new JButton("保存文件");
        filePanel.add(new JLabel("请输入文件名: "));
        filePanel.add(fileNameField);
        filePanel.add(loadFileButton);
        filePanel.add(saveFileButton);

        stylePanel = new JPanel();
        stylePanel.setLayout(new GridLayout(7, 2));
        fontOpt = new JComboBox<>(new String[]{"Serif", "Agency FB",
```

```
        "Arial", "Calibri", "Cambrian",
            "Century Gothic", "Comic Sans MS", "Courier New",
                "Forte", "Garamond",
            "Monospaced", "Segoe UI", "Times New Roman",
                "Trebuchet MS"});
    fontStyleOpt = new JComboBox<>(new String[]{"Plain", "Bold",
        "Italic"});
    sizeOpt = new JSlider(10, 72, 24);
    sizeOpt.setPaintLabels(true);
    sizeOpt.setPaintTicks(true);
    locOpt = new JComboBox<>(new String[]{"上下布局",
        "左右布局"});
    styleFileNameField = new JTextField();
    changeStyleButton = new JButton("更改样式");
    loadStyleButton = new JButton("加载样式");
    saveStyleButton = new JButton("保存样式");
    stylePanel.add(new JLabel("字体"));
    stylePanel.add(fontOpt);
    stylePanel.add(new JLabel("字型"));
    stylePanel.add(fontStyleOpt);
    stylePanel.add(new JLabel("字号"));
    stylePanel.add(sizeOpt);
    stylePanel.add(new JLabel("布局"));
    stylePanel.add(locOpt);
    stylePanel.add(new JLabel("请输入样式文件名: "));
    stylePanel.add(styleFileNameField);
    stylePanel.add(changeStyleButton);
    stylePanel.add(loadStyleButton);
    stylePanel.add(saveStyleButton);

    add(filePanel, BorderLayout.NORTH);
    add(stylePanel, BorderLayout.SOUTH);
    add(scrollPane, BorderLayout.CENTER);
    pack();
```

```
        loadFileButton.addActionListener(new LoadFileAction());
        saveFileButton.addActionListener(new SaveFileAction());
        changeStyleButton.addActionListener(new ChangeStyleAction());
        loadStyleButton.addActionListener(new LoadStyleAction());
        saveStyleButton.addActionListener(new SaveStyleAction());

        setSize(WIDTH, HEIGHT);
        setLocationByPlatform(true);
    }

    private class LoadFileAction implements ActionListener
    {
        @Override
        public void actionPerformed(ActionEvent e)
        {
            String fileName = fileNameField.getText();
            String content;
            try {
                BufferedReader reader = new BufferedReader(new
                   FileReader(fileName));
                content = reader.lines().collect
                    (Collectors.joining(System.lineSeparator()));
                reader.close();
                JOptionPane.showMessageDialog(contentArea,
                   "Successfully loaded!");
            } catch (FileNotFoundException ex)
            {
                int option =
                   JOptionPane.showConfirmDialog(contentArea, "File
                   not found. Create a new file?");
                if (option == JOptionPane.YES_OPTION)
                {
                    try {
                        (new File(fileName)).createNewFile();
                    } catch (IOException ex1) {
```

```
                    JOptionPane.showMessageDialog(contentArea,
                        ex1.getMessage());
                }
            }
            content = "";
        } catch (Exception ex)
        {
            content = "";
            JOptionPane.showMessageDialog(contentArea,
                ex.getMessage());
        }
        contentArea.setText(content);
    }
}

private class SaveFileAction implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e) {
        String fileName = fileNameField.getText();
        BufferedWriter writer;
        try {
            writer = new BufferedWriter(new
                FileWriter(fileName));
            writer.write(contentArea.getText());
            writer.close();
            JOptionPane.showMessageDialog(contentArea,
                "Successfully saved!");
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(contentArea,
                ex.getMessage());
        }
    }
}
```

```
private class ChangeStyleAction implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e) {
        String font = (String) fontOpt.getSelectedItem();
        int fontStyle = fontStyleOpt.getSelectedIndex();
        int size = sizeOpt.getValue();
        String loc = (String) locOpt.getSelectedItem();
        setStyle(font, fontStyle, size, loc);
    }
}

private class LoadStyleAction implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        String fileName = styleFileNameField.getText();
        Properties properties = new Properties();
        try {
            FileInputStream reader = new
               FileInputStream(fileName);
            properties.load(reader);
            reader.close();
            JOptionPane.showMessageDialog(contentArea,
               "Successfully loaded!");
            String font = properties.getProperty("font",
               "Serif");
            int fontStyle =
               Integer.parseInt(properties.getProperty("fontStyle",
               "0"));
            int size =
               Integer.parseInt(properties.getProperty("size",
               "24"));
            String loc = properties.getProperty("loc",
```

```
                "上下布局");
            setStyle(font, fontStyle, size, loc);
            fontOpt.setSelectedItem(font);
            fontStyleOpt.setSelectedItem(fontStyle);
            sizeOpt.setValue(size);
            locOpt.setSelectedItem(loc);
        } catch (Exception ex)
        {
            JOptionPane.showMessageDialog(contentArea,
                ex.getMessage());
        }
    }
}

private class SaveStyleAction implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent e) {
        String fileName = styleFileNameField.getText();
        Properties properties = new Properties();
        properties.setProperty("font", (String)
            fontOpt.getSelectedItem());
        properties.setProperty("fontStyle",
            String.valueOf(fontStyleOpt.getSelectedIndex()));
        properties.setProperty("size",
            String.valueOf(sizeOpt.getValue()));
        properties.setProperty("loc", (String)
            locOpt.getSelectedItem());
        try {
            FileOutputStream writer = new
                FileOutputStream(fileName);
            properties.store(writer, "Style Settings");
            writer.close();
            JOptionPane.showMessageDialog(contentArea,
                "Successfully saved!");
```

```
                } catch (IOException ex) {
                    JOptionPane.showMessageDialog(contentArea,
                        ex.getMessage());
                }
            }
        }
    }
```