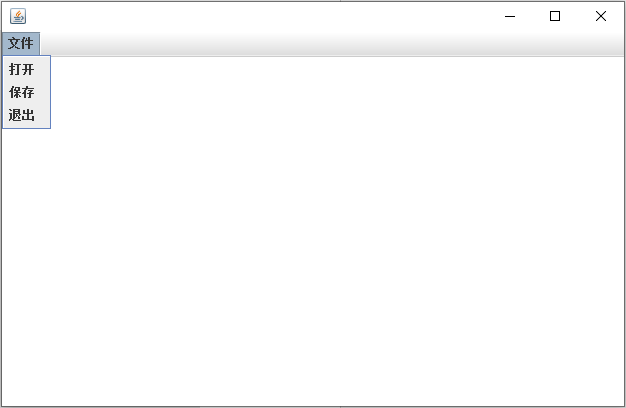
【案例11-1】 简易记事本

**【案例介绍】**

**1.案例描述**

本案例要求利用Java Swing 图形组件开发一个图形化简易记事本。记事本功能包括文本编辑、保存文本到指定路径、打开指定路径下的文本、退出等。

**2.运行结果**



运行结果

**【案例目标】**

* 学会分析“简易计算器”程序实现的逻辑思路。
* 能够独立完成“简易计算器”程序的源代码编写、编译及运行。
* 掌握Java Swing界面编程的应用。

**【案例分析】**

（1）记事本界面整体可以看做是一个容器窗口。

（2）从运行结果中的记事本界面的布局效果显示内容可以看出，该图形化界面有最小化、放大缩小以及关闭按钮，以及菜单栏、菜单、文本域。菜单栏可以使用JMenuBar实现，文本域可以使用JTextArea来实现，菜单可以使用JMenu来实现，菜单项可以使用JMenuItem来实现。

（3）为了使窗口可以最小化、放大缩小、关闭还必须为这些按钮注册监听器，进行相应的事件处理。

（4）定义一个程序入口，用于启动Swing案例程序。

**【案例实现】**

MyMenu.java

1. import java.awt.\*;
2. import java.awt.event.ActionEvent;
3. import java.awt.event.ActionListener;
4. import java.io.\*;
5. import javax.swing.\*;
6. class MyNotePad extends JFrame implements ActionListener {
7. private JTextArea jta = null;
8. private JMenuBar jmb = null;
9. private JMenu jm = null;
10. private JMenuItem jmiOpen = null;
11. private JMenuItem jmiSave = null;
12. private JMenuItem jmiExit = null;
13. private JFileChooser jfc = null;
14. public MyNotePad() {
15. jta = new JTextArea();
16. this.setLayout(new BorderLayout());
17. this.add(jta);
18. jmb = new JMenuBar();
19. jm = new JMenu("文件");
20. jmiOpen = new JMenuItem("打开");
21. jmiOpen.addActionListener(this);
22. jmiOpen.setActionCommand("打开");
23. jmiSave = new JMenuItem("保存");
24. jmiSave.addActionListener(this);
25. jmiSave.setActionCommand("保存");
26. jmiExit = new JMenuItem("退出");
27. jmiExit.addActionListener(this);
28. jmiExit.setActionCommand("退出");
29. jm.add(jmiOpen);
30. jm.add(jmiSave);
31. jm.add(jmiExit);
32. jmb.add(jm);
33. this.setJMenuBar(jmb);
34. this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
35. this.setSize(400, 300);
36. this.setVisible(true);
37. }
38. @Override
39. public void actionPerformed(ActionEvent e) {
40. // TODO Auto-generated method stub
41. String str = e.getActionCommand();
42. if (str.equals("打开")) {
43. System.out.println("打开");
44. jfc = new JFileChooser();
45. jfc.setDialogTitle("请选择文件！");
46. jfc.showOpenDialog(null);
47. jfc.setVisible(true);
48. File file = jfc.getSelectedFile();
49. BufferedReader br = null;
50. try {
51. FileReader fReader = new FileReader(file);
52. br = new BufferedReader(fReader);
53. String readStr = "";
54. String allCode = "";
55. while ((readStr = br.readLine()) != null) {
56. allCode += readStr + "\r\n";
57. }
58. jta.setText(allCode);
59. } catch (Exception e2) {
60. e2.printStackTrace();
61. // TODO: handle exception
62. } finally {
63. try {
64. br.close();
65. } catch (IOException e1) {
66. // TODO Auto-generated catch block
67. e1.printStackTrace();
68. }
69. }
70. } else if (str.equals("保存")) {
71. JFileChooser jfc = new JFileChooser();
72. jfc.setDialogTitle("已保存");
73. jfc.showSaveDialog(null);
74. File file = jfc.getSelectedFile();
75. BufferedWriter bw = null;
76. try {
77. FileWriter fw = new FileWriter(file);
78. bw = new BufferedWriter(fw);
79. String jtaStr = jta.getText();
80. bw.write(jtaStr);
81. } catch (Exception e2) {
82. // TODO: handle exception
83. e2.printStackTrace();
84. } finally {
85. try {
86. bw.close();
87. } catch (IOException e1) {
88. // TODO Auto-generated catch block
89. e1.printStackTrace();
90. }
91. }
92. } else if (str.equals("退出")) {
93. System.exit(0);
94. }
95. }
96. }
97. public class MyMenu {
98. public static void main(String[] str) {
99. MyNotePad notePad = new MyNotePad();
100. }
102. }

上述代码中，第7-13行代码，声明了文本域、菜单栏、菜单、菜单项、打开、保存、退出以及选择文件等对象。MyNotePad()方法中，对文本域、菜单栏、菜单、菜单项、打开、保存、退出等对象进行实例化，并将打开、保存、退出放入菜单容器，将菜单容器放入菜单栏容器，将菜单栏放入窗口容器。并设置了窗口容器的大小。actionPerformed()方法中，编写了打开文件、保存、以及退出的逻辑。最后，在main()方法中对容器对象进行实例化，用于启动Swing案例程序。