

## Labnodpad

```
FileManager.java notepad.java
1 import java.io.BufferedReader;
2 import java.io.File;
3 import java.io.FileInputStream;
4 import java.io.FileOutputStream;
5 import java.io.IOException;
6 import java.io.InputStreamReader;
7 import java.io.OutputStreamWriter;
8 import java.io.PrintWriter;
9
10 public class FileManager {
11     public static String fileReader(File file) {
12         try(FileInputStream fin = new FileInputStream(file);
13             InputStreamReader in = new InputStreamReader(fin);
14             BufferedReader bin = new BufferedReader(in))
15         {
16             String text="", keep;
17             while((keep=bin.readLine()) != null) {
18                 text += keep;
19                 text+="\n";
20             }
21             return text;
22         }
23         catch (IOException e) {
24             System.out.println(e);
25             return "";
26         }
27         catch (Exception e) {
28             System.out.println(e);
29             return "";
30         }
31     }
32     public static void fileWriter(File file, String text) {
33         FileOutputStream fout;
34         OutputStreamWriter oout;
35         PrintWriter p;
36         try{
37             fout = new FileOutputStream(file);
38             oout = new OutputStreamWriter(fout);
39             p = new PrintWriter(oout);
40             for(int i=0;i<text.length();i++) {
41                 if(text.charAt(i) == '\n')
42                     p.println();
43                 else
44                     p.print(text.charAt(i));
45             }
46             p.close();
47             oout.close();
48             fout.close();
49         }
50         catch (Exception e) {
51             System.out.println(e);
52         }
53     }
54 }
```

```
notepad.java
1 import java.awt.event.ActionEvent;
2 import java.awt.event.ActionListener;
3 import java.io.File;
4 import javax.swing.JFileChooser;
5 import javax.swing.JFrame;
6 import javax.swing.JMenu;
7 import javax.swing.JMenuBar;
8 import javax.swing.JMenuItem;
9 import javax.swing.JScrollPane;
10 import javax.swing.JTextArea;
11
12 public class notepad implements ActionListener{
13     private JFrame fr;
14     private JMenuBar menubar;
15     private JMenuItem s1, s2, s3, s4, s5;
16     private JMenu menu1, menu2;
17     private JTextArea ta;
18     private JFileChooser fileChooser;
19     private Instruction instruction;
20     private File file;
21
22     public void init() {
23         instruction = new Instruction();
24         fr = new JFrame("Notepad");
25         fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26         menubar = new JMenuBar();
27
28         fileChooser = new JFileChooser(new File("C:\\Users\\itkmitl\\eclipse-workspace\\Notepad"));
29
30         menu1 = new JMenu(instruction.getMenubarInstruct(1)); // File
31         menu2 = new JMenu(instruction.getMenubarInstruct(2)); // Edit
32
33         s1 = new JMenuItem(instruction.getFileInstruction(1)); // new
34         s2 = new JMenuItem(instruction.getFileInstruction(2)); // open
35         s3 = new JMenuItem(instruction.getFileInstruction(3)); // save
36         s4 = new JMenuItem(instruction.getFileInstruction(4)); // save as
37         s5 = new JMenuItem(instruction.getFileInstruction(5)); // exit
38         menu1.add(s1);
39         menu1.add(s2);
40         menu1.add(s3);
41         menu1.add(s4);
42         menu1.addSeparator();
43         menu1.add(s5);
44     }
45 }
```

notepad.java

```

44     s2.addActionListener(this);
45     s3.addActionListener(this);
46     s4.addActionListener(this);
47     s5.addActionListener(this);
48
49
50     fr.setJMenuBar(menubar);
51     menubar.add(menu1);
52     menubar.add(menu2);
53
54     ta = new JTextArea();
55     JScrollPane jScrollPane = new JScrollPane(ta);
56     fr.setVisible(true);
57     fr.setSize(300, 200);
58     fr.add(jScrollPane);
59
60 }
61 public static void main(String[] args) {
62     notepad obj = new notepad();
63     obj.init();
64 }
65 @Override
66 public void actionPerformed(ActionEvent arg0) {
67     String cmd = arg0.getActionCommand();
68     //exit
69     if(cmd.equals(instruction.getFileInstruction(5))) {
70         System.exit(0);
71     }
72     //open
73     else if(cmd.equals(instruction.getFileInstruction(2))) {
74         int returnVal = fileChooser.showOpenDialog(null);
75         if (returnVal == JFileChooser.APPROVE_OPTION) {
76             file = fileChooser.getSelectedFile();
77             ta.setText(FileManager.fileReader(file));
78         }
79     }
80     //save
81     else if(cmd.equals(instruction.getFileInstruction(3))) {
82         FileManager.fileWriter(file, ta.getText());
83     }
84 }
85 //save as
86 else if(cmd.equals(instruction.getFileInstruction(4))) {
87     else if(cmd.equals(instruction.getFileInstruction(4))) {
88         int returnVal = fileChooser.showSaveDialog(null);
89         if (returnVal == JFileChooser.APPROVE_OPTION) {
90             file = fileChooser.getSelectedFile();
91             FileManager.fileWriter(file, ta.getText());
92         }
93     }
94 }
95 }

```

FileManager.java

Instruction.java

notepad.java

```

1 public class Instruction {
2     public String getFileInstruction(int i) {
3         String instruct = "";
4         switch(i) {
5             case 1: instruct = "New";break;
6             case 2: instruct = "Open";break;
7             case 3: instruct = "Save";break;
8             case 4: instruct = "Save as";break;
9             case 5: instruct = "Exit";break;
10        }
11        return (instruct);
12    }
13    public String getMenuBarInstruct(int i) {
14        String instruct = "";
15        switch(i) {
16            case 1: instruct = "File";break;
17            case 2: instruct = "Edit";break;
18        }
19        return (instruct);
20    }
21 }

```