Text Editor

Documentation

Name : Java Text Editor

Authors : Subhanu Majumder, Shinjinee Mukherjee, Sri Krishna Agrawal

Description : This is a simple java text editor. The editor does basic features like text editing (cut, copy, paste) and text wrapping along with the options to save and open files of any type, including txt, .java, .py and .docx. The editor can also print out a certain document.

Functions :

- Constructor : Initializes all the GUI elements like JTextArea and JMenu etc.
- void actionPerformed : Impliments all functions related to the anonymous ActionListener's like saving a file, printing, text wrapping etc.
- public static void main : The main method of the class, it creates a new object of the class.

API used : None

Github Link : https://github.com/subhanuTHSmajumder/TextEditor

Source Code

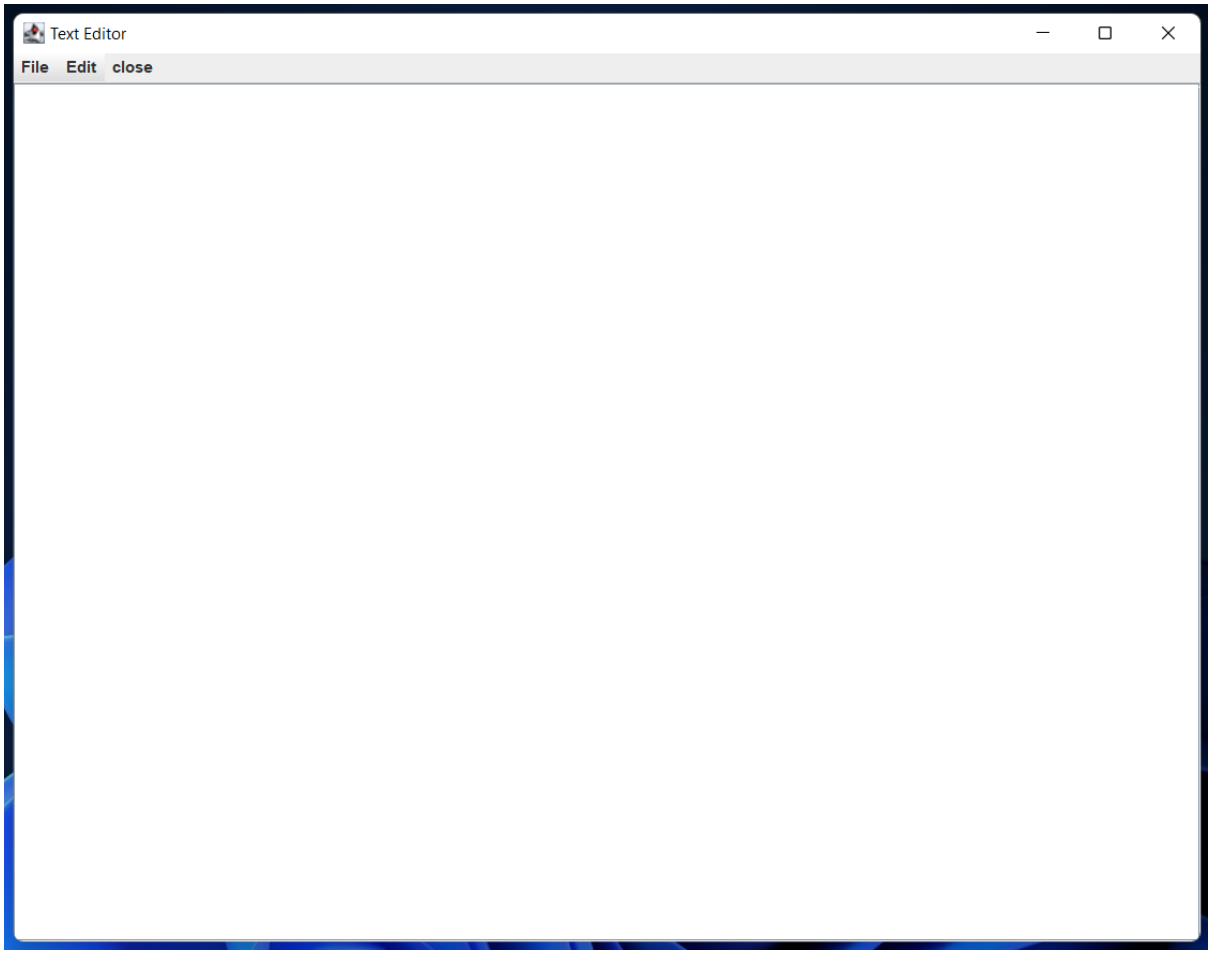
```
//import all the required java packages

import javax.swing.*;
import java.awt.*;
import java.io.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

//A new class that has the functionalities of JFrame and impliments ActionListners by default, letting us use them anonymously
public class Main extends JFrame implements ActionListener {

    private final JTextArea editor; // Declare text area
    private final JFrame frame; // Declare new frame
    private int c = 0; // Declare integer variable and set it to 0

    // Declaring a constructor to innitialize all the data members
    public Main() {

        // Innitialize different gui data members
        frame = new JFrame("Text Editor");
        editor = new JTextArea();
        JScrollPane scrollPane = new JScrollPane(editor);
        JMenuBar menuBar = new JMenuBar();
        JMenu fileMenu = new JMenu("File");
```

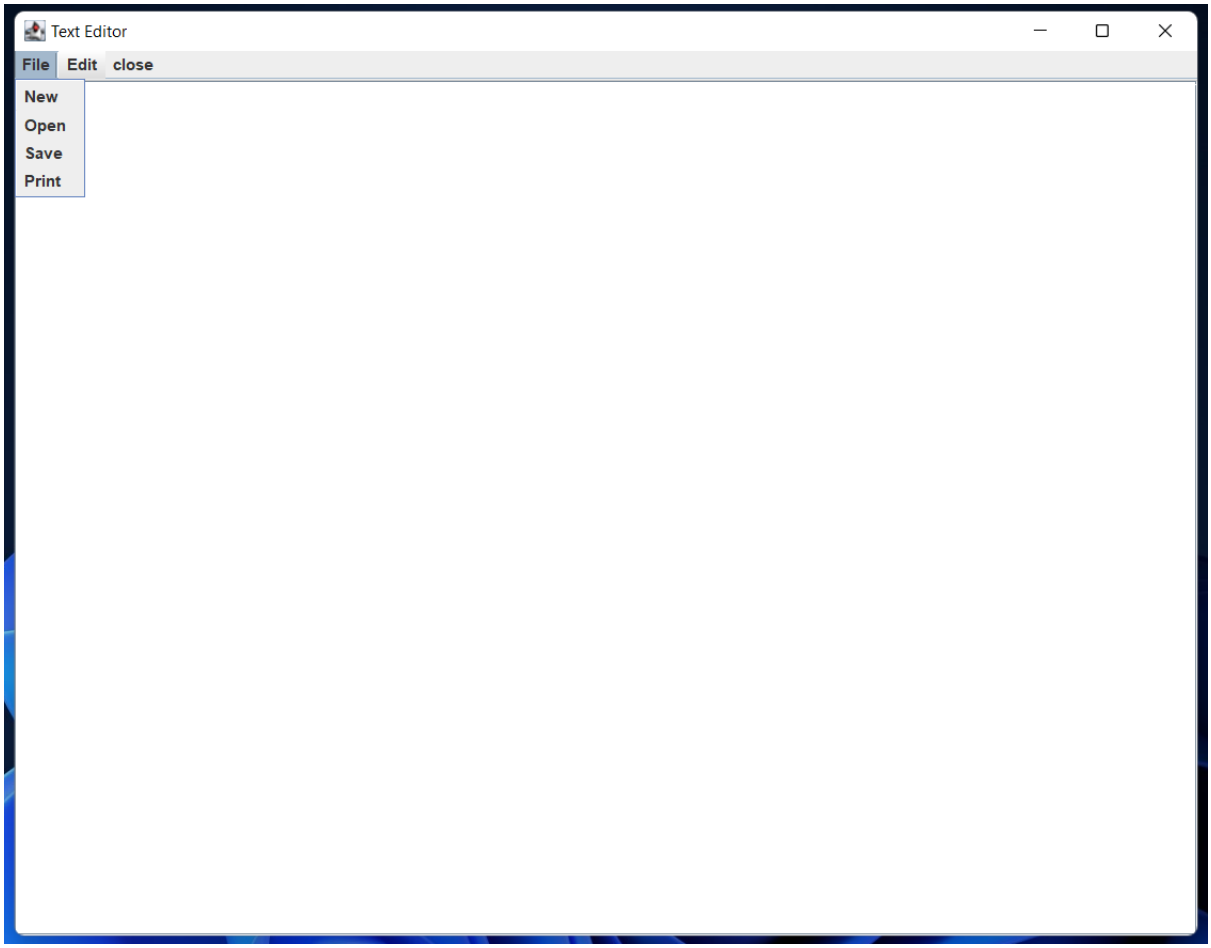
```
        // Declare new sub menu item
        JMenuItem newFile = new JMenuItem("New");
        JMenuItem openFile = new JMenuItem("Open");
        JMenuItem saveFile = new JMenuItem("Save");
        JMenuItem printFile = new JMenuItem("Print");
        JMenuItem cutText = new JMenuItem("cut");
        JMenuItem copyText = new JMenuItem("copy");
        JMenuItem pasteText = new JMenuItem("paste");
        // JMenuItem replaceText = new JMenuItem("replace");
        JMenuItem wrapText = new JMenuItem("wrap text");
        JMenuItem closeFile = new JMenuItem("close");
        JMenu editFile = new JMenu("Edit");

        editor.setMargin(new Insets(10, 10, 10, 10)); // Set margin for text area

        // Add anonymous action listeners
        newFile.addActionListener(this);
        openFile.addActionListener(this);
        saveFile.addActionListener(this);
        printFile.addActionListener(this);
        cutText.addActionListener(this);
        copyText.addActionListener(this);
        pasteText.addActionListener(this);
        // replaceText.addActionListener(this);
        wrapText.addActionListener(this);
        closeFile.addActionListener(this);

        // Add the sub menu items to the parent menu item
        fileMenu.add(newFile);
        fileMenu.add(openFile);
        fileMenu.add(saveFile);
        fileMenu.add(printFile);
        editFile.add(cutText);
        editFile.add(copyText);
        editFile.add(pasteText);
        //JMenuItem replaceText = new JMenuItem("replace");
        editFile.add(wrapText);

        // Add menu item to the menu bar
        // menuBeditFile.add(replaceText);ar.add(menuItem);
        menuBar.add(fileMenu);
        menuBar.add(editFile);
        menuBar.add(closeFile);

        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);// Set the default close operation
        frame.setSize(900, 700); // Set the innitial size of the window

        // Add all the gui items to the frame
        frame.setJMenuBar(menuBar);
        frame.add(scrollPane);

        frame.setVisible(true); // Make the frame visible
    }

    @Override
    // overriding a class function to assign tasks to the menu items via action
    // listeners
    public void actionPerformed(ActionEvent e) {

        String s = e.getActionCommand(); // Gets the value of the menu bar items through anonymous click listener

        // Switch case statement to assign seperate tasks to the different menubar items
        switch (s) {

            // Cut action
            case "cut":
                editor.cut();
                break;

            // Copy action
            case "copy":
                editor.copy();
                break;

            // Paste option
            case "paste":
                editor.paste();
                break;

            case "replace":

                break;
```

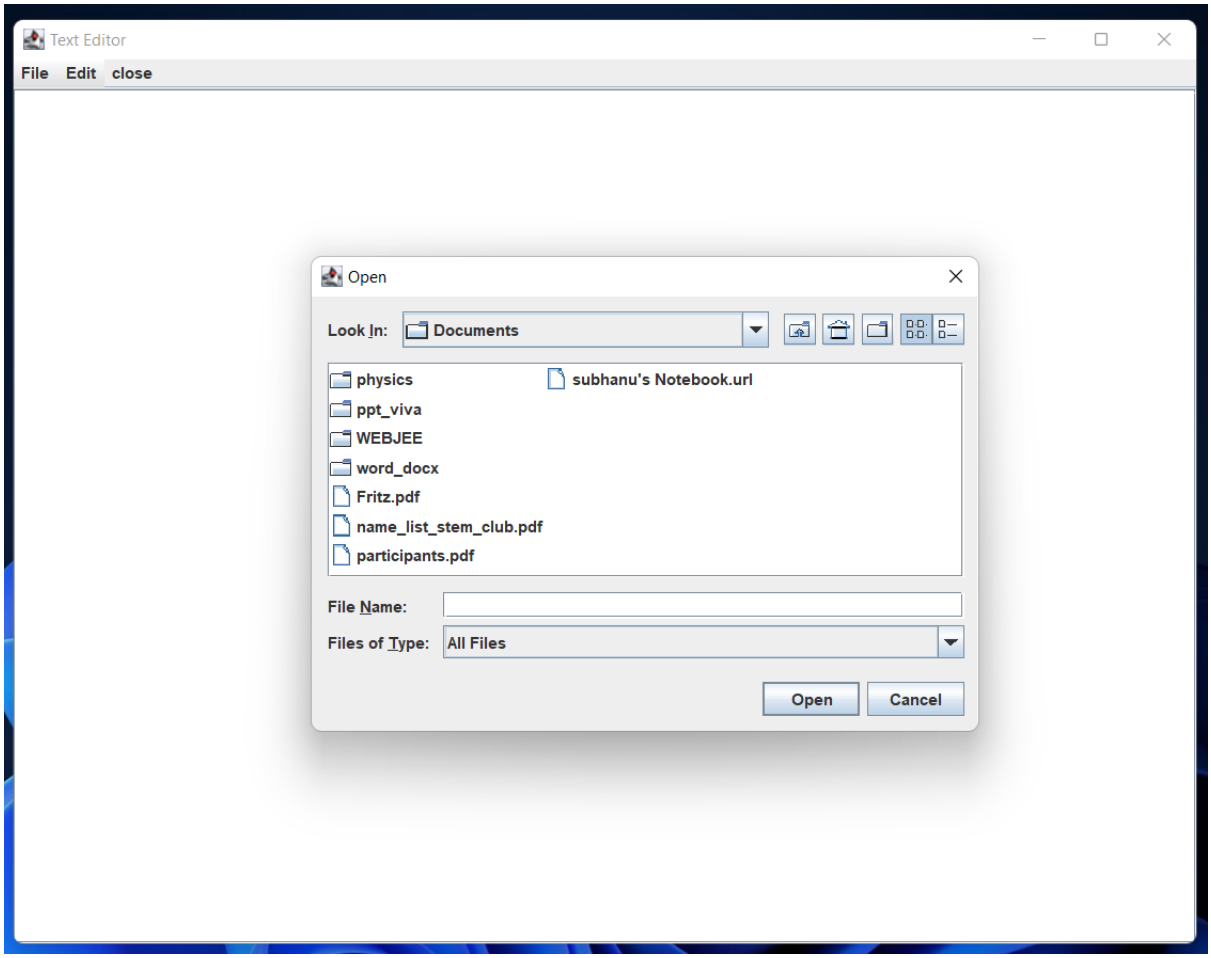
```
            // Wrap text functionality
            case "wrap text":
                c++;
                if (c % 2 != 0) {
                    editor.setLineWrap(true);
                    editor.setWrapStyleWord(true);
                } else {
                    editor.setLineWrap(false);
                    editor.setWrapStyleWord(false);
                }
                break;

            // Save option
            case "Save": {
                // Declare a new chooser, to choose the location of saving the file
                JFileChooser j = new JFileChooser("/home");
                int r = j.showSaveDialog(null);

                // Check wether the user has selected a location to save the file
                if (r == JFileChooser.APPROVE_OPTION) {
                    // Extract the location set by the user from the chooser object
                    File file = new File(j.getSelectedFile().getAbsolutePath());
                    try {

                        // Writing all the content to the assigned file
                        FileWriter fr = new FileWriter(file, false);
                        BufferedWriter br = new BufferedWriter(fr);
                        // Extract the text from the text area
                        br.write(editor.getText());
                        br.flush();
                        br.close();
                    } catch (Exception evt) {
                        // Generate a new default window to show an error message
                        JOptionPane.showMessageDialog(frame, evt.getMessage());
                    }
                } else
                    // Generate a new default window to show a cancellation message
                    JOptionPane.showMessageDialog(frame, "the user cancelled the operation");
                break;

            }

            // Print option
            case "Print":
                try {
                    // Try to print the file using the inbuilt method print()
                    editor.print();
                } catch (Exception evt) {
                    // Generate a new default window to show an error message
                    JOptionPane.showMessageDialog(frame, evt.getMessage());
                }

                break;

            // Open file option
            case "Open": {
                // Declare a new chooser, to choose the location of saving the file
                JFileChooser j = new JFileChooser("/home:");
                int r = j.showOpenDialog(null);
                // Check wether the user has selected a valid file
                if (r == JFileChooser.APPROVE_OPTION) {
                    // Extract the location used by the user from the chooser object
                    File fi = new File(j.getSelectedFile().getAbsolutePath());
                    try {

                        // Read from file and set the text of the text area to be the contents of the
                        // file
                        String s1, sl;
                        FileReader fr = new FileReader(fi);
                        BufferedReader br = new BufferedReader(fr);
                        sl = br.readLine();
                        while ((s1 = br.readLine()) != null) {
                            sl = sl.concat("\n").concat(s1);
                        }
                        editor.setText(sl);
                        br.close();
                    } catch (Exception evt) {
                        // Generate a new default window to show an error message
                        JOptionPane.showMessageDialog(frame, evt.getMessage());
                    }
                } else
```

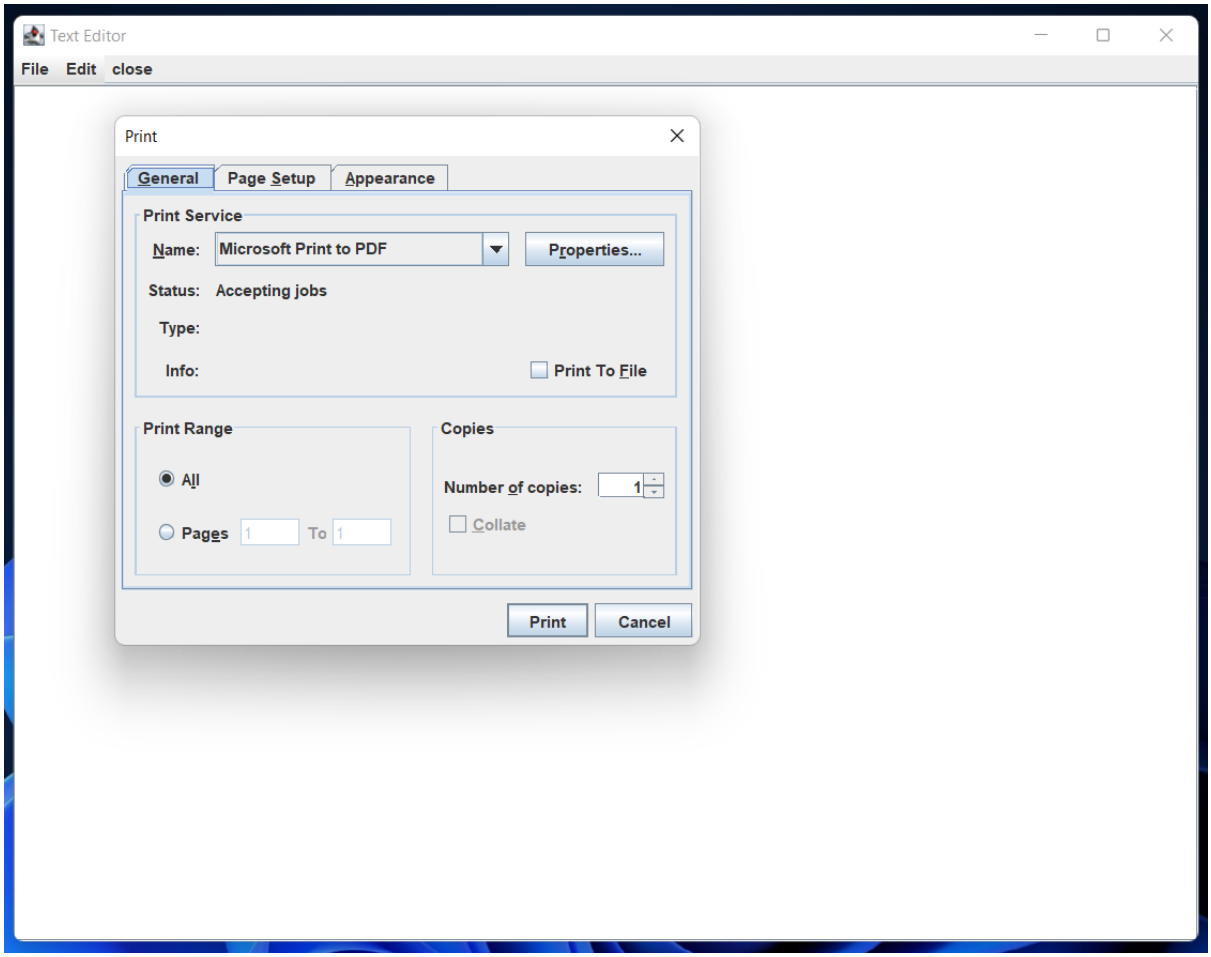
```
                    // Generate a new default window to show a cancellation message
                    JOptionPane.showMessageDialog(frame, "the user cancelled the operation");
                break;

            }

            // Create new file option
            case "New":
                // Make a new empty file
                editor.setText("");

                break;

            // Close app function
            case "close":
                // Close the text editor and exit from the program by setting the frame
                // visiblity to false
                frame.setVisible(false);
                System.exit(1);
                break;
        }
    }

    // Main method
    public static void main(String[] args) {
        // New instance of the class Main
        new Main();
    }
}
```

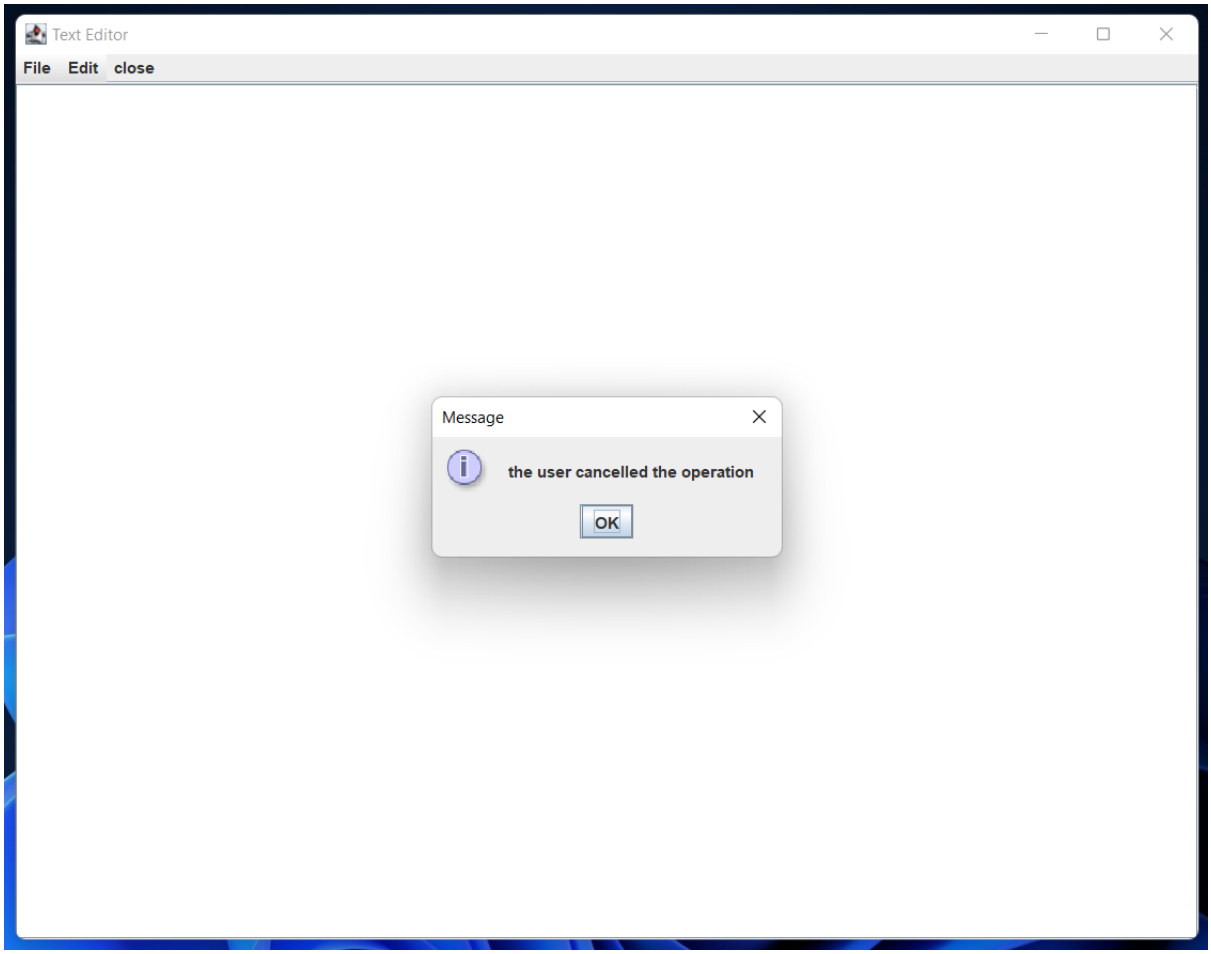
Output Screenshots

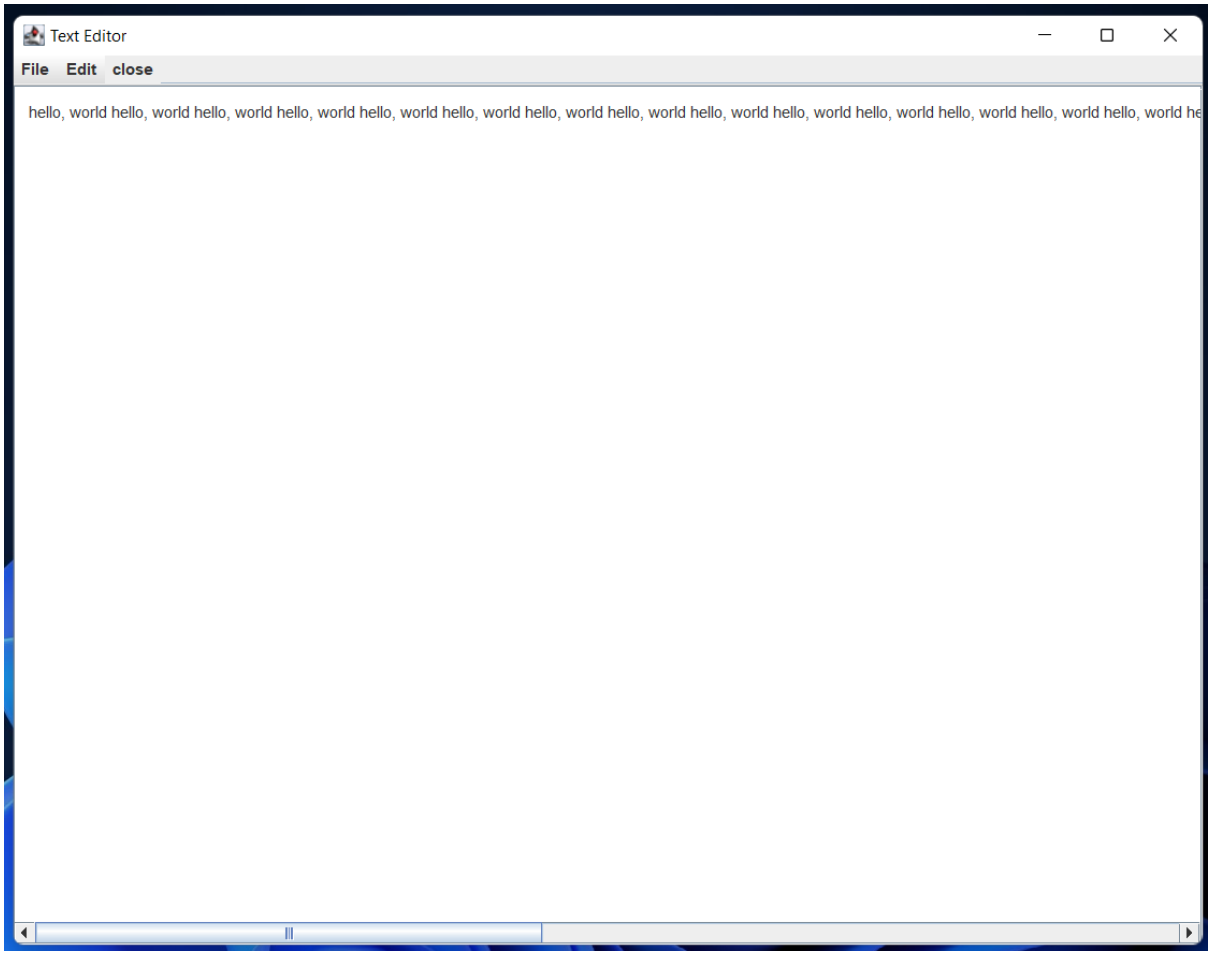


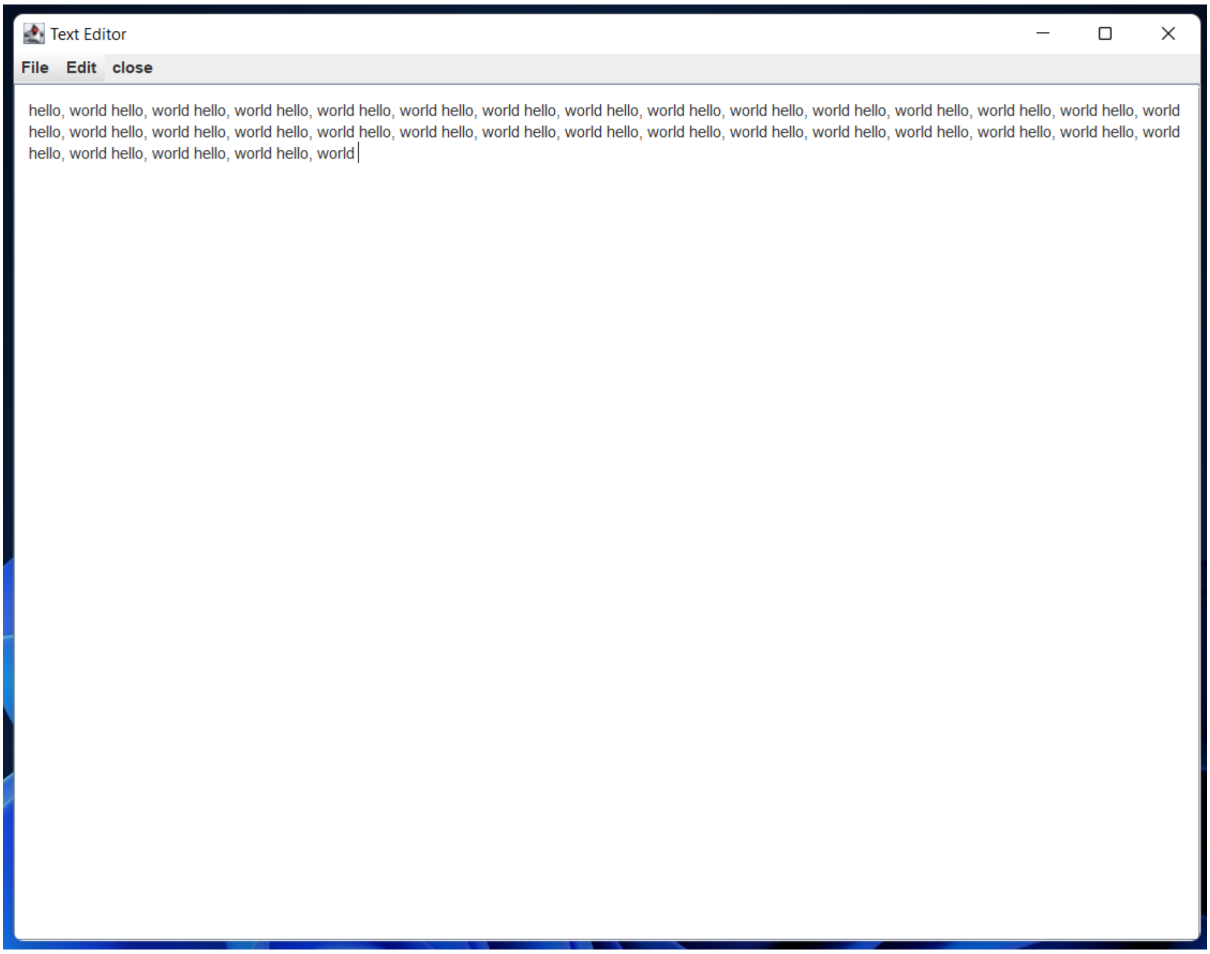

Text Editor
File
Edit
close
New
Open
Save
Print

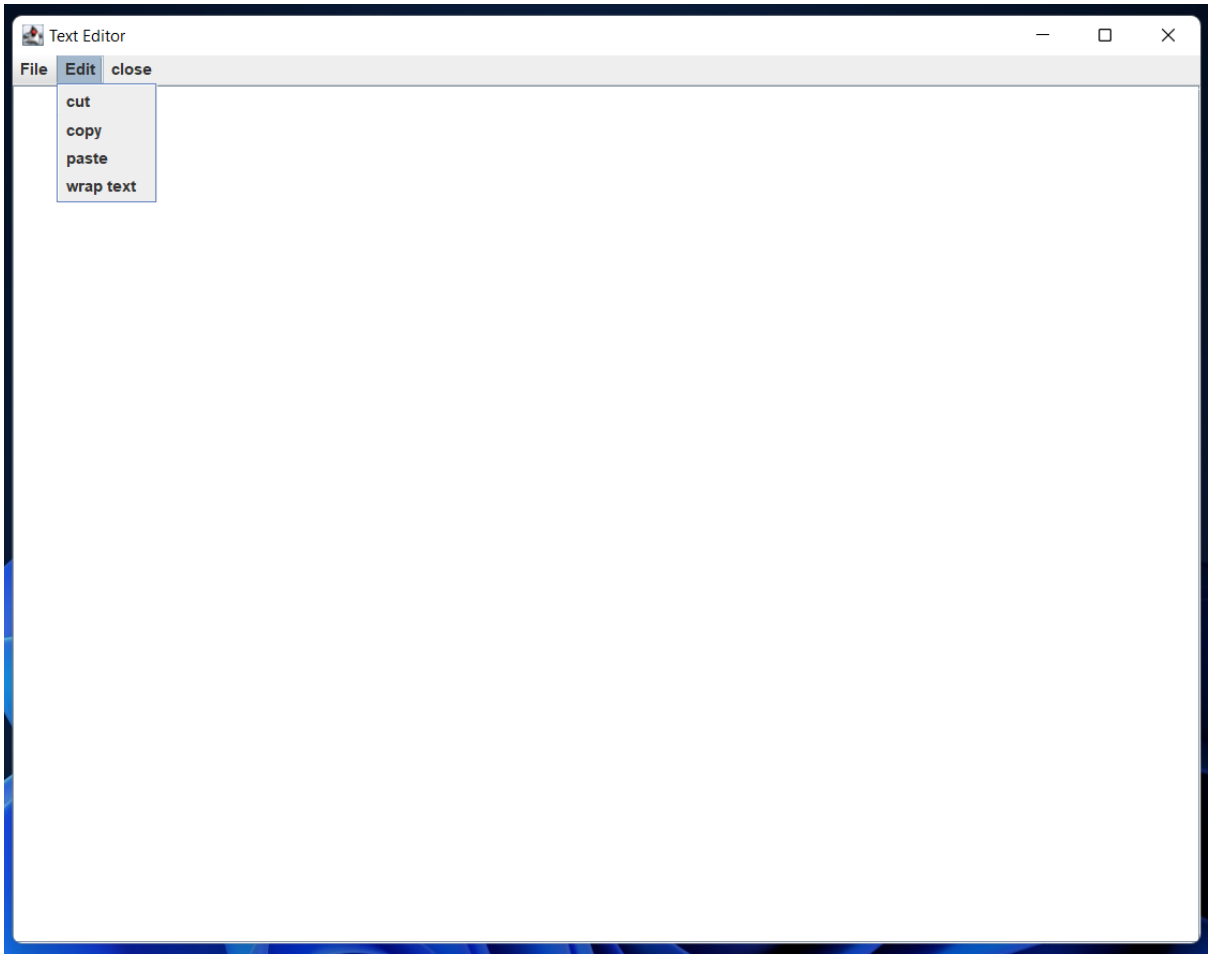

Text Editor
File
Edit
close
Open
Look In:
Documents
physics
ppt_viva
WEBJEE
word_docx
Fritz.pdf
name_list_stem_club.pdf
participants.pdf
subhanu's Notebook.url
File Name:
Files of Type:
All Files
Open
Cancel


Text Editor
File
Edit
close
Print
General
Page Setup
Appearance
Print Service
Name:
Microsoft Print to PDF
Properties...
Status:
Accepting jobs
Type:
Info:
Print To File
Print Range
All
Pages
1
To
1
Copies
Number of copies:
1
Collate
Print
Cancel


Text Editor
File
Edit
close
Message
the user cancelled the operation
OK


Text Editor
File
Edit
close
hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world he


Text Editor
File
Edit
close
hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world
hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world hello, world
hello, world hello, world hello, world hello, world


Text Editor
File
Edit
close
cut
copy
paste
wrap text