Text Editor (SEAS Notepad)

Team Member

Vaishwi Patel : 1741079

Arpit Patel : 1741074

Mentor

Hiral Vegda

- ## Brief Description:

In daily life, we have to remember many things like our daily schedules, meetings, timetable etc, and also need to write essays, reports to represents our ideas. But we are human beings so we forget many times. Due to it, we need a tool which helps to make our work easy.

We are going to design a simple text editor like notepad. For that purpose, we are using data structures like a linked list, queue, stack. And for display, GUI of java will be used for creating a text editor. A text editor is a program that allows you to open, view, and edit plain text files. For cut, copy, paste and delete command, a linked list would be implemented. We use stack data structure for undo command. We are going to use the Java programming language. We use the different functionality of Java. Like GUI for user interface and display output. File handling for doing an operation like save, open and Rename command on existing or new file.

- ## List of Data Structures used with logic design

1) Linked List:

Linked list consists of nodes where each node contains a data field and a reference (link) to the next node in the list. Linked list's operation like insertion, searching and deletation are helpful in making cut, paste and delete logic. When user enter any word in text Area that word insert at last position in linked list. And like that linked list is create.

2) Stack:

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be FILO (First In Last Out). Stack's operation like push and pop are helpful in making undo logic. When user do any operation we push operation ID and location (linked list position where operation is done) into stack and when user want to undo we just pop last operation and place it where last modified.

3) 1D Array:

When user enter any line in text area that line is in String. So when we want to insert word by word in linked list we separate out word with the use of space. And then we insert one by one words into lined list.

- **Operations to be perform on each data structure:**

1) Insertion:

When user enter any word in text Area that word insert at last position in linked list. Also when user do paste operation which words are in stack are insert at where caret is located in linked list which we are created.

2) Deletion:

When user do cut or delete operation which words are selected in text area are delete from our linked list.

3) Push:

When user do any operation that operation ID and that modification are pushed into stack.

4) Pop:

When user do Undo operation whatever last modification pushed into stack is pop and that modification located at position which we are pushed into stack with that modification.

5) Searching:

When user do find operation new frame is open where user would enter text witch he/she want to find from text area. And then whatever text user want to find that text search into linked list which we are created. If text is find into text area that text is highlighted otherwise error massage is appear.

- **List of programs**
    - **Filename:** Notepad.java
- **Source codes:**

1. Open New File:

```
openFile.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
        FileDialog fileDialog = new FileDialog(frame, "Open File", FileDialog.LOAD);

        fileDialog.setVisible(true);

        if(fileDialog.getFile() !=null) {
                filename = fileDialog.getDirectory() + fileDialog.getFile();
                frame.setTitle(fileDialog.getFile());
        }
        try{
        BufferedReader reader = new BufferedReader(new FileReader(filename));
                StringBuilder sb = new StringBuilder();
                String line = null;
                while((line = reader.readLine()) !=null) {
                        sb.append(line +"\n");
                }
                textArea.setText(sb.toString());
                reader.close();
        }catch(IOException e1) {
                System.out.println("File not found");
                }
        }
});
```

2. Save File: (Save / Save As)

```
public boolean saveFile(int cnt) {
        FileDialog fileDialog = null;
        if( cnt == 0) {
                fileDialog = new FileDialog(frame, "Save File", FileDialog.SAVE);
                fileDialog.setVisible(true);
                cnt++;                  // check for file already save or nor?

                if(fileDialog.getFile() != null) {
                        fileName = fileDialog.getDirectory() + fileDialog.getFile();
                        frame.setTitle(fileDialog.getFile());
                }
                try {
                        FileWriter fileWriter = new FileWriter(fileName);
                        saveText = textArea.getText();
                        fileWriter.write(saveText);
                        //frame.setTitle(fileDialog.getFile());
                        fileWriter.close();
                } catch (Exception e) {
                        System.out.println("File Not Found");
                }
        }
        else {
                try {
                        FileWriter fileWriter = new FileWriter(fileName);
                        saveText = textArea.getText();
                        fileWriter.write(saveText);
                        //frame.setTitle(fileDialog.getFile());
                        fileWriter.close();
                } catch (Exception e) {
                        System.out.println("File Not Found");
                }
        }
}
```

3. Cut:

```
        String SelectedWords = textArea.getSelectedText();
        String text = textArea.getText();
        LinkedList list = getTextList(text);
        int posSelStart = textArea.getSelectionStart();

        int lenOFSelectedText = SelectedWords.length();
        String finalText = "";
        int len =0;
        int pos=0;
        String hf;
        for(int i=0 ; list.get(i+1)!=null ; i++ ) {

                len = len + list.get(i).toString().length()+1;
                if(posSelStart==len) {

                        pos =i+1;
                        hf="fws";
                        break;
                }
                else if(posSelStart>len & posSelStart<(len+list.get(i+1).toString().length())){

                        pos=i+1;
                        hf = "hw";
                        break;
                }
                else {
                        pos=i+1;
                hf = "fis";
                        break;
                }

        }
                if(list.get(pos).toString().length()==lenOFSelectedText ||
(list.get(pos).toString().length()+2) == lenOFSelectedText || (list.get(pos).toString().length()+1)
== lenOFSelectedText ) {

                        list.remove(pos);
                        for(int k =0 ; k!=pos ; k++) {
                                finalText += list.get(k).toString()+" ";
                                if(k==pos-1)
```

```
                                        finalText+=list.get(k).toString();
                            }
                            if((list.get(pos).toString().length()+2) == lenOFSelectedText)
                                  finalText+="";
                            else if((list.get(pos).toString().length()+1) == lenOFSelectedText)
                                  finalText+=" ";
                            else
                                  finalText+="  ";
                            for(int k = pos+1 ; list.get(k)!=null ; k++) {
                                  finalText += list.get(k).toString()+" ";
                                  if(list.get(k+1)==null)
                                        finalText += list.get(k).toString();
                            }
                            System.out.println("final text:"+finalText);

                      }

                      for(int k=pos+1;list.get(k)!=null ; k++) {

                            if(hf == "fws") {
                                  if(list.get(k).toString().length()==lenOFSelectedText) {

                                  }
                            }
                      }
                return finalText;
```

4. Find:

```
          public void highlight(JTextComponent textComp ,String pattern) {
                removeHighlights(textComp);

                try {
                      Highlighter hilite = textComp.getHighlighter();

                      Document doc = textComp.getDocument();
                      String text = doc.getText(0, doc.getLength());
                      int pos=0;
                      int cnt=0;
                while((pos=text.toUpperCase().indexOf(pattern.toUpperCase() , pos)) >=0) {

                        System.out.print("YES");
                        hilite.addHighlight(pos, pos+pattern.length() , myHighlightPainter );
                        pos+=pattern.length();
                        cnt++;
                }
                if(cnt==0){

                        JFrame frameF = new JFrame();
                        JButton btnClose = new JButton("OK");
                        frameF.setBounds(100, 100, 450, 300);
                        frameF.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                        frameF.getContentPane().setLayout(null);
                        frameF.setVisible(true);

                        //frameF.add(btnClose);


                        btnClose.setFont(new Font("Tahoma", Font.BOLD, 11));
                        btnClose.addActionListener(new ActionListener() {
                                public void actionPerformed(ActionEvent arg0) {
                                        frameF.setVisible(false);
                                }
                        });
                        btnClose.setBounds(181, 143, 89, 23);
                        frameF.getContentPane().add(btnClose);

                        JTextPane txtpnTextNotFound = new JTextPane();
                        txtpnTextNotFound.setForeground(Color.RED);
                        txtpnTextNotFound.setBackground(Color.WHITE);
                        txtpnTextNotFound.setFont(new Font("Tahoma", Font.BOLD, 14));
                        txtpnTextNotFound.setText("! Text Not Found");
                        txtpnTextNotFound.setBounds(165, 76, 124, 20);
                        frameF.getContentPane().add(txtpnTextNotFound);
                }

        }catch(Exception e) {

        }
}
```

5. Font:

```
if(fontDialog==null)
        fontDialog=new FontChooser (textArea.getFont());
if(fontDialog.showDialog ( frame,"Choose a font"))
        textArea.setFont (fontDialog.createFont());
```

6. Foreground:

```
void showForegroundColorDialog()
{
        if(fcolorChooser==null)
           fcolorChooser=new JColorChooser();
        if(foregroundDialog==null)
           foregroundDialog=JColorChooser.createDialog
              (this.frame,  "Set Text color...",  false,  fcolorChooser,
               new ActionListener()
              {public void actionPerformed(ActionEvent ev){
                 textArea.setForeground(fcolorChooser.getColor());}},
              null);

        foregroundDialog.setVisible(true);
}
```

7. Background:

```
void showBackgroundColorDialog()
{
        if(bcolorChooser==null)
           bcolorChooser=new JColorChooser();
        if(backgroundDialog==null)
           backgroundDialog=JColorChooser.createDialog
              (this.frame,  "Set Pad color...",  false, bcolorChooser,
              new ActionListener()
              {public void actionPerformed(ActionEvent ev){
                 textArea.setBackground(bcolorChooser.getColor());}},
              null);
        backgroundDialog.setVisible(true);
}
```

- **References:**

1. "Linked Lists implementation" https://www.cs.cmu.edu/~adamchik/15-121/lectures/Linked%20Lists/linked%20lists.html.
2. "Overview (Java Platform SE 7) - Oracle Docs." https://docs.oracle.com/javase/7/docs/api/.
3. "java - How to split a String by space - Stack Overflow." 26 Oct. 2011, https://stackoverflow.com/questions/7899525/how-to-split-a-string-by-space/30220543.
4. "GUI Programming - Java Programming Tutorial." http://www.ntu.edu.sg/home/ehchua/programming/java/j4a_gui.html.