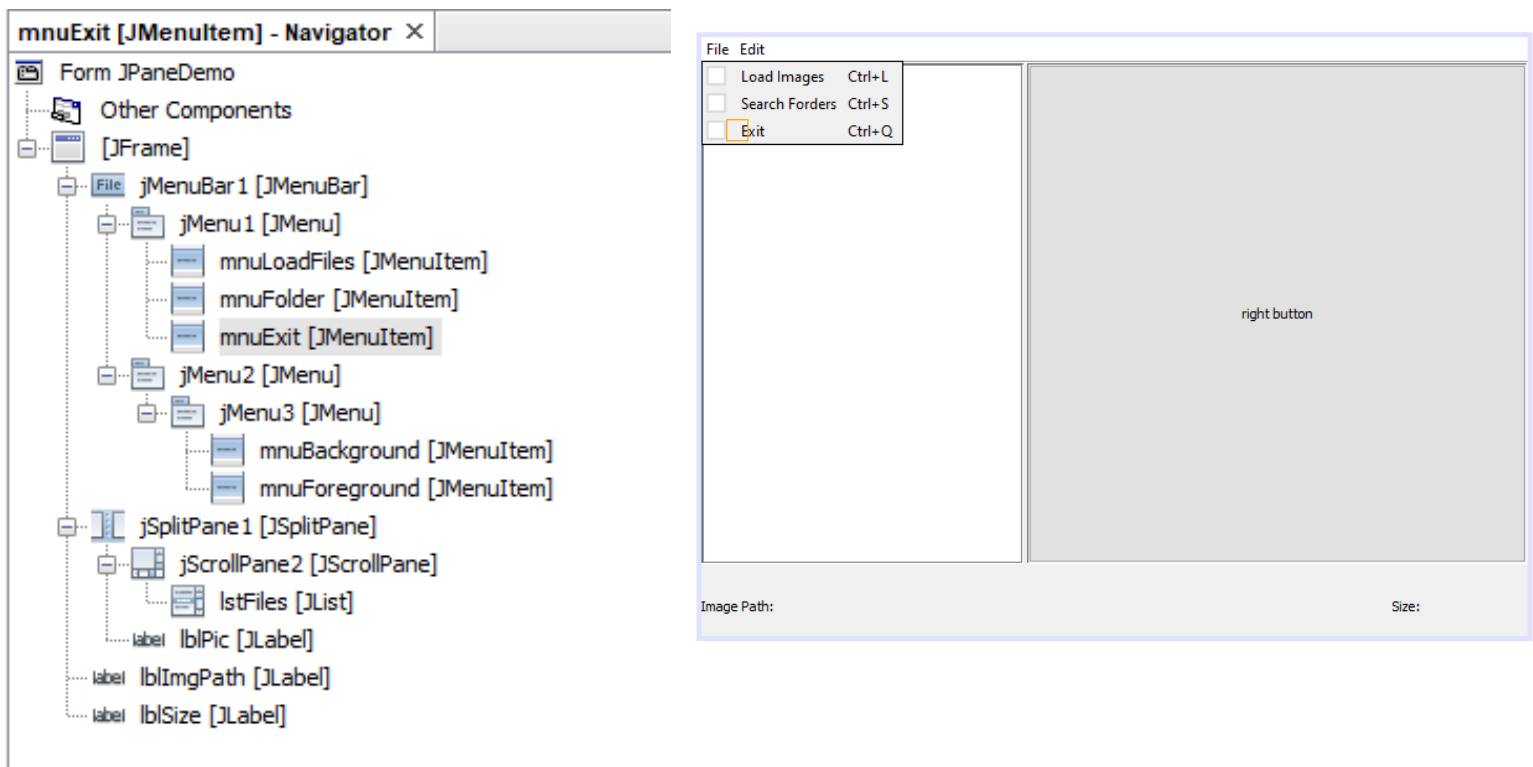


## Class Task:

Implement a photo viewer app that loads image files from a folder into a list. When selected items are clicked on list, the image shows up in the label/panel on the right side of the split pane.

- JSplitPane
- JScrollPane
- JOptionPane
- JColorChooser
- JFileChooser
- ImageIcon class
- ImageIO
- BufferedImage class



## Code snippets:

- LoadIMGFile ()
- lstFilesMouseClicked ()

*When user clicks on the file names in the List, corresponding images get loaded in the right pane.*

```

private void loadImageFile(String imagePath) throws IOException{
    BufferedImage img = null;
    File file = new File(imagePath);
    img = ImageIO.read(file);

    Image dimg = img.getScaledInstance(lblPic.getWidth(), lblPic.getHeight(),
        Image.SCALE_SMOOTH);

    ImageIcon pic = new ImageIcon (dimg);
    lblPic.setIcon(pic);

    double size = (double) (file.length()/1024.0);
    DecimalFormat actualSize = new DecimalFormat("##");

    //System.out.println(file.length()/1024);
    lblImgPath.setText(" Path: " + imagePath);
    lblSize.setText("Size: " + actualSize.format(size) + " KB");
}

```

```

private void lstFilesMouseClicked(java.awt.event.MouseEvent evt) {
    //System.out.println(lstFiles.getModel().getSize());
    if (lstFiles.getModel().getSize()>0){
        String fileName = lstFiles.getSelectedValue();
        String filePath = currentPath + "/" + fileName;
        // System.out.println(filePath);
        try {
            loadImageFile(filePath);
        } catch (IOException ex) {
            Logger.getLogger(JPaneDemo.class.getName()).log(Level.SEVERE, null, ex);
        }
    } else{
        JOptionPane.showMessageDialog(this, "No image exists", "Error!", JOptionPane.ERROR_MESSAGE);
    }
}

```

### Choosing a folder to load the image file names into List:

- **loadIMGFile()**

*This method filters the .png and .jpg files in side a directory and loads their names into the List*

- **mnuFolderActionPerformed**

*This menu-click method opens a JFileChooser and sets it up in a way that only Directories can be accessed.*

```

DefaultListModel <String> listModel;
String currentPath ;

public void loadIMGFile(String path){
    File file = new File(path);
    String[] list = file.list(new FilenameFilter() {
        @Override
        public boolean accept(File dir, String name) {
            if(name.toLowerCase().endsWith(".png") || (name.toLowerCase().endsWith(".jpg"))){
                return true;
            } else {
                return false;
            }
        }
    });
    listModel = new DefaultListModel<>();

    for(String f:list){
        System.out.println(f);
        listModel.addElement(f);
    }

    lstFiles.setModel(listModel);
    //lstFiles.setSelectedIndex(0);
}

```

```

private void mnuFolderActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser fc = new JFileChooser();
    fc.setCurrentDirectory(new java.io.File(".")); // start at application current directory
    fc.setName("Choose a Directory");
    fc.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    int returnVal = fc.showOpenDialog(this);

    if(returnVal == JFileChooser.APPROVE_OPTION) {
        File selectedFolder = fc.getSelectedFile();
        currentPath = selectedFolder.toPath().toString();
        loadIMGFile(currentPath);
    }
}

```