## Government Polytechnic ,Washim

A

MICRO PROJECT REPORT ON

**“FONT WINDOW”**

****

**Submitted By group of**

**1. Pavan P. Faltankar ( 1700310215)**

**2. Sacchidanand B.Linge ( 1800310215)**

**3. Vaibhav S.Ingle ( 1800310207)**

**4. Pavan S.Ingole ( 1700310216)**

***(Final Year Diploma in Information Technology)***

**Principal**

**Dr. V.R.Mankar**

***( Government Polytechnic, Washim)***

**DEPARTMENT OF OF INFORMATION TECHNOLOGY**

**2019- 2020**

**Government Polytechnic,Washim CERTIFICATE**

**This is to certify that**

**1. Pavan P. Faltankar**

**2. Sacchidanand B.Linge**

**3. Vaibhav S.Ingle**

**4. Pavan S.Ingole**

**Final year Students of Information Technology have submitted a project report on**

**“FONT WINDOW”**

**During the Fifth semester of the academic year 2019-2020 in a satisfactory manner in the partial fulfillment for the requirement for the Diploma in**

**“Information Technology” awarded by**

**Maharashtra State Board of Technical Education, Mumbai.**

**Mr. M. S. Hule**  **Mrs M. R. Lolure *( Lecturer in Info. Tech. Department)*** *(I/C HOD ,Information Technology Department)*

**Dr. V.R.Mankar**

**Principal**

***( Government Polytechnic, Washim)***

**DEPARTMENT OF INFORMATION TECHNOLOGY**

****

CONTENTS

|  |  |  |
| --- | --- | --- |
| SR.NO. | INDEX | PAGE NO. |
| 1 | introduction | 4 |
| 2 | used packages | 4 |
| 3 | java jtextfield | 5 |
| 4 | java jlabel | 7 |
| 5 | java jlist | 9 |
| 6 | used layout | 11 |
| 7 | output | 12 |

Introduction

In this project we used following some swing components that are given as follows:

1. JTextfield
2. JLabel
3. JList

Packages used in this project:

* Import java.awt.\*;
* Import javax.swing.\*;
* Import java.applet.\*;
* Import java.awt.event.\*;
* Import javax.swing.event.ListSelectionListener.\*;
* Import javax.swing.event.ListSelectionEvent.\*;

This project is designed to apply various background colors, font colors, foreground colors to target string.

We used gridlayout() for setting position of all swing components in applet used in program.

We used Form12 as a classname which extends from JApplet.

**Java JTextField**

The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.

**JTextField class declaration**

Let's see the declaration for javax.swing.JTextField class.

1. **public** **class** JTextField **extends** JTextComponent **implements** SwingConstants

**Commonly used Constructors:**

|  |  |
| --- | --- |
| **Constructor** | **Description** |
| JTextField() | Creates a new TextField |
| JTextField(String text) | Creates a new TextField initialized with the specified text. |
| JTextField(String text, int columns) | Creates a new TextField initialized with the specified text and columns. |
| JTextField(int columns) | Creates a new empty TextField with the specified number of columns. |

**Commonly used Methods:**

|  |  |
| --- | --- |
| **Methods** | **Description** |
| void addActionListener(ActionListener l) | It is used to add the specified action listener to receive action events from this textfield. |
| Action getAction() | It returns the currently set Action for this ActionEvent source, or null if no Action is set. |
| void setFont(Font f) | It is used to set the current font. |
| void removeActionListener(ActionListener l) | It is used to remove the specified action listener so that it no longer receives action events from this textfield. |

* **Java JLabel**

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

JLabel class declaration

Let's see the declaration for javax.swing.JLabel class.

1. **public** **class** JLabel **extends** JComponent **implements** SwingConstants, Accessible

* **Commonly used Constructors:**

|  |  |
| --- | --- |
| **Constructor** | **Description** |
| JLabel() | Creates a JLabel instance with no image and with an empty string for the title. |
| JLabel(String s) | Creates a JLabel instance with the specified text. |
| JLabel(Icon i) | Creates a JLabel instance with the specified image. |
| JLabel(String s, Icon i, int horizontalAlignment) | Creates a JLabel instance with the specified text, image, and horizontal alignment. |

* **Commonly used Methods:**

|  |  |
| --- | --- |
| **Methods** | **Description** |
| String getText() | t returns the text string that a label displays. |
| void setText(String text) | It defines the single line of text this component will display. |
| void setHorizontalAlignment(int alignment) | It sets the alignment of the label's contents along the X axis. |
| Icon getIcon() | It returns the graphic image that the label displays. |
| int getHorizontalAlignment() | It returns the alignment of the label's contents along the X axis. |

# Java JList

The object of JList class represents a list of text items. The list of text items can be set up so that the user can choose either one item or multiple items. It inherits JComponent class.

## JList class declaration

Let's see the declaration for javax.swing.JList class.

1. **public** **class** JList **extends** JComponent **implements** Scrollable, Accessible

### Commonly used Constructors:

|  |  |
| --- | --- |
| **Constructor** | **Description** |
| JList() | Creates a JList with an empty, read-only, model. |
| JList(ary[] listData) | Creates a JList that displays the elements in the specified array. |
| JList(ListModel<ary> dataModel) | Creates a JList that displays elements from the specified, non-null, model. |

### Commonly used Methods:

|  |  |
| --- | --- |
| **Methods** | **Description** |
| Void addListSelectionListener(ListSelectionListener listener) | It is used to add a listener to the list, to be notified each time a change to the selection occurs. |
| int getSelectedIndex() | It is used to return the smallest selected cell index. |
| ListModel getModel() | It is used to return the data model that holds a list of items displayed by the JList component. |
| void setListData(Object[] listData) | It is used to create a read-only ListModel from an array of objects. |

# Java GridLayout

The GridLayout is used to arrange the components in

rectangular grid. One component is displayed in each

rectangle.

### Constructors of GridLayout class

1. **GridLayout():** creates a grid layout with one column per component in a row.
2. **GridLayout(int rows, int columns):** creates a grid layout

with the given rows and columns but no gaps between the components.

1. **GridLayout(int rows, int columns, int hgap, int vgap):**

creates a grid layout with the given rows and columns

alongwith given horizontal and vertical gaps.

* Output:

