Paint Application

**Report On**

To Create A Paint Application By Inheriting

JFrame Class. 2(a)

Submitted to:- Prepared By:-

**Mr. Mohan Ghimire Prakash Chandra Awal**

**Roll no:- 21**

**S.E 3rd Sem(morning)**

Introduction

My Project is:

To create a Paint Application by Inheriting JFrame class where user can select the color.

***JColorChooser*** class should be used for creating the application.

Features of the Project

The feature of the Project is that user can choose any color

they want from the ***JColorChooser*** panel and can draw a line. In simple language, the features of the Project is to create a Paint Application.

Description about my Code

In order to create my Paint Application, I have created a class called Paint which has inherits **JFrame** class for creating JFrame and has implements two events(interfaces) called ActionListener and MouseMotionListener.

I have imported 3 things i.e. import javax.swing.\*, import java.awt.\* and import java.awt.events.

There is another external class called MyColorChooser in which main method is created. Inside the main method, frame is created having width and size equals to 700px. Title is also given for the frame i.e. “Paint Application.”

In the Paint class I have created a constructor where **JButton**, **Container**, **Cursor** and **Color** objects has been created. The Layout of the Container is set to **FlowLayout**. **JButton** size set to width 200px and height 20px along with distance from x-axis is 25px and y-axis is 0px. Cursor is ***HAND\_CURSOR*** which means when user hover over the Button , Cursor changes into a HAND.

Object of JButton i.e. b1 has assigned a event called ActionListener(b1.addActionListener). Container is assigned with another event called addMouseMotionListener.

Three methods have been created and is override as they are abstract method. They are as follows: -

1.public void actionPerformed (**ActionEvent** arg0)

2.public void mouseDragged (**MouseEvent** arg0)

3.public void mouseMoved (**MouseEvent** arg0) hasn’t been used.

Inside the **actionPerformed** (ActionEvent arg0) method which is of interface class ActionListener, I have created **JColorChooser** which takes 3 arguments i.e. component, String Title and initial color.

**MouseDragged** (ActionEvent arg0) method is of interface class MouseMotionListener. Inside it, I have created object of Graphics. **Graphics** object ‘g’ takes arguments ‘col’ which is object of JColorChooser which is used to draw a line in the frame with the help of below code: -

g.fillOval(arg0.getX( ), arg0.getY( ), width , height)

In this way, I have created a sample of Paint Application.

**Paint application code**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class Paint extends JFrame implements ActionListener, MouseMotionListener {

Container c;

JButton b1;

Color col;

Paint() {

c = this.getContentPane();

// Setting FlowLayout center

FlowLayout fl = new FlowLayout(FlowLayout.CENTER);

c.setLayout(fl);

// Creating Button

b1 = new JButton("Choose color");

b1.setBounds(250, 0, 200, 30);

// To set Cursor

Cursor crs = new Cursor(Cursor.HAND\_CURSOR);

b1.setCursor(crs);

// For font size of button

Font f = new Font("Arial", Font.BOLD, 20);

b1.setFont(f);

b1.addActionListener(this);

addMouseMotionListener(this);

c.add(b1);

}

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

col = JColorChooser.showDialog(this, "Select Color", Color.blue);

b1.setForeground(col);

}

@Override

public void mouseDragged(MouseEvent arg0) {

// TODO Auto-generated method stub

Graphics g = getGraphics();

g.setColor(col);

g.fillOval(arg0.getX(), arg0.getY(), 20, 20);

}

@Override

public void mouseMoved(MouseEvent arg0) {

// TODO Auto-generated method stub

}

}

public class MyColorChooser {

public static void main(String[] args) {

// TODO Auto-generated method stub

Paint frame = new Paint();

frame.setTitle("Paint Application");

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setBounds(500, 250, 700, 700);

}

}

OOP features implemented

For creating this Paint Application, different OOP features has been implemented like: -

1.Object and Class

2.Inheritance

3.Abstraction and Encapsulation

4.Polymorphism

References

For Creating this Paint Application, I have taken help from the following sources: -

1. <https://www.javatpoint.com/>
2. Programming with JAVA (Textbook)
3. YouTube

THE EnD