**Paint Brush Documentation**

**1 – Program description :**

This is a paint brush java application that provides the option to draw rectangle , oval or line in any color of (red , green , blue)

It also provides the option to draw using free hand in any color of the mentioned colors

It also provides the option to erase any shape the user drawed

It also provides the option to clear the whole screen and it provides undo option to go back the last action the user took at any time

**2 – My workflow in building the program :**

Firstly I created a simple applet program that draws rectangle with x , y , width , height

Then I created color buttons that determines the value of the color that I will use in drawing the rectangle and set it default value to black

Then I added new color attribute to the rectangle class

Then I created checkbox to determine if this rectangle will be filled with a specific color or not

Then I added Boolean attribute to the rectangle class which get the status of the checkbox

I created arraylist from the type rectangle to save the draawed rectangles in it and display them at the same time

Then I created the same component but for different shapes(oval , line ) and made all the shapes inherit from one class called shape and I changed the way of storing the shapes

I created arraylist of type shape and stored in it all the drawed shapes to display them in the same order that they are drawn in it

I created freehand class that extends shape class

It draws line but it update and save the end point with every drag so it will be freeline

I created eraser class thet extends shape class

It draws sequence of white small rectangles to erase the drawn shapes

I created clear method that clears the arraylist which stores all the shapes

I created stack that stores every drawing action the user take

And I used it to undo the last action the user did by deleting the last object from the shape arraylist

**3 – the role of each class and method in the code :**

**1 – painter class :** The main class which extends applet and contain the whole code

it contains declaration for many variables :

startx , starty , endx , endy : temporary variables I used them to store the mouse coordinates in every position

shapes : arrraylist of shape type

Actionstack : stack of Action type

And other buttons and variables that I will describe in the coming parts

**2 – shape class :** The parent abstract class that has four attributes x1 , y1 , x2 , y2

And abstract empty method called draw

**3- rectangle class** : child class that inherites the shape class and its constructor takes 6 parameters : x1 , x2 , y1 , y2 ,color , fillEnabled

fillEnabled : it is a Boolean variable which represents if the shape is filled or not

and the class have a draw function which checks on the value of fillEnabled variable and based on this value it draws a filled rectangle or unfilled rectangle

**4- oval class :** child class that inherites the shape class and its constructor takes 6 parameters : x1 , x2 , y1 , y2 ,color , fillEnabled

fillEnabled : it is a Boolean variable which represents if the shape is filled or not

and the class have a draw function which checks on the value of fillEnabled variable and based on this value it draws a filled oval or unfilled oval

**5 - line class :** child class that inherites the shape class and its constructor takes 5 parameters : x1 , x2 , y1 , y2 ,color

and the class have a draw function which draw line

**6 - freeHand class :** child class that inherites the shape class and its constructor takes 5 parameters : x1 , x2 , y1 , y2 ,color

and the class have a draw function which draw free line which is multiple connected lines

**7- eraser class :** it is a child class that inherites the shape class and its constructor takes 6 parameters : x1 , x2 , y1 , y2 ,color

and the class have a draw function which draws sequence of white small rectangles

**8 – init Method :** it is lifecycle method from the applet

The first two lines inside it was desiging the layout of button pane

Then I added two instances of mymouselistener class as a listener for mouse events and mouse motion events

Then I added label and listener to each button and checkbox in the program

**9 – paint Method:** it is lifecycle method from the applet

I used a for each loop that draws every shape from the shapes arraylist

The next block of code : I created a nested if else that starts by checking the value of the drawing variable

If it is true then the program checks the value of currentdrawing variable to determine which shape will be drawn and then it checks the value of fillEnabled variable to determine if it will be filled or not

The purpose of the last block of code is to display the current shape that is being drawn at the real time

**10 – setButtonSize Method :** it is a simple method which sets the size of the button

**11 – ColorButton class :** it is a simple class which extends Button class

It is responsible for drawing color on the colors buttons

**12 - MyMouseListener class :** it is a class which extends the mouseAdapter class which providethree main methods : mousePressed , mouseDragged , mouseReleased

**mousePressed :** it checks the value of the drawEnabled variable and if it is true the method execute this :

it is responsiple for saving the coordinates where the mouse clicked and then it sets the value of the drawing variable to true

**mouseDragged :** it checks the values of the drawing and drawEnabled variables and if the both are true it excutes this :

it responsible for storing and updating the coordinates of any dragging the user will do

and also it has a specific role in drawing freehand and eraser shapes

it stores the coordinates of endX , endY in startX , startY

the purpse of the previous part is making the display of freehanad or the eraser looks like a connected shape not only sperated parts

**mouseReleased :** it checks the values of the drawing and drawEnabled variables and if the both are true it excutes this :

it stores the coordinates where the user releases the mouse

and then sets the value of drawing variable to false

the next block of code checks the value of the currentdrawing to determine which shape is being drawn and adds this shape to the shapes arraylist and then stores every action in stack to use it in the undo part

**13 – rectangleListener class :** it is a class which called when the user press the rectangle button and sets the value of currentDrawing to 0 and sets drawEnabled to true

currentDrawing : it is an int variable which have a value represents the shape which I currently drawing at this moment

drawEnabled : it is a Boolean variable which takes initial value of false .. if its value equals true this means the user in the drawing mode

**14 – ovalListener class :** it is a class which called when the user press the oval button and sets the value of currentDrawing to 1 and sets drawEnabled to true

currentDrawing : it is an int variable which have a value represents the shape which I currently drawing at this moment

drawEnabled : it is a Boolean variable which takes initial value of false .. if its value equals true this means the user in the drawing mode

**15 – lineListener class :** it is a class which called when the user press the line button and sets the value of currentDrawing to 2 and sets drawEnabled to true

currentDrawing : it is an int variable which have a value represents the shape which I currently drawing at this moment

drawEnabled : it is a Boolean variable which takes initial value of false .. if its value equals true this means the user in the drawing mode

**16– freeHandListener class :** it is a class which called when the user press the freeHand button and sets the value of currentDrawing to 3 and sets drawEnabled to true

currentDrawing : it is an int variable which have a value represents the shape which I currently drawing at this moment

drawEnabled : it is a Boolean variable which takes initial value of false .. if its value equals true this means the user in the drawing mode

**17 – eraserListener class :** it is a class which called when the user press the eraser button and sets the value of currentDrawing to 4 and sets drawEnabled to true

currentDrawing : it is an int variable which have a value represents the shape which I currently drawing at this moment

drawEnabled : it is a Boolean variable which takes initial value of false .. if its value equals true this means the user in the drawing mode

**18 - clearListener class :** it is a class which called when the user press the clear button

It clears the shapes listArray which will store the drawn shapes

**19 – ColorListener class :** it is a class which called when the user press on any color button

its constructor takes color as a parameter and passes this value to method that called setcurrentColor

**20– fillCheckBoxListener class :** it is a class which called when the user press on the fillcheckbox button

It returns the status of the checkbox that will determine if the shape will be filled or not

**21 – setCurrentColor Method :** it is a simple function which responsible for passing the value of a specific color to the currentColor variable

**22 – getCurrentColor Method:** it is a simple method which responsible for returning the value of the currentColor variable

**23 - getfillEnabled Method:** it is a simple method which returns the value of the fillEnabled variable

**24 – undoListener class :**

It us a class which called when the user press on the undo button

It uses the actionstack to get the last action the user did and remove it from the arraylist shapes

**25 – Action class :** it is a class that represents a drawing action and its constructor takes a parameter of shape type

**26 – itemStateChanged Method:** it is an empty method but I wrote it because the painter class implements the ItemListener class then it is mandatory to override this method

**Note :** I tried to provide the undo option (as I mentioned before) but it only applied to shapes(rectangle , oval , line )

But unfortunately I could not apply it to freehand and eraser however my tries so I wanted to clarify that