IP - Mini Project
Noise Morpher

Vishal Naidu - 38

Project Abstract

The above mentioned project's objective is to create a separate image by morphing the noise contained within an image with another. Using percentage morphing techniques, one can control the amount of morphing to do with each source. The screenshots of the images pasted below also contain the percentages of each source as captions. Image pixel extraction using the library ImageIO extracts each pixel in the negative format.

Program Code

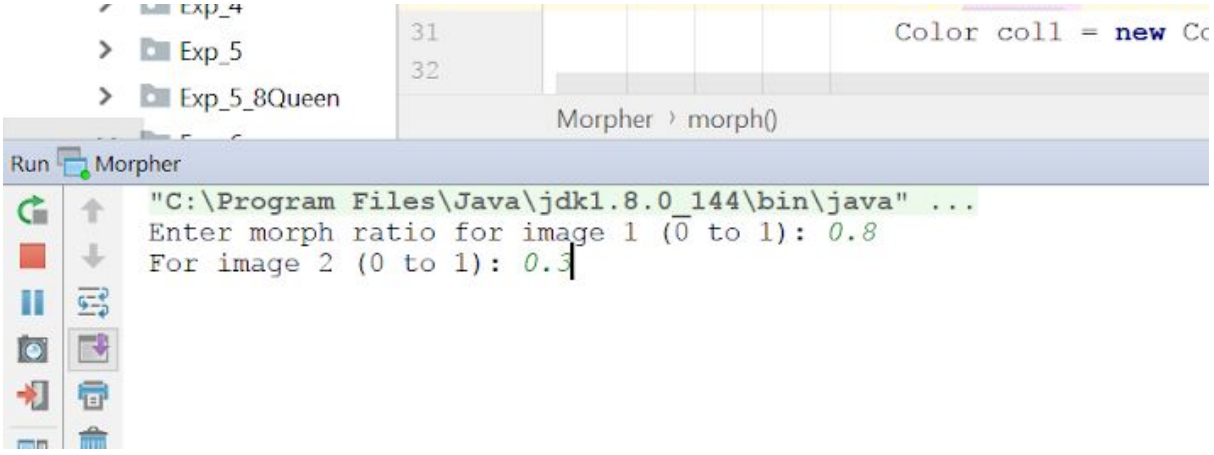
Image Morpher in Java

```
package IPMiniProject;

import java.awt.*;
import java.awt.image.BufferedImage;
import java.awt.image.ConvolveOp;
import java.awt.image.Kernel;
import java.io.File;
import java.util.Scanner;
import javax.imageio.ImageIO;

public class Morpher
{
   public static BufferedImage morph(BufferedImage src1, float ratio1,
BufferedImage src2, float ratio2)
   {
       int width = src1.getWidth();
       int height = src1.getHeight();

       BufferedImage img = new BufferedImage(
               width,
               height,
               BufferedImage.TYPE_INT_RGB
       );

       try
       {
           for(int i=0;i<height;i++)
           {
```

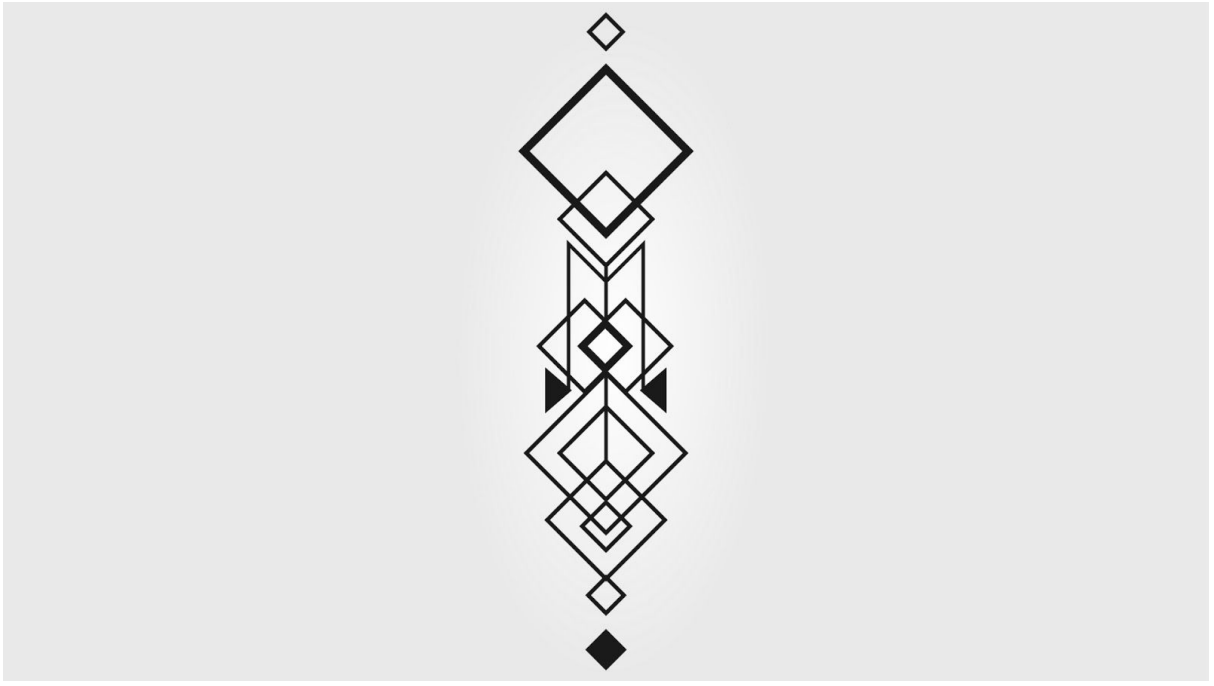
```
                for(int j=0;j<width;j++)
                {
                    int intcol1 = src1.getRGB(j, i);
                    Color col1 = new Color(intcol1, true);

                    int intcol2 = src2.getRGB(j, i);
                    Color col2 = new Color(intcol2, true);

                    int RGBout = (int)((col1.getRGB()*ratio1 +
col2.getRGB()*ratio2)/(ratio1+ratio2));
                    System.out.println(RGBout);
                    img.setRGB(j, i, RGBout);
//                      img.setRGB(j, i, 0);
                }
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }

        return img;
    }



    public static void main(String[] args) throws Exception
    {
        Scanner sc = new Scanner(System.in);

        if(args[0] == null || args[1] == null)
        {
            System.out.println("INSUFFICIENT INPUTS!\nEXITING...");
        }

        System.out.print("Enter morph ratio for image 1 (0 to 1): ");
        float ratio1 = sc.nextFloat();
        if(ratio1>1 || ratio1<0)
        {
            ratio1 = 0.5f;
        }

        System.out.print("For image 2 (0 to 1): ");
        float ratio2 = sc.nextFloat();
        if(ratio2>1 || ratio2<0)
        {
            ratio2 = 0.5f;
        }


        BufferedImage src1 = ImageIO.read(new File(args[0]));
```

```
        BufferedImage src2 = ImageIO.read(new File(args[1]));


        BufferedImage bfImg = morph(src1, ratio1, src2, ratio2);
        ImageIO.write(bfImg, "jpeg",new File("E:\\ViperOut.jpg"));
    }
}
```

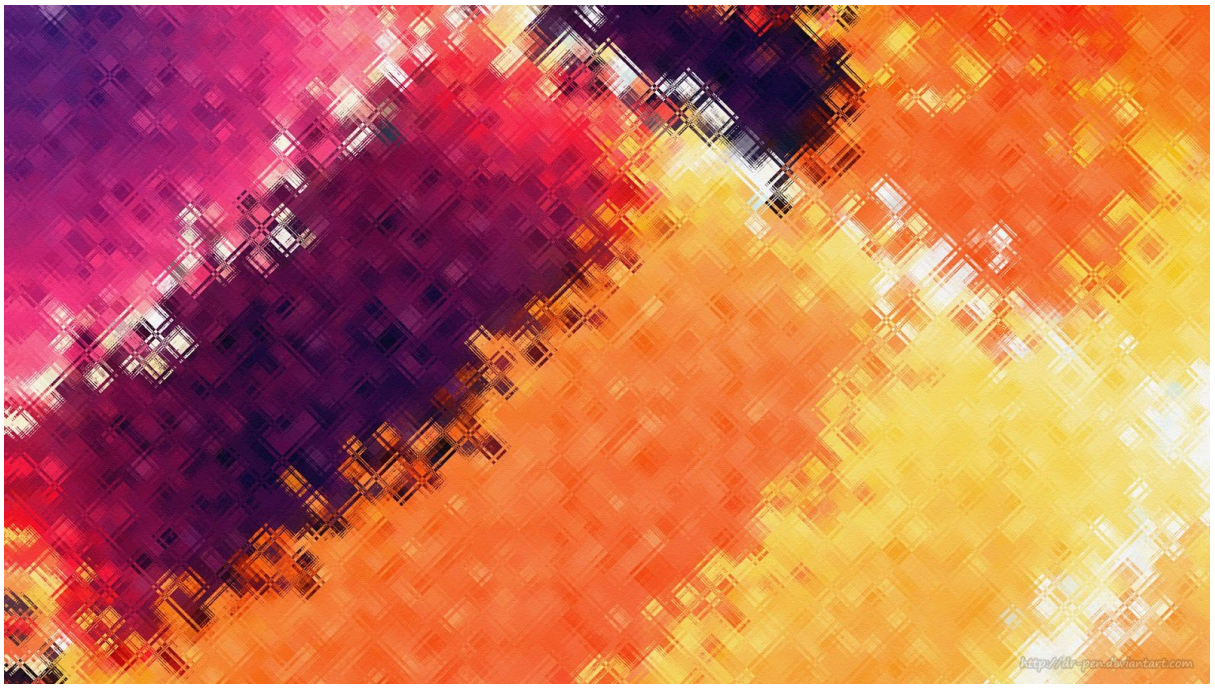
Program Execution

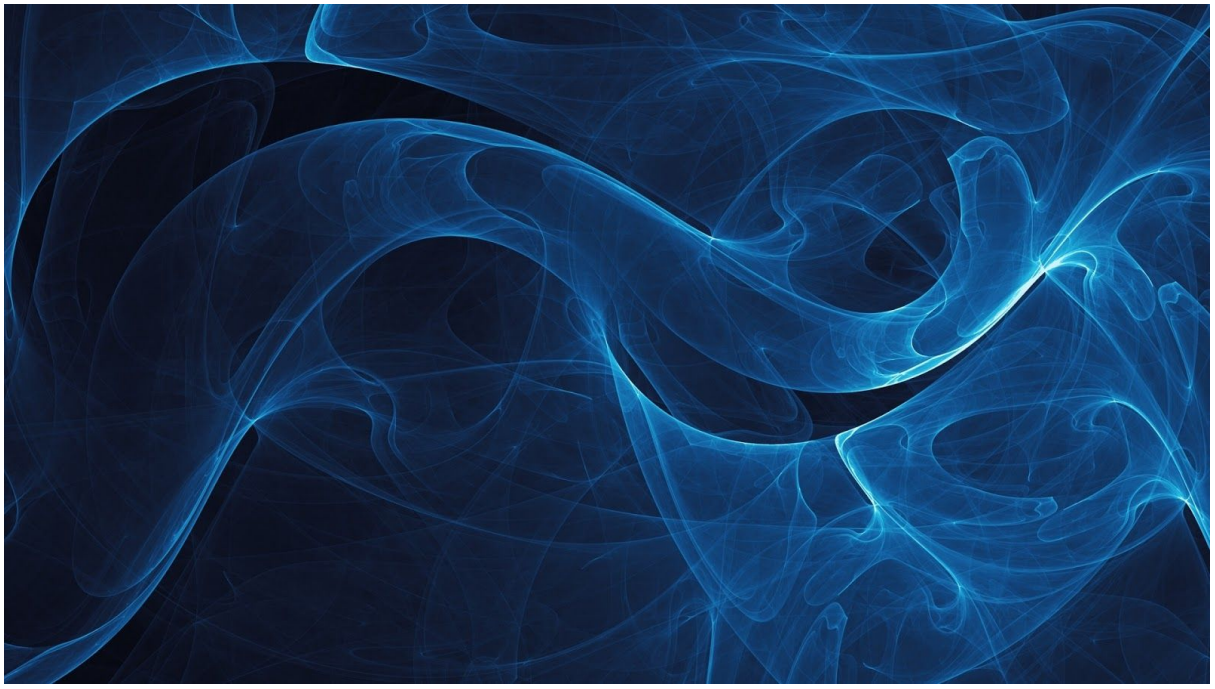


Source 1



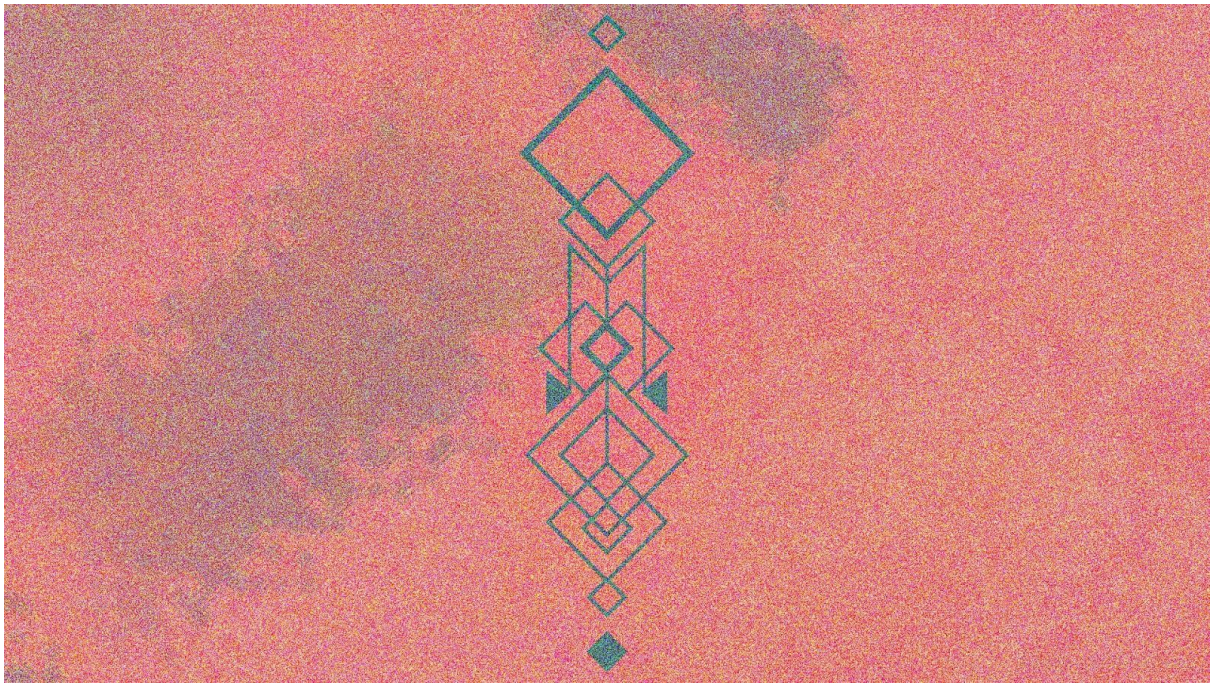
Source 2





OUTPUT ViperOut.jpg

(0.7 & 0.3)



(0.99995 & 0.00005)