Encryptor Explained

```java
package me.postmus.joris;

import java.awt.Color;

public class Encryptor {
    int width = 200;
    int height = 200;
    BufferedImage img = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
    private static final DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy.MM.dd HH.mm.ss");
    File f = null;
    JFrame frame = new JFrame("Code");
    Color backgroundColor = new Color(119,255,245);

    public String readFile(String pathname) throws IOException {
        File file = new File(pathname);
        StringBuilder fileContents = new StringBuilder((int)file.length());
        Scanner scanner = new Scanner(file);
        String lineSeparator = System.getProperty("line.separator");

        try {
            while(scanner.hasNextLine()) {
                fileContents.append(scanner.nextLine() + lineSeparator);
            }
            return fileContents.toString();
        } finally {
            scanner.close();
        }
    }

    public int[][] encryptText(String text) {
        int textLength = text.length();
        int[][] encryptedText = new int[textLength/3+1][3];

        int currentPixel = -1;
        for(int i = 0; i < textLength; i++) {
            char curChar = text.charAt(i);
            int charNumeric = Character.getNumericValue(curChar);
            if (i%3 == 0) {currentPixel++;}
            if (charNumeric > 0 && charNumeric < 36) {
                encryptedText[currentPixel][i%3] = charNumeric*7;
            }else {
                encryptedText[currentPixel][i%3] = 1;
            }
        }
        return encryptedText;
    }

    public BufferedImage generateImg(int[][] encryptedText) {
        int currentPixel = 0;
        int endStatement = 0;
        int pixelLength = encryptedText.length;
        for(int y = 0; y < height; y++) {
          for(int x = 0; x < width; x++) {
              if (currentPixel < pixelLength) {
              //Actual Words
              int r = encryptedText[currentPixel][0]; //red
              int g = encryptedText[currentPixel][1]; //green
              int b = encryptedText[currentPixel][2]; //blue
              Color color = new Color(r, g, b);
              int p = color.getRGB();
              img.setRGB(x, y, p);
              currentPixel++; } else {
              //Random
                  if(endStatement == 0) {endStatement = 1; img.setRGB(x, y, new Color(0,200,134).getRGB());} else {
              int r = (int)(Math.random()*256); //red
              int g = (int)(Math.random()*256); //green
              int b = (int)(Math.random()*256); //blue
              Color color = new Color(r, g, b);
              int p = color.getRGB();
              img.setRGB(x, y, p);
                      }
              }
          }
        }

        try{
            Window window = new Window();
            LocalDateTime now = LocalDateTime.now();

            f = new File(window.getDirectory() + "JPCode " + dtf.format(now) + ".png");
            ImageIO.write(img, "png", f);
            System.out.println("Printing on " + f.getPath());
            System.out.println("Encrypted Image succesfully printed.");
        }catch(IOException e){
            System.out.println("Error: " + e);
        }
        return img;
    }
```

Algorithm that takes the path of a text file and then reads it and returns a string value

This algorithm takes a string and turns every letter into a number and then puts it into a 2D array

This chunk of code loops through every character in the string and then turns it into a Colored value (from 0 to 255) and then puts it into a 2D array

This algorithm takes a 2D array and then turns it into a JPCode image

This chunk of code loops through each color and then turns it into a Colored pixel on the grid

This chunk of code is called when there are no more characters left but there is still space left so it generates randomly colored pixels

This chunk of code saves the image on the Hard Drive

Decryptor Explained

```java
package me.postmus.joris;

import java.awt.image.BufferedImage;

public class Decryptor {

    public void print2DArray(int[][] array) {
        for(int r=0; r<array.length; r++) {
               for(int c=0; c<array[r].length; c++)
                   System.out.print(array[r][c] + " ");
               System.out.println();
            }
    }

    public String arrayToString(int[][] pixelArray) {
        String encryptedText = "";

        check:
        for(int y = 0; y < 200; y++) {
            for(int x = 0; x < 200; x++) {
                int currentRGB = pixelArray[x][y];

                int red = (currentRGB >> 16) & 0xFF;
                int green = (currentRGB >> 8) & 0xFF;
                int blue = currentRGB & 0xFF;

                if (red == 0) {break check;} else {
                encryptedText = encryptedText + getChar(red);
                encryptedText = encryptedText + getChar(green);
                encryptedText = encryptedText + getChar(blue);
                }
            }
        }

        return encryptedText;
    }

    public int[][] pictureToArray(BufferedImage img) {
        int[][] pixelArray = new int[200][200];
        for (int y = 0; y < 200; y++) {
            for (int x = 0; x < 200; x++) {
                pixelArray[x][y] = img.getRGB(x, y);
            }
        }

        return pixelArray;
    }

    public char getChar(int number) {
        if (number > 1) {
            int curChar = number / 7;
            //Loop through all letters of alphabet
            if(curChar == 10) {return 'a';} else
            if(curChar == 11) {return 'b';} else
            if(curChar == 12) {return 'c';} else
            if(curChar == 13) {return 'd';} else
            if(curChar == 14) {return 'e';} else
            if(curChar == 15) {return 'f';} else
            if(curChar == 16) {return 'g';} else
            if(curChar == 17) {return 'h';} else
            if(curChar == 18) {return 'i';} else
            if(curChar == 19) {return 'j';} else
            if(curChar == 20) {return 'k';} else
            if(curChar == 21) {return 'l';} else
            if(curChar == 22) {return 'm';} else
            if(curChar == 23) {return 'n';} else
            if(curChar == 24) {return 'o';} else
            if(curChar == 25) {return 'p';} else
            if(curChar == 26) {return 'q';} else
            if(curChar == 27) {return 'r';} else
            if(curChar == 28) {return 's';} else
            if(curChar == 29) {return 't';} else
            if(curChar == 30) {return 'u';} else
            if(curChar == 31) {return 'v';} else
            if(curChar == 32) {return 'w';} else
            if(curChar == 33) {return 'x';} else
            if(curChar == 34) {return 'y';} else
            if(curChar == 35) {return 'z';} else
            {return ' ';}
        }else {return ' ';}
    }
}
```

This algorithm takes a 2D array with colours and turns them into letters after which it puts it into a string

This part reads the R,G,B values off a pixel

This part then turns those values into letters and adds it to the final string

This algorithm takes a JPCode image and then turns it into a 2D array holding all it's individual pixel's

This algorithm takes a r,g,b value (0-255) and then turns it into a letter