CT255 Assignment 3

```
/*
 * @author Dawid Szczesny
 * @id 21300293
 */

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Stegano1 {

    public static void main(String[] args) {

        String arg1, arg2, arg3, arg4;
        Boolean err = false;

        if (args != null && args.length > 1) { // Check for minimum number of arguments
            arg1 = args[0];
            arg2 = args[1];
            if (arg2.equals("")) {
                err = true;
            } else if (arg1.equals("A") && args.length > 3) {
                // Get other arguments
                arg3 = args[2];
                arg4 = args[3];
                if (arg3.equals("") || arg4.equals("")) {
                    err = true;
                } else {
                    // Hide bitstring
                    hide(arg2, arg3, arg4);
                }
            } else if (arg1.equals("E")) {
                // Extract bitstring from text
                retrieve(arg2);
            } else {
                err = true;
            }
        } else {
            err = true;
        }

        if (err == true) {
            System.out.println();
            System.out.println("Use: Stegano1 <A:E><Input File><OutputFile><Bitstring>");
            System.out.println("Example: Stegano1 A inp.txt out.txt 0010101");
```

```
            System.out.println("Example: Stegano1 E inp.txt");

        }
    }

    // hides binString into the inpFile and outputs it into outFile
    static void hide(String inpFile, String outFile, String binString) {
        //
        BufferedReader reader;
        BufferedWriter writer;

        try {
            reader = new BufferedReader(new FileReader(inpFile));
            writer = new BufferedWriter(new FileWriter(outFile));
            String line = reader.readLine();

            // check if binString is odd
            // and add a 0 at the end if it is
            binString = binString.length() % 2 == 1 ? binString + "0" : binString;

            int index = 0;
            // changes binString to an array of characters
            char binStringArr[] = binString.toCharArray();
            while (line != null) {
                // Your code starts here
                // checks if there is still binString to loop through

                // if the line is empty skip
                // occurs on newlines, paragraph separators
                if (line.equals("")) {
                    writer.newLine();
                    line = reader.readLine();
                }

                // checks if there's binary string to loop through
                if (index <= binStringArr.length - 1) {
                    if (binStringArr[index] == '0') {
                        line += " ";
                    } else {
                        line += "  ";
                    }

                    // problem 2 part
                    // ------------------------------------------------------------------
-------------
                    // u200f and u200e are both invisible characters and will not appear i
n a text
                    // editor
                    // u200f represents 0
                    // u200e represents 1

                    // line.replaceFirst replaces the first instance of a space and replac
es it with
                    // either
```

```
                    // \u200f or \u200e
                    if (binStringArr[index + 1] == '0') {
                        line = line.replaceFirst(" ", " \u200f");
                    } else {
                        line = line.replaceFirst(" ", " \u200e");
                    }
                    // --------------------------------------------------------------------
----------------
                }

                // Store amended line in output file
                writer.write(line);
                writer.newLine();
                // index++ for problem 1
                index += 2; // we increment by 2 as we are encrypting 2 bits at a time
                // read next line
                line = reader.readLine();
            }
            reader.close();
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    // finds our encryption from a file
    static void retrieve(String inpFile) {
        BufferedReader reader;
        String bitString = "";
        try {
            reader = new BufferedReader(new FileReader(inpFile));
            String line = reader.readLine();
            while (line != null) {
                // System.out.println(line);
                // Your code starts here

                if (line.length() - 1 > 0) {
                    // gets only the last character
                    String foo = line.substring(line.length() - 2);
                    if (foo.contains("  ")) {
                        bitString += "1";
                    } else if (foo.contains(" ")) {
                        bitString += "0";
                    }

                }

                // problem 2 part
                // ------------------------------------------------------------
                // checks if the line contains our invisible characters
                // if it does we retrieve them and check whether
                // they are equal to 1 or 0
                // \u200f = 0
```

```
                // \u200e = 1
                if (line.contains("\u200f")) {
                    bitString += "0";
                } else if (line.contains("\u200e")) {
                    bitString += "1";
                }
                // ---------------------------------------------------------------

                // read next line
                line = reader.readLine();

            }
            System.out.println(bitString);
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}
```