

Dec 06, 16 13:44

AssEx2.java

Page 1/1

```
/**
 * Programming AE2
 * Creates and shows the cipher GUI
 */
public class AssEx2
{
    /**
     * The main method
     * @param args the arguments
     */
    public static void main(String [] args)
    {
        CipherGUI CipherGUI = new CipherGUI();
        CipherGUI.setVisible(true);
    }
}
```

Dec 06, 16 16:52

CipherGUI.java

Page 1/7

```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;
import java.util.Scanner;

/**
 * Programming AE2
 * Class to display cipher GUI and listen for events
 */
public class CipherGUI extends JFrame implements ActionListener
{
    //instance variables which are the components
    private JPanel top, bottom, middle;
    private JButton monoButton, vigenereButton;
    private JTextField keyField, messageField;
    private JLabel keyLabel, messageLabel;

    //application instance variables
    //including the 'core' part of the textfile filename
    //some way of indicating whether encoding or decoding is to be done

    private String keyword;
    private String messagefilename;
    boolean encodeordecode;

    /**
     * The constructor adds all the components to the frame
     */
    public CipherGUI()
    {
        this.setSize(400,150);
        this.setLocation(100,100);
        this.setTitle("Cipher GUI");
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.layoutComponents();
    }

    /**
     * Helper method to add components to the frame
     */
    public void layoutComponents()
    {
        //top panel is yellow and contains a text field of 10 characters
        top = new JPanel();
        top.setBackground(Color.yellow);
        keyLabel = new JLabel("Keyword:");
        top.add(keyLabel);
        keyField = new JTextField(10);
        top.add(keyField);
        this.add(top, BorderLayout.NORTH);

        //middle panel is yellow and contains a text field of 10 characters
        middle = new JPanel();
        middle.setBackground(Color.yellow);
        messageLabel = new JLabel("Message file:");
        middle.add(messageLabel);
        messageField = new JTextField(10);
        middle.add(messageField);
        this.add(middle, BorderLayout.CENTER);

        //bottom panel is green and contains 2 buttons
        bottom = new JPanel();
        bottom.setBackground(Color.green);
        //create mono button and add it to the top panel
        monoButton = new JButton("Process Mono Cipher");
        monoButton.addActionListener(this);

```

Dec 06, 16 16:52

CipherGUI.java

Page 2/7

```

        bottom.add(monoButton);
        //create vigenere button and add it to the top panel
        vigenereButton = new JButton("Process Vigenere Cipher");
        vigenereButton.addActionListener(this);
        bottom.add(vigenereButton);
        //add the top panel
        this.add(bottom, BorderLayout.SOUTH);
    }

    /**
     * Listen for and react to button press events
     * (use helper methods below)
     * @param e the event
     */
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource().equals(monoButton))//response in the case that the
        monocipher button is pressed
        {
            if(!getKeyword()&&processFileName()&&processFile(false)&&checkD
uplicateChar())//process the file and get the keyword and also validate the dat
a checking that the keyword does not have duplicate characters
            {
                JOptionPane.showMessageDialog(this, "Please enter a valid keywor
d or a valid file name. A valid keyword has only capital letters and no duplicate characters");
            }
        }
        else if(e.getSource().equals(vigenereButton))//response in the case
that the Vigenere cipher is pressed
        {
            if(!getKeyword()&&processFileName()&&processFile(true))//proce
ss the file and get the keyword and also validate the data
            {
                JOptionPane.showMessageDialog(this, "Please enter a valid keywor
d or a valid file name. A valid keyword has only capital letters");
            }
        }
    }

    /**
     * Obtains cipher keyword
     * If the keyword is invalid, a message is produced
     * @return whether a valid keyword was entered
     */
    private boolean getKeyword()
    {
        keyword=keyField.getText();
        int index=0;
        //check for that the keyword is all uppercase
        for (int i=0; i<keyword.length(); i++)
        {
            index=(int)((keyword.charAt(i))-'A');
            if((index<0)|| (index>25))
            {
                return false;
            }
        }

        //check that there is a keyword
        if((keyword!=null)&&(keyword!=""))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

```

Dec 06, 16 16:52

CipherGUI.java

Page 3/7

```

/**
 * Obtains filename from GUI
 * The details of the filename and the type of coding are extracted
 * If the filename is invalid, a message is produced
 * The details obtained from the filename must be remembered
 * @return whether a valid filename was entered
 */
private boolean processFileName()
{
    messagefilename=messageField.getText();

    //check whether or not to perform an encryption or decryption based
    //on the filename otherwise it informs the user that the filename does not follow
    //the convention
    if((messagefilename.charAt(messagefilename.length()-1))=='P')
    {
        encodeordecode=true;
    }
    else if((messagefilename.charAt(messagefilename.length()-1))=='C')
    {
        encodeordecode=false;
    }
    else
    {
        JOptionPane.showMessageDialog(this, "Please use the correct filename convention");
        return false;
    }

    //check if there is a message name
    if((messagefilename!=null)&&(messagefilename!=""))
    {
        return true;
    }
    else
    {
        return false;
    }
}

//method to check for duplicate characters in a keyword
//returns true if there are no duplicate characters and false if there are
//this is only a necessary check for the monocipher
private boolean checkDuplicateChar()
{
    char[] alphabetcounts=new char[26];

    //create alphabet
    alphabetcounts = new char [26];
    for (int i = 0; i < 26; i++)
        alphabetcounts[i] = 0;

    //count the each of the characters in keyword
    for (int j=0; j<keyword.length(); j++)
    {
        if(((int)(keyword.charAt(j)-'A')>=0)&&(((int)(keyword.charAt(j)-'A')<26)))
        {
            alphabetcounts[((int)(keyword.charAt(j)-'A'))]++;
        }
    }

    //check for duplicate characters in the keyword

```

Dec 06, 16 16:52

CipherGUI.java

Page 4/7

```

    boolean duplicatenotpresent=true;

    for (int j=0; j<26; j++)
    {
        if(alphabetcounts[j]>1)// if a character occurs more than
        //once in the keyword then we have a duplication that the monoalphabetic cipher
        //cannot handle
        {
            duplicatenotpresent=false;
        }
    }

    return duplicatenotpresent;
}

/**
 * Reads the input text file character by character
 * Each character is encoded or decoded as appropriate
 * And written to the output text file
 * @param vigenere whether the encoding is Vigenere (true) or Mono (false)
 * @return whether the I/O operations were successful
 */
private boolean processFile(boolean vigenere)
{
    FileReader reader;
    FileWriter writercipher;
    FileWriter writereport;
    try {

        writereport = new FileWriter(messagefilename.substring(0,
        messagefilename.length()-1)+"F.txt");

        if(vigenere)
        {
            //check if the user wants to encode or decode and
            //create the appropriate file readers and writers
            if(encodeordecode)
            {
                reader=new FileReader(messagefilename.substring(0, messagefilename.length()-1)+"P.txt");
                writercipher=new FileWriter(messagefilename.substring(0, messagefilename.length()-1)+"C.txt");
            }
            else
            {
                reader=new FileReader(messagefilename.substring(0, messagefilename.length()-1)+"C.txt");
                writercipher=new FileWriter(messagefilename.substring(0, messagefilename.length()-1)+"D.txt");
            }
        }

        //set up the vigenere cipher instance and
        //the letterfrequencies instance
        VCipher vcipher=new VCipher(keyword);
        LetterFrequencies rep=new LetterFrequencies();

        char currentchar=' ';
        char currentcipherchar=' ';

        int c=' ';
        int i=0;
        int keywordletterindex=0;

        while(true)
        {

```

Dec 06, 16 16:52	CipherGUI.java	Page 5/7
	<pre> c=reader.read(); if(c!=-1) { break; } //convert the character read in integer format into char format currentchar=(char) c; //integer to check if the charac ter is an upper-case letter if not then don't increment through the keyword int relativeASCII=(int)(currentc har-'A'); ASCII<26)) { keywordletterindex=i%(ke yword.length()); i++; //encode or decode the current c haracter if(encodeordecode) { currentcipherchar=vciphe r.encode(currentchar, keywordletterindex); } else { currentcipherchar=vciphe r.decode(currentchar, keywordletterindex); } rep.addChar(currentcipherchar); //write to the cipher file or pl aintext file writercipher.write(currentcipher char); } //generate the report string String report=rep.getReport(); //write the string to the file that was created writereport.write(report); //close the various readers and writers reader.close(); writereport.close(); writercipher.close(); return true; } else { //check if the user wants to encode or d ecode and create the appropriate file readers and writers if(encodeordecode) { reader=new FileReader(messagefil ename.substring(0, messagefilename.length()-1)+"P.txt"); writercipher=new FileWriter(mess agefilename.substring(0, messagefilename.length()-1)+"C.txt"); </pre>	

Dec 06, 16 16:52	CipherGUI.java	Page 6/7
	<pre> else { reader=new FileReader(messagefil ename.substring(0, messagefilename.length()-1)+"C.txt"); writercipher=new FileWriter(mess agefilename.substring(0, messagefilename.length()-1)+"D.txt"); } //set up the monocipher and the letterfr equencies instances MonoCipher mcipher=new MonoCipher(keywor d); LetterFrequencies rep=new LetterFrequenc ies(); //initialise character variables char currentchar=' '; char currentcipherchar=' '; int c=' '; while(true) { c=reader.read(); if(c!=-1) { break; } //convert the character read in currentchar=(char) c; //encode and then decode the cur rent character if(encodeordecode) { currentcipherchar=mciphe r.encode(currentchar); } else { currentcipherchar=mciphe r.decode(currentchar); } //add to the appropriate charact er count rep.addChar(currentcipherchar); //write to the cipher file or pl aintext file writercipher.write(currentcipher char); } //generate the report string String report=rep.getReport(); //write the string to the file that was created writereport.write(report); //close the various readers and writers reader.close(); writereport.close(); writercipher.close(); return true; </pre>	

Dec 06, 16 16:52

CipherGUI.java

Page 7/7

```
        }  
        catch (Exception e)  
        {  
            e.printStackTrace();  
            return false;  
        }  
    }  
}
```

Dec 06, 16 13:35

LetterFrequencies.java

Page 1/2

```

/**
 * Programming AE2
 * Processes report on letter frequencies
 */
public class LetterFrequencies
{
    /** Size of the alphabet */
    private final int SIZE = 26;

    /** Count for each letter */
    private int [] alphaCounts;

    /** The alphabet */
    private char [] alphabet;

    /** Average frequency counts */
    private double [] avgCounts = {8.2, 1.5, 2.8, 4.3, 12.7, 2.2, 2.0, 6.1,
7.0,
                                0.2, 0.8, 4.0, 2.
4, 6.7, 7.5, 1.9, 0.1, 6.0,
                                6.3, 9.1, 2.8
, 1.0, 2.4, 0.2, 2.0, 0.1};

    /** Character that occurs most frequently */
    private char maxCh;

    /** Total number of characters encrypted/decrypted */
    private int totChars;

    private int maxindex;
    /**
     * Instantiates a new letterFrequencies object.
     */
    public LetterFrequencies()
    {
        alphaCounts=new int[SIZE];
        for (int i = 0; i < SIZE; i++)
            alphaCounts[i] = 0;

        alphabet=new char[SIZE];
        for (int i = 0; i < SIZE; i++)
            alphabet[i] = (char)('A' + i);
    }

    /**
     * Increases frequency details for given character
     * @param ch the character just read
     */
    public void addChar(char ch)
    {
        //calculate the index of the character within the alphabet array
        int index=(int)(ch-'A');

        //check that the character is an uppercase letter and if it is t
hen increment the appropriate element of the count array
        if(!((index<0)|| (index>25)))
        {
            totChars++;
            alphaCounts[index]++;
        }
    }

    /**
     * Gets the maximum frequency
     * @return the maximum frequency
     */
    private double getMaxPC()
    {

```

Dec 06, 16 13:35

LetterFrequencies.java

Page 2/2

```

        int max=0;//set the max to zero initially

        for(int i=0; i<SIZE; i++)
        {
            if(alphaCounts[i]>=max)//using greater or equal than ens
ures that the variable max is updated even when a particular characters count is
the same as the current maximum; as a result the maxindex will be the last in t
he alphabet with a character count equal to the max
            {
                max=alphaCounts[i];//if the current element of t
he array is greater than the current maximum then set the maximum to that value
                maxindex=i;
            }

        }

        return max*100/(double)totChars;
    }

    /**
     * Returns a String consisting of the full frequency report
     * @return the report
     */
    public String getReport()
    {
        //the title and column names are set out at the beginning before
the loop goes through and adds to the string called report
        //the string will then be printed to a file in the cipherGUI cla
ss

        String report="LETTER ANALYSIS\r\nLetter\tFreq\tFreq%\tAvgFreq\tDiff\r\n";
        for (int j=0; j<SIZE; j++)
        {
            report+=String.format("%c\t%dt\t%.2f\t%.2f\t%.2f\n", alphabet[j
], alphaCounts[j], alphaCounts[j]*100/((double)totChars), avgCounts[j], alphaCou
nts[j]*100/((double)totChars)-avgCounts[j]);
        }

        double maxpc=getMaxPC();
        report+=String.format("The character with the largest percentage is %c with a percen
tage frequency of %.2f", alphabet[maxindex], maxpc);
        return report; // replace with your code
    }
}

```

Dec 06, 16 14:58

MonoCipher.java

Page 1/2

```

/**
 * Programming AE2
 * Contains monoalphabetic cipher and methods to encode and decode a character.
 */
public class MonoCipher
{
    /** The size of the alphabet. */
    private final int SIZE = 26;

    /** The alphabet. */
    private char [] alphabet;

    /** The cipher array. */
    private char [] cipher;

    /**
     * Instantiates a new mono cipher.
     * @param keyword the cipher keyword
     */
    public MonoCipher(String keyword)
    {
        //create alphabet
        alphabet = new char [SIZE];
        for (int i = 0; i < SIZE; i++)
            alphabet[i] = (char)('A' + i);

        //create a cipher array
        cipher=new char[26];
        int k=25;
        int match=0;
        int i=0;

        //generate the cipher array based on the keyword and the
        monocipher scheme
        while(i<SIZE)
        {
            if(i<keyword.length())//while the array index is
            within the bounds of the keyword populate the array with the correspond charact
            er in the keyword
            {
                cipher[i]=keyword.charAt(i);
                System.out.print(cipher[i]);
                i++;
            }
            else
            {
                for(int m=0; m<keyword.length(); m++)
                {
                    if(keyword.charAt(m)==alphabet[k]
                    {
                        match++;
                        break;
                    }
                }
                if(match==0)
                {
                    cipher[i]=alphabet[k];
                    System.out.print(cipher[i]);
                    i++;
                }
                k--;
            }
            match=0;
        }

        // create first part of cipher from keyword

```

Dec 06, 16 14:58

MonoCipher.java

Page 2/2

```

        // create remainder of cipher from the remaining characters of t
        he alphabet
        // print cipher array for testing and tutors

        /**
         * Encode a character
         * @param ch the character to be encoded
         * @return the encoded character
         */
        public char encode(char ch)
        {
            //initialise the index to an impossible array value to indicate
            whether or not the character was found
            int index=(int)(ch-'A');

            //check if the character was found in the alphabet and then return t
            he corresponding encrypted character otherwise do nothing
            if((index>=0)&&(index<26))
            {
                return cipher[index];
            }
            else
            {
                return ch;
            }
        }

        /**
         * Decode a character
         * @param ch the character to be encoded
         * @return the decoded character
         */
        public char decode(char ch)
        {
            //initialise the index to an impossible array value to indicate
            whether or not the character was found
            int index=-1;

            //find the index of the input character using a linear search
            for(int i=0; i<SIZE ; i++)
            {
                if(ch==cipher[i])
                {
                    index=i;
                    break;
                }
            }

            //check if the character was found in the cipher alphabet and then ret
            urn the corresponding decrypted character otherwise do nothing
            if(index!=-1)
            {
                return alphabet[index];
            }
            else
            {
                return ch;
            }
        }
    }

```

Dec 06, 16 13:34

VCipher.java

Page 1/2

```

/**
 * Programming AE2
 * Class contains Vigenere cipher and methods to encode and decode a character
 */
public class VCipher
{
    private char [] alphabet; //the letters of the alphabet
    private final int SIZE = 26;
    private String keyword;
    // more instance variables

    /**
     * The constructor generates the cipher
     * @param keyword the cipher keyword
     */
    public VCipher(String keyword)
    {
        this.keyword=keyword;
        //create alphabet
        alphabet = new char [SIZE];
        for (int i = 0; i < SIZE; i++)
            alphabet[i] = (char)('A' + i);
    }

    /**
     * Encode a character
     * @param ch the character to be encoded
     * @return the encoded character
     */
    public char encode(char ch, int keywordletterindex)
    {
        //get the indices of the current keyword character and the current message character
        int keyindex=(int)(keyword.charAt(keywordletterindex)-'A');
        int messindex=(int)(ch-'A');
        char encoded=' ';

        //check that the characters corresponding to these indices are capital letters, if they are encode them otherwise do nothing
        if(((keyindex>=0)&&(keyindex<26))&&((messindex>=0)&&(messindex<26)))
        {
            encoded=alphabet[(keyindex+messindex)%26];
        }
        else
        {
            encoded=ch;
        }
        return encoded;
    }

    /**
     * Decode a character
     * @param ch the character to be decoded
     * @return the decoded character
     */
    public char decode(char ch, int keywordletterindex)
    {
        //get the indices of the current keyword character and the current message character
        int keyindex=(int)(keyword.charAt(keywordletterindex)-'A');
        int messindex=(int)(ch-'A');
        char decoded=' ';

        //check that the characters corresponding to these indices are capital letters, if they are encode them otherwise do nothing
        if(((keyindex>=0)&&(keyindex<26))&&((messindex>=0)&&(messindex<26)))
        {
            decoded=alphabet[(messindex-keyindex+26)%26];
        }
    }
}

```

Dec 06, 16 13:34

VCipher.java

Page 2/2

```

    }
    else
    {
        decoded=ch;
    }

    return decoded;
}

```