

Gui verbessern

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
```

```
/**
 *
 * Beschreibung.
 *
 * @version 1.0 vom 28.01.2006
 * @author
 */
```

Variablennamen
sinnvoll vorgeben

```
public class Gui extends JFrame {
    // Anfang Variablen
```

```
    private JLabel labelEingabe, labelSchluessel, labelAusgabe;
    private JTextField editSchluessel;
    private JButton codiereButton, decodiereButton, endeButton;
    private JTextArea ausgabe;
    private TextInterface text;
    private static VigenereInterface vigenere;
```

```
// Ende Variablen
```

```
/**
 * Konstruktor
 */
private Gui(String title, int breite, int hoehe) {
    // Frame-Initialisierung
    super(title);
    text=new Text();
    setJMenuBar(machMenueLeiste());
    setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent evt) { beendenAction(); }
    });
```

Menu in
eigene
Methode
ausgliedern

Aktionen
definieren,
die von
mehreren
Stellen
aufgerufen
werden
können

```
// Attribute
setSize(breite, hoehe);
setLocation(20, 20);
Container contentPane = getContentPane();
contentPane.setLayout(null);
```

Position
definieren

```
// Anfang Komponenten
labelEingabe = new JLabel("Eingabe");
labelEingabe.setBounds(10, 10, 100, 20);
contentPane.add(labelEingabe);

labelSchluessel = new JLabel("Schluessel");
labelSchluessel.setBounds(10, 80, 100, 20);
contentPane.add(labelSchluessel);

labelAusgabe = new JLabel("Ausgabe");
labelAusgabe.setBounds(170, 10, 100, 20);
contentPane.add(labelAusgabe);

// Textfelder
editSchluessel = new JTextField("a");
editSchluessel.setBounds(10, 110, 100, 20);
contentPane.add(editSchluessel);

// Buttons
codiereButton = new JButton ("codiere");
codiereButton.setBounds(10, 40, 100, 20);
contentPane.add(codiereButton);

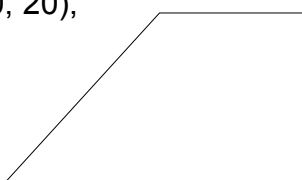
decodiereButton = new JButton ("decodiere");
decodiereButton.setBounds(10, 160, 100, 20);
contentPane.add(decodiereButton);

endeButton = new JButton ("ENDE");
endeButton.setBounds(10, 200, 100, 20);
contentPane.add(endeButton);

// Textarea
ausgabe = new JTextArea();
ausgabe.setText(null);
JScrollPane scroll = new JScrollPane(ausgabe);
scroll.setBounds(160, 40, 200, 200);
contentPane.add(scroll);

// Ende Komponenten

setResizable(false);
setVisible(true);
```



Ausgabe mit
Text Area
und
Scrollbalken

```
// Anfang Ereignisprozeduren
endeButton.addActionListener(new ActionListener() { // reagiert auf MAUSCLICK ENDE
    public void actionPerformed(ActionEvent e) { beendenAction(); }
});
codiereButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) { codierenAction(); }
});
decodiereButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) { deCodierenAction(); }
});

// Ende Ereignisprozeduren

}

/**
 * Definition der Menüleiste
 */
private JMenuBar machMenuLeiste() {
    // enthaltene Objekte definieren
    JMenuBar menuLeiste = new JMenuBar();

    JMenu menuDatei = new JMenu("Datei");
    JMenuItem menuEintragOeffnen = new JMenuItem("Öffnen");
    JMenuItem menuEintragSpeichern = new JMenuItem("Speichern");
    JMenuItem menuEintragEnde = new JMenuItem("Ende");

    JMenu menuBearbeiten = new JMenu("Bearbeiten");
    JMenuItem menuEintragCodiere = new JMenuItem("Codiere");
    JMenuItem menuEintragDecodieren = new JMenuItem("Decodiere");
    JMenuItem menuEintragHaeufigkeiten = new JMenuItem("Häufigkeiten");
    JMenuItem menuEintragSichereHaeufigkeiten = new JMenuItem("sichere in Datei");
    JMenuItem menuEintragPositionen = new JMenuItem("an Positionen");

    // Listener definieren
    menuEintragOeffnen.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) { OeffnenAction(); }
    });
    menuEintragSpeichern.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) { SpeichernAction(); }
    });
    menuEintragEnde.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) { beendenAction(); }
    });
    menuEintragCodiere.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) { codierenAction(); }
    });
    menuEintragDecodieren.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) { deCodierenAction(); }
    });
}
```

s.o.

(s.o.) Menu
in eigene
Methode
ausgliedern

```
});  
menueEintragHaeufigkeiten.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) { haeufigkeitenAction(); }  
});  
menueEintragSichereHaeufigkeiten.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) { sichereHaeufigkeitenAction(); }  
});  
menueEintragPositionen.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) { positionenAction(); }  
});  
  
// hinzufügen  
menueDatei.add(menueEintragOeffnen);  
menueDatei.add(menueEintragSpeichern);  
menueDatei.add(menueEintragEnde);  
  
menueBearbeiten.add(menueEintragCodieren);  
menueBearbeiten.add(menueEintragDecodieren);  
menueBearbeiten.add(menueEintragHaeufigkeiten);  
menueBearbeiten.add(menueEintragSichereHaeufigkeiten);  
menueBearbeiten.add(menueEintragPositionen);  
  
menueLeiste.add(menueDatei);  
menueLeiste.add(menueBearbeiten);  
  
return menueLeiste;  
}
```

Menu -
Methode ist
hier zuende

```
// Actions
private void OeffnenAction(){
    JFileChooser fc = new JFileChooser("E:\\JAVA\\JAVA-
Kryptologie\\Projekte\\Vigenere");
    int returnVal = fc.showOpenDialog(this);
    if (returnVal == JFileChooser.APPROVE_OPTION) {
        text.waehleDateiNamen(fc.getCurrentDirectory()+"\\"+fc.getSelectedFile().getNam
e());
    }
    if (JOptionPane.showConfirmDialog(
        this, "Text ersetzen?", "bestätigen !" ,
        JOptionPane.OK_CANCEL_OPTION)==JOptionPane.OK_OPTION){
        text.dateiLesen();
        ausgabe.setText(text.gibText());
    }
}

private void SpeichernAction(){
    JFileChooser fc = new JFileChooser("E:\\JAVA\\JAVA-
Kryptologie\\Projekte\\Vigenere");
    int returnVal = fc.showSaveDialog(this);
    if (returnVal == JFileChooser.APPROVE_OPTION) {
        text.waehleDateiNamen(fc.getCurrentDirectory()+"\\"+fc.getSelectedFile().getNam
e());
    }
    if (JOptionPane.showConfirmDialog(
        this, "Text ersetzen?", "bestätigen !" ,
        JOptionPane.OK_CANCEL_OPTION)==JOptionPane.OK_OPTION){
        text.neuerText(ausgabe.getText());
        text.dateiSchreiben();
    }
}

private void beendenAction(){
    if (JOptionPane.showConfirmDialog(this, "wirklich beenden?", "prüfen !" ,
        JOptionPane.OK_CANCEL_OPTION)==JOptionPane.OK_OPTION)
        System.exit(0);
}

private void keySetzen(){
    String key = editSchluessel.getText();
    StringBuffer temp=new StringBuffer();
    for (int i=0;i<key.length();i++)
        if (vigenere.istGrossBuchstabe(Character.toUpperCase(key.charAt(i))))
            temp.append(Character.toUpperCase(key.charAt(i)));
        if (temp.toString().equals("")) editSchluessel.setText("Buchstaben!");
        else vigenere.setzeSchluessel(temp.toString());
}
```

in JAVA:
doppelter \
notwendig!

fehlerhafte
Aktionen
abfangen

fehlerhafte
Aktionen
abfangen

fehlerhafte
Aktionen
abfangen

fehlerhafte
Aktionen
abfangen

fehlerhafte
Aktionen
abfangen

fehlerhafte
Aktionen
abfangen

```
private void codierenAction(){
    keySetzen();
    vigenere.setzeOriginalText(ausgabe.getText());
    vigenere.verschluessele();
    labelAusgabe.setText("codiert:");
    ausgabe.setText(vigenere.zeigeText());
}
private void deCodierenAction(){
    keySetzen();
    vigenere.setzeCodetext(ausgabe.getText());
    vigenere.entschluessele();
    labelAusgabe.setText("DECODIERT:");
    ausgabe.setText(vigenere.zeigeText());
}
private void haeufigkeitenAction(){
    new AusgabeHaeufigkeiten(this,
        "Häufigkeiten",
        false,
        vigenere.gibHaeufigkeiten(),
        vigenere.gibKappa(),
        vigenere.gibLaenge());
}
private void sichereHaeufigkeitenAction(){
    AusgabeHaeufigkeiten temp=
    new AusgabeHaeufigkeiten(this,
        "Häufigkeiten",
        false,
        vigenere.gibHaeufigkeiten(),
        vigenere.gibKappa(),
        vigenere.gibLaenge());
    temp.schreibeInTextDatei();
}
private void positionenAction(){
    int laenge=editSchluessel.getText().length();
    for (int i=0;i<laenge;i++)
        new AusgabeHaeufigkeiten(this,
            "Häufigkeit an Positionen",
            false,
            vigenere.gibHaeufigkeiten(i,laenge),
            i);
}

public static void main(String[] args) {
    vigenere = new Vigenere();
    new Gui("Gui", 400, 330);
}
}
```

allgemein
schlecht:
Kommentare
fehlen!