|  |  |  |
| --- | --- | --- |
| Übung Nr.: 3  Jahrgang: 5BHME  Gruppe: 3B  Betreuer: SN |  | Protokollabgabe:  Solldatum: 11.11.2024  Ist-Datum:  Note:  Note Deutsch: |
|  |  |  |
| Protokoll | | |
| **THEMA:** | | |
| Automatisierte Frequenzkennlinienaufnahme | | |

|  |  |
| --- | --- |
| Tag: | * | 2. Laborgruppendurchgang |
| Zeit: | 10:30 bis 13:15 |
| Ort: | HTBLA Kaindorf, PRR-Labor |
| Anwesend: | Weiß Lukas, Wang Bowen |
| Schriftführer: | Weiß Lukas |

Aufgabenstellung

Es soll ein Java Programm erstellt werden, welches den Amplitudengang eines Vierpols

aufnehmen kann. Dazu soll eine GUI erstellt werden, mit welcher ein Frequenzumrichter angesteuert wird (mithilfe WG810) und das Spannungssignal per Multimeter (Fluke45) wieder eingelesen und ausgegeben wird.

Resümee

Aufgrund von Zeitmangel konnten wir noch nicht in der GUI die Daten einlesen. Neben dem bisherigen Fortschritt haben wir heute die einfache GUI abgeschlossen und die Funktionen "Anpassen", "Löschen" und "Speichern" programmiert. Während dieser Übung konnten wir unser Know-how im Bereich der GUI-Erstellung erneut vertiefen.

|  |  |  |
| --- | --- | --- |
| Weiß Lukas | Wang Bowen | Datum  11.11.2024 |

Inhaltsverzeichnis

Zeitplan der Laboreinheit 3

Zeitplan 3

Aufgabe B) 3

Programmierung 3

Data - Kurvenformen 3

Data – Messwert 4

Data – Messwerte 5

GUI - Amplitudendiagramm 7

GUI - FlukeListModel 13

Klassen: Fluke, SimpleSerial, WG810 14

Verwendete Betriebsmittel 14

Programmierung 14

Daten - Kurvenformen 14

Daten – Messwert 14

Daten – Messwerte 15

GUI 16

FlukeListModel 22

Klassen: Fluke, SimpleSerial, WG810 22

Verwendete Betriebsmittel 22

Aufgabe a) 24

Java-Dateien einfügen 24

Fluke45 V2019 Änderung 24

WG810 V2019 Änderung 24

Verwendete Betriebsmittel 24

Anhang 24

# Zeitplan der Laboreinheit

## Zeitplan

* 10:30 – 11:30 🡺 Besprechung der Aufgabe
* 11:30 – 13:15 🡺 Selbständiges Arbeiten

# Ein Bild, das Text, Screenshot, Display, Software enthält. Automatisch generierte BeschreibungAufgabe B)

Vorschlag für Gestaltung der GUI

# Programmierung

## Data - Kurvenformen

package ue01.data;

/\*\*

\*

\* @author

\*/

public enum Kurvenform

{

DREIECK,RECHTECK,SINUS;

private final static String[] text =

{

"Dreieck", "Rechteck", "Sinus"

};

@Override

public String toString()

{

return text[ordinal()];

}

}

## Data – Messwert

package ue01.data;

import java.io.\*;

public class Messwert

{

public static int size()

{

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

// Datenelemente

private final Kurvenform kurvenform;

private final double frequenz;

private final double amplitudeein;

private final double amplitudeaus;

public Messwert(Kurvenform kurvenform, double frequenz, double amplitudeein, double amplitudeaus)

throws Exception

{

this.kurvenform = kurvenform;

this.frequenz = frequenz;

this.amplitudeein = amplitudeein;

this.amplitudeaus = amplitudeaus;

}

public Kurvenform getKurvenform()

{

return kurvenform;

}

public double getFrequenz()

{

return frequenz;

}

public double getAmplitudeein()

{

return amplitudeein;

}

public double getAmplitudeaus()

{

return amplitudeaus;

}

@Override

public String toString()

{

return String.format("%s: %.3f Hz: %.3f V: %.3f V",kurvenform, frequenz, amplitudeein, amplitudeaus);

}

public void writeTo(BufferedWriter writer)

throws Exception

{

writer.write(String.format("%s: %.3f Hz: %.3f V: %.3f V",

kurvenform, frequenz, amplitudeein, amplitudeaus));

}

}

## Data – Messwerte

package ue01.data;

import java.io.\*;

import java.util.\*;

/\*\*

\*

\* @author

\*/

public class Messwerte

{

private final List<Messwert> messwerte = new ArrayList<>();

{

try

{

add(new Messwert(Kurvenform.SINUS, 100.00, 10.000,9.000));

}

catch (Exception ex) { }

}

public List<Messwert> getMesswerte()

{

return messwerte;

}

public int size()

{

return messwerte.size();

}

public boolean add(Messwert e)

{

return messwerte.add(e);

}

public Messwert get(int index)

{

return messwerte.get(index);

}

public Messwert set(int index, Messwert element)

{

return messwerte.set(index, element);

}

public Messwert remove(int index)

{

return messwerte.remove(index);

}

public void writeTo(BufferedWriter writer)

throws Exception

{

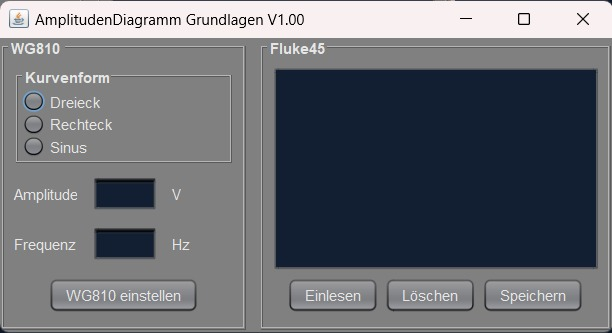
for (Messwert messwert : messwerte)

messwert.writeTo(writer);

}

}

## GUI - Amplitudendiagramm



package ue01.gui;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileOutputStream;

import java.io.OutputStreamWriter;

import java.util.ArrayList;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import ue01.WG810V2019;

import ue01.data.Messwert;

import ue01.data.Messwerte;

/\*\*

\*

\* @author

\*/

public class AmplitudenDiagrammGUI extends javax.swing.JFrame

{

private final Messwerte messwerte = new Messwerte();

private final FlukeListModel model = new FlukeListModel(messwerte.getMesswerte());

/\*\*

\* Creates new form AmplitudenDiagramm

\*/

public AmplitudenDiagrammGUI()

{

initComponents();

listenfeld.setModel(model);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents()

{

java.awt.GridBagConstraints gridBagConstraints;

bgKurvenform = new javax.swing.ButtonGroup();

pMain = new javax.swing.JPanel();

pEast = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

listenfeld = new javax.swing.JList<>();

jPanel1 = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

pCenter = new javax.swing.JPanel();

pKurvenform = new javax.swing.JPanel();

rbDreieck = new javax.swing.JRadioButton();

rbRechteck = new javax.swing.JRadioButton();

rbSinus = new javax.swing.JRadioButton();

pEingabe = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

tfFrequenz = new javax.swing.JFormattedTextField();

tfAmplitude = new javax.swing.JFormattedTextField();

pButton = new javax.swing.JPanel();

btEinstellen = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DO\_NOTHING\_ON\_CLOSE);

setTitle("AmplitudenDiagramm Grundlagen V1.00");

setLocationByPlatform(true);

addWindowListener(new java.awt.event.WindowAdapter()

{

public void windowClosing(java.awt.event.WindowEvent evt)

{

onX(evt);

}

});

getContentPane().setLayout(new java.awt.BorderLayout(8, 8));

pMain.setLayout(new java.awt.BorderLayout(8, 8));

pEast.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(javax.swing.border.EtchedBorder.RAISED), "Fluke45", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.TOP), javax.swing.BorderFactory.createEmptyBorder(5, 5, 5, 5)));

pEast.setLayout(new java.awt.BorderLayout());

jScrollPane1.setViewportView(listenfeld);

pEast.add(jScrollPane1, java.awt.BorderLayout.CENTER);

jButton1.setText("Einlesen");

jButton1.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt)

{

onEinlesen(evt);

}

});

jPanel1.add(jButton1);

jButton2.setText("Löschen");

jButton2.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt)

{

onLoeschen(evt);

}

});

jPanel1.add(jButton2);

jButton3.setText("Speichern");

jButton3.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt)

{

onSpeichern(evt);

}

});

jPanel1.add(jButton3);

pEast.add(jPanel1, java.awt.BorderLayout.SOUTH);

pMain.add(pEast, java.awt.BorderLayout.EAST);

pCenter.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(javax.swing.border.EtchedBorder.RAISED), "WG810", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.TOP), javax.swing.BorderFactory.createEmptyBorder(5, 5, 5, 5)));

pCenter.setLayout(new java.awt.BorderLayout(8, 8));

pKurvenform.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(javax.swing.border.EtchedBorder.RAISED), "Kurvenform", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.TOP), javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1)));

pKurvenform.setLayout(new java.awt.GridLayout(0, 1));

bgKurvenform.add(rbDreieck);

rbDreieck.setText("Dreieck");

pKurvenform.add(rbDreieck);

bgKurvenform.add(rbRechteck);

rbRechteck.setText("Rechteck");

pKurvenform.add(rbRechteck);

bgKurvenform.add(rbSinus);

rbSinus.setText("Sinus");

pKurvenform.add(rbSinus);

pCenter.add(pKurvenform, java.awt.BorderLayout.NORTH);

java.awt.GridBagLayout pEingabeLayout = new java.awt.GridBagLayout();

pEingabeLayout.columnWidths = new int[] {0, 12, 0, 12, 0, 12, 0, 12, 0, 12, 0};

pEingabeLayout.rowHeights = new int[] {0, 12, 0};

pEingabe.setLayout(pEingabeLayout);

jLabel3.setText("Amplitude");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 0;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(jLabel3, gridBagConstraints);

jLabel4.setText("Frequenz");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 2;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(jLabel4, gridBagConstraints);

jLabel1.setText("Hz");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 4;

gridBagConstraints.gridy = 2;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(jLabel1, gridBagConstraints);

jLabel2.setText("V");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 4;

gridBagConstraints.gridy = 0;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(jLabel2, gridBagConstraints);

tfFrequenz.setColumns(4);

tfFrequenz.setValue(1000.0);

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 2;

gridBagConstraints.gridy = 2;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(tfFrequenz, gridBagConstraints);

tfAmplitude.setColumns(4);

tfAmplitude.setValue(10.0);

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 2;

gridBagConstraints.gridy = 0;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(tfAmplitude, gridBagConstraints);

pCenter.add(pEingabe, java.awt.BorderLayout.CENTER);

btEinstellen.setText("WG810 einstellen");

btEinstellen.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt)

{

onEinstellen(evt);

}

});

pButton.add(btEinstellen);

pCenter.add(pButton, java.awt.BorderLayout.SOUTH);

pMain.add(pCenter, java.awt.BorderLayout.CENTER);

getContentPane().add(pMain, java.awt.BorderLayout.CENTER);

pack();

}// </editor-fold>

private void onLoeschen(java.awt.event.ActionEvent evt)

{

listenfeld.setModel(new FlukeListModel(new ArrayList<>()));

tfAmplitude.setValue(0);

tfFrequenz.setValue(0);

}

private void onEinlesen(java.awt.event.ActionEvent evt)

{

//TODO

}

private void onSpeichern(java.awt.event.ActionEvent evt)

{

try

{

final JFileChooser chooser = new JFileChooser();

if (chooser.showSaveDialog(this) == JFileChooser.APPROVE\_OPTION);

{

final File file = chooser.getSelectedFile();

try (

final BufferedWriter writer =

new BufferedWriter(

new OutputStreamWriter(

new FileOutputStream(file), "utf8")))

{

messwerte.writeTo(writer);

}

}

}

catch (Exception ex)

{

}

}

private void onEinstellen(java.awt.event.ActionEvent evt)

{

try (WG810V2019 wg810 = new WG810V2019("COM6"))

{

wg810.setRemote(true);

System.out.println("Remote: "+wg810.getRemote());

if(rbSinus.isSelected()) wg810.setKurvenform(WG810V2019.FORMSIN, 0, true);

else if(rbDreieck.isSelected()) wg810.setKurvenform(WG810V2019.FORMTRI, 0, true);

else if(rbRechteck.isSelected()) wg810.setKurvenform(WG810V2019.FORMRECT, 0, true);

wg810.setFrequenz(((Number)tfFrequenz.getValue()).doubleValue(),true);

wg810.setAmplitude(((Number)tfAmplitude.getValue()).doubleValue(),true);

wg810.setRemote(false);

System.out.println("Remote: "+wg810.getRemote());

}

catch (Exception ex)

{

ex.printStackTrace();

}

}

private void onX(java.awt.event.WindowEvent evt)

{

if(JOptionPane.showConfirmDialog(

this, "Programm wirklich beenden?", "Sicherheitsabfrage",

JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION)

dispose();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[])

{

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try

{

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())

{

if ("Nimbus".equals(info.getName()))

{

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

}

catch (ClassNotFoundException ex)

{

java.util.logging.Logger.getLogger(AmplitudenDiagrammGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

catch (InstantiationException ex)

{

java.util.logging.Logger.getLogger(AmplitudenDiagrammGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

catch (IllegalAccessException ex)

{

java.util.logging.Logger.getLogger(AmplitudenDiagrammGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

catch (javax.swing.UnsupportedLookAndFeelException ex)

{

java.util.logging.Logger.getLogger(AmplitudenDiagrammGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable()

{

public void run()

{

new AmplitudenDiagrammGUI().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.ButtonGroup bgKurvenform;

private javax.swing.JButton btEinstellen;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JList<Messwert> listenfeld;

private javax.swing.JPanel pButton;

private javax.swing.JPanel pCenter;

private javax.swing.JPanel pEast;

private javax.swing.JPanel pEingabe;

private javax.swing.JPanel pKurvenform;

private javax.swing.JPanel pMain;

private javax.swing.JRadioButton rbDreieck;

private javax.swing.JRadioButton rbRechteck;

private javax.swing.JRadioButton rbSinus;

private javax.swing.JFormattedTextField tfAmplitude;

private javax.swing.JFormattedTextField tfFrequenz;

// End of variables declaration

}

## GUI - FlukeListModel

package ue01.gui;

import java.util.List;

import javax.swing.AbstractListModel;

import ue01.data.Messwert;

public class FlukeListModel extends AbstractListModel<Messwert>

{

private final List<Messwert> messwerte;

public FlukeListModel(java.util.List<ue01.data.Messwert> messwerte)

{

this.messwerte = messwerte;

}

@Override

public int getSize()

{

return messwerte.size();

}

@Override

public Messwert getElementAt(int index)

{

return messwerte.get(index);

}

}

## Klassen: Fluke, SimpleSerial, WG810

Siehe: <https://elearn3.htl-kaindorf.at/mod/folder/view.php?id=1477>

# Verwendete Betriebsmittel

* Messgerät Fluke45 40-11/1995/1/1
* Frequenzgenerator WG-820 420-3/2010/1/4
* Digitus - Serieller Adapter
* PC – NetBeans IDE 21

# Programmierung

## Daten - Kurvenformen

package ue01.data;

/\*\*

\*

\* @author

\*/

public enum Kurvenform

{

DREIECK,RECHTECK,SINUS;

private final static String[] text =

{

"Dreieck", "Rechteck", "Sinus"

};

@Override

public String toString()

{

return text[ordinal()];

}

}

## Daten – Messwert

package ue01.data;

public class Messwert

{

public static int size()

{

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

// Datenelemente

private final Kurvenform kurvenform;

private final double frequenz;

private final double amplitudeein;

private final double amplitudeaus;

public Messwert(Kurvenform kurvenform, double frequenz, double amplitudeein, double amplitudeaus)

throws Exception

{

this.kurvenform = kurvenform;

this.frequenz = frequenz;

this.amplitudeein = amplitudeein;

this.amplitudeaus = amplitudeaus;

}

public Kurvenform getKurvenform()

{

return kurvenform;

}

public double getFrequenz()

{

return frequenz;

}

public double getAmplitudeein()

{

return amplitudeein;

}

public double getAmplitudeaus()

{

return amplitudeaus;

}

## Daten – Messwerte

package ue01.data;

import java.time.LocalDate;

import java.time.Month;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author

\*/

public class Messwerte

{

private final List<Messwert> messwerte = new ArrayList<>();

{

try

{

add(new Messwert(Kurvenform.SINUS, 100.00, 10.000,9));

}

catch (Exception ex) { }

}

public int size()

{

return messwerte.size();

}

public boolean add(Messwert e)

{

return messwerte.add(e);

}

public Messwert get(int index)

{

return messwerte.get(index);

}

public Messwert set(int index, Messwert element)

{

return messwerte.set(index, element);

}

public Messwert remove(int index)

{

return messwerte.remove(index);

}

}

## GUI

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

package ue01.gui;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author

\*/

public class AmplitudenDiagramm extends javax.swing.JFrame

{

/\*\*

\* Creates new form AmplitudenDiagramm

\*/

public AmplitudenDiagramm()

{

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents()

{

java.awt.GridBagConstraints gridBagConstraints;

bgKurvenform = new javax.swing.ButtonGroup();

pMain = new javax.swing.JPanel();

pEast = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

listenfeld = new javax.swing.JList<>();

jPanel1 = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

pCenter = new javax.swing.JPanel();

pKurvenform = new javax.swing.JPanel();

rbDreieck = new javax.swing.JRadioButton();

rbRechteck = new javax.swing.JRadioButton();

rbSinus = new javax.swing.JRadioButton();

pEingabe = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

tfFrequenz = new javax.swing.JFormattedTextField();

tfAmplitude = new javax.swing.JFormattedTextField();

pButton = new javax.swing.JPanel();

btEinstellen = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DO\_NOTHING\_ON\_CLOSE);

setTitle("AmplitudenDiagramm Grundlagen V1.00");

setLocationByPlatform(true);

addWindowListener(new java.awt.event.WindowAdapter()

{

public void windowClosing(java.awt.event.WindowEvent evt)

{

onX(evt);

}

});

getContentPane().setLayout(new java.awt.BorderLayout(8, 8));

pMain.setLayout(new java.awt.BorderLayout(8, 8));

pEast.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(javax.swing.border.EtchedBorder.RAISED), "Fluke45", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.TOP), javax.swing.BorderFactory.createEmptyBorder(5, 5, 5, 5)));

pEast.setLayout(new java.awt.BorderLayout());

jScrollPane1.setViewportView(listenfeld);

pEast.add(jScrollPane1, java.awt.BorderLayout.CENTER);

jButton1.setText("Einlesen");

jButton1.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt)

{

onEinlesen(evt);

}

});

jPanel1.add(jButton1);

jButton2.setText("Löschen");

jButton2.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt)

{

onLoeschen(evt);

}

});

jPanel1.add(jButton2);

jButton3.setText("Speichern");

jButton3.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt)

{

onSpeichern(evt);

}

});

jPanel1.add(jButton3);

pEast.add(jPanel1, java.awt.BorderLayout.SOUTH);

pMain.add(pEast, java.awt.BorderLayout.EAST);

pCenter.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(javax.swing.border.EtchedBorder.RAISED), "WG810", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.TOP), javax.swing.BorderFactory.createEmptyBorder(5, 5, 5, 5)));

pCenter.setLayout(new java.awt.BorderLayout(8, 8));

pKurvenform.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(javax.swing.border.EtchedBorder.RAISED), "Kurvenform", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.TOP), javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1)));

pKurvenform.setLayout(new java.awt.GridLayout(0, 1));

bgKurvenform.add(rbDreieck);

rbDreieck.setText("Dreieck");

pKurvenform.add(rbDreieck);

bgKurvenform.add(rbRechteck);

rbRechteck.setText("Rechteck");

pKurvenform.add(rbRechteck);

bgKurvenform.add(rbSinus);

rbSinus.setText("Sinus");

pKurvenform.add(rbSinus);

pCenter.add(pKurvenform, java.awt.BorderLayout.NORTH);

java.awt.GridBagLayout pEingabeLayout = new java.awt.GridBagLayout();

pEingabeLayout.columnWidths = new int[] {0, 12, 0, 12, 0, 12, 0, 12, 0, 12, 0};

pEingabeLayout.rowHeights = new int[] {0, 12, 0};

pEingabe.setLayout(pEingabeLayout);

jLabel3.setText("Amplitude");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 0;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(jLabel3, gridBagConstraints);

jLabel4.setText("Frequenz");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 2;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(jLabel4, gridBagConstraints);

jLabel1.setText("Hz");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 4;

gridBagConstraints.gridy = 2;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(jLabel1, gridBagConstraints);

jLabel2.setText("V");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 4;

gridBagConstraints.gridy = 0;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(jLabel2, gridBagConstraints);

tfFrequenz.setColumns(4);

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 2;

gridBagConstraints.gridy = 2;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(tfFrequenz, gridBagConstraints);

tfAmplitude.setColumns(4);

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 2;

gridBagConstraints.gridy = 0;

gridBagConstraints.anchor = java.awt.GridBagConstraints.LINE\_START;

pEingabe.add(tfAmplitude, gridBagConstraints);

pCenter.add(pEingabe, java.awt.BorderLayout.CENTER);

btEinstellen.setText("WG810 einstellen");

btEinstellen.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt)

{

onEinstellen(evt);

}

});

pButton.add(btEinstellen);

pCenter.add(pButton, java.awt.BorderLayout.SOUTH);

pMain.add(pCenter, java.awt.BorderLayout.CENTER);

getContentPane().add(pMain, java.awt.BorderLayout.CENTER);

pack();

}// </editor-fold>//GEN-END:initComponents

private void onLoeschen(java.awt.event.ActionEvent evt)//GEN-FIRST:event\_onLoeschen

{//GEN-HEADEREND:event\_onLoeschen

}//GEN-LAST:event\_onLoeschen

private void onEinlesen(java.awt.event.ActionEvent evt)//GEN-FIRST:event\_onEinlesen

{//GEN-HEADEREND:event\_onEinlesen

// TODO add your handling code here:

}//GEN-LAST:event\_onEinlesen

private void onSpeichern(java.awt.event.ActionEvent evt)//GEN-FIRST:event\_onSpeichern

{//GEN-HEADEREND:event\_onSpeichern

// TODO add your handling code here:

}//GEN-LAST:event\_onSpeichern

private void onEinstellen(java.awt.event.ActionEvent evt)//GEN-FIRST:event\_onEinstellen

{//GEN-HEADEREND:event\_onEinstellen

// TODO add your handling code here:

}//GEN-LAST:event\_onEinstellen

private void onX(java.awt.event.WindowEvent evt)//GEN-FIRST:event\_onX

{//GEN-HEADEREND:event\_onX

if(JOptionPane.showConfirmDialog(

this, "Programm wirklich beenden?", "Sicherheitsabfrage",

JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION)

dispose();

}//GEN-LAST:event\_onX

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[])

{

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try

{

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())

{

if ("Nimbus".equals(info.getName()))

{

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

}

catch (ClassNotFoundException ex)

{

java.util.logging.Logger.getLogger(AmplitudenDiagramm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

catch (InstantiationException ex)

{

java.util.logging.Logger.getLogger(AmplitudenDiagramm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

catch (IllegalAccessException ex)

{

java.util.logging.Logger.getLogger(AmplitudenDiagramm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

catch (javax.swing.UnsupportedLookAndFeelException ex)

{

java.util.logging.Logger.getLogger(AmplitudenDiagramm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable()

{

public void run()

{

new AmplitudenDiagramm().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.ButtonGroup bgKurvenform;

private javax.swing.JButton btEinstellen;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JList<String> listenfeld;

private javax.swing.JPanel pButton;

private javax.swing.JPanel pCenter;

private javax.swing.JPanel pEast;

private javax.swing.JPanel pEingabe;

private javax.swing.JPanel pKurvenform;

private javax.swing.JPanel pMain;

private javax.swing.JRadioButton rbDreieck;

private javax.swing.JRadioButton rbRechteck;

private javax.swing.JRadioButton rbSinus;

private javax.swing.JFormattedTextField tfAmplitude;

private javax.swing.JFormattedTextField tfFrequenz;

// End of variables declaration//GEN-END:variables

}

## FlukeListModel

package ue01.gui;

import java.util.List;

import javax.swing.AbstractListModel;

import ue01.data.Messwert;

public class FlukeListModel extends AbstractListModel<Messwert>

{

private final List<Messwert> messwerte;

public FlukeListModel(java.util.List<ue01.data.Messwert> messwerte)

{

this.messwerte = messwerte;

}

@Override

public int getSize()

{

return messwerte.size();

}

@Override

public Messwert getElementAt(int index)

{

return messwerte.get(index);

}

}

## Klassen: Fluke, SimpleSerial, WG810

Siehe: <https://elearn3.htl-kaindorf.at/mod/folder/view.php?id=1477>

# Verwendete Betriebsmittel

* Messgerät Fluke45 40-11/1995/1/1
* Frequenzgenerator WG-820 420-3/2010/1/4
* Digitus - Serieller Adapter
* PC - NetBeans

# Aufgabe a)

## Java-Dateien einfügen

Als erstes mussten wir die Java-Dateien *„Fluke45V2019.java“,* *„SimpleSerialV2019.java“* und *„WG810V2019.java“* vom E-Learn3 Serverherunterladen und in das Projekt einbinden. Anschließend mussten wir die *„jSerialComm-2.7.0.jar“* Datei als Libary noch hinzufügen und alle package Namen ändern.

Source packages 🡺 properties 🡺 libaries 🡺 unter classpath die zu benutzende Datei auswählen

## Fluke45 V2019 Änderung

In Zeile 58 mussten wir das COM auf die richtige Schnittstelle anpassen, welches in unserem Falle „COM4“ war

## WG810 V2019 Änderung

In Zeile 961 im WG810 Programmcode mussten wir das COM in unserem Falle auf „COM3“ setzen

## Verwendete Betriebsmittel

* Messgerät Fluke45 40-11/1995/1/1
* Frequenzgenerator WG-820 420-3/2010/1/4
* Digitus - Serieller Adapter

## Anhang

* Siehe Programmcode:

<https://elearn3.htl-kaindorf.at/mod/folder/view.php?id=1477>