```
/**************
 * File: SwingApplet.java
 * @autor
 */
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.applet.*;
import javax.swing.*;
import javax.swing.filechooser.*;
import java.lang.*;
import java.lang.Runtime;
import java.util.*;
import java.io.*;
import java.io.InputStream.*;
import java.io.OutputStream;
import java.lang.Math.*;

public class ABSApplet extends JApplet
{
      public static ARABS ModelAR_ABS;
      public static MyLp Solver;
      public static Prediction Pred;
      public static ReadData1 Red;
      public static Diagnosis Dig;
      public static OutputFrame out;

      String Name1 = "Model", Name2 = "Parameters",Name3 = "Calculation";
      String IFile = new String();
      String Model = new String();
      String DefaultFile = new String("test");
      String Dir= new String();
      String F = new String();
      String line="";
      int width = 150, height = 28, Left;
      int Iterations;
      int ParamWidth, ParamHeight, fc=0;
      boolean Bcompare= false;
      boolean Task = true;                      /// means to perform prognosis.
      public static int FileIdent;
      ReadDataFile  RDF;
      PerformCount  PC;
      PerformDiagnosis PD;
      JProgressBar      wProgressBar = new JProgressBar();
      public static JTextArea t = new JTextArea(14,40);
      public static JTextArea t1 = new JTextArea(8,40);
      public static JTextArea t2 = new JTextArea(8,40);
      JButton w11Button, w12Button,
                w1Button, w2Button, w3Button, w4Button;
      ButtonGroup g = new ButtonGroup();
      JRadioButton
                  rb1 = new JRadioButton(" Use data file", true),
                  rb2 = new JRadioButton(" Enter data in a field", false);
      JLabel  wLabel,
                  w11Label,  w13Label,
                w20Label, w200Label, w21Label, w22Label,  w2nLabel[] = new
JLabel[ModelAR_ABS.InputsNumber],
                 w31Label;
                  Choice w12Label;
```

```
      JTextField w11TextField,
                 w2nTextField[] = new JTextField[ModelAR_ABS.InputsNumber];
      JComboBox w11ComboBox;
      JTabbedPane wTabbedPane = new JTabbedPane();
      JPanel wPanel, w1Panel, w2Panel, w3Panel;
      JScrollPane wScrollPane, w1ScrollPane, w2ScrollPane;
      JPopupMenu wPopupMenu1, wPopupMenut1, wPopupMenu2, wPopupMenu3;
      JMenuItem wMenuItem1,wMenuItem2,wMenuItem3,
wMenuItemt1,wMenuItemt2,wMenuItemt3, wMenuItemt4;
      Color wColor = new Color(222, 222, 222);
      Vector wItVector = new Vector(0), wFVector = new Vector(0);
      JFrame  wGFrame;
//-----------------Choise
listener-------------------------------------------------------------------
      class IL1 implements ItemListener
{
      String Tasktemp;
    private void update(boolean what)
      {
            Task= what;
            for (int i=3; i<=6; i++) {
                  w2nTextField[i].setEnabled(what);
                  w2nTextField[i].setVisible(what);
                  w2nLabel[i].setVisible(what);
            }
            w2nLabel[1].setVisible(what);
      }
      public void itemStateChanged(ItemEvent e)
      {
            Object object= e.getSource();
            ItemSelectable ie = e.getItemSelectable();
            Tasktemp =  ie.getSelectedObjects()[0].toString();

            if   (Tasktemp=="Prognosis")
            {
                  update(true);
                  w2ScrollPane.setVisible(false);
                  w20Label.setVisible(false);
                  w200Label.setVisible(false);
                  w2nTextField[1].setText(""+5);
                  ARABS.class1= 5;
                  w2nTextField[0].setText(""+1);
                  ARABS.class0= 1;
                  w2nTextField[2].setText(""+40);
                  ARABS.class2= 40;
                  w2nTextField[1].removeFocusListener( wTSFocusListenerD);
                  w2nTextField[1].addFocusListener(wTSFocusListener);
            }
             else
             {
                   w2nTextField[1].removeFocusListener( wTSFocusListener);
                   w2nTextField[1].addFocusListener(wTSFocusListenerD);
                   w2nTextField[1].setText(""+0.0);
                   ARABS.class1= 0;
                   w2nTextField[0].setText(""+14);
                   ARABS.class0= 14;
                   w2nTextField[2].setText(""+24);
                   ARABS.class2= 24;
//System.out.println(" "+ARABS.class2);
```

```
                  update(false);
                  w2ScrollPane.setVisible(true);
                  w200Label.setVisible(true);
                  w20Label.setVisible(true);
            }
      }
}

//--------------- MouseListener ------------------Popup Menu
adapter-------------------------------
      class CPMouseAdapter extends MouseAdapter
      {
            public void mouseReleased(MouseEvent e)
            {
                  if (e.isPopupTrigger())
                  {
                        if(e.getSource()==w1Panel)
                              wPopupMenu1.show(e.getComponent(), e.getX(),
e.getY());
                        else if     (e.getSource()==w2Panel)
                              wPopupMenu2.show(e.getComponent(), e.getX(),
e.getY());
                        else if     (e.getSource()==t1)
                              wPopupMenut1.show(e.getComponent(), e.getX(),
e.getY());
                        else
                              wPopupMenu3.show(e.getComponent(), e.getX(),
e.getY());
                  }
            }
      }
CPMouseAdapter wPMouseAdapter = new CPMouseAdapter();

//------------------ActionListener---------------Help File
Invoker----------------------------------------
class Popup1 implements ActionListener
{
      ReadDataFile  RDF= new ReadDataFile();
      public void actionPerformed(ActionEvent e)
      {
            String Fi = new String();
            String Fj = new String();
            Object object = e.getSource();
            getAppletContext().showStatus("View Help file");
            if  (object== wMenuItemt4)
                  t1.selectAll();
            else
            {
            if (object== wMenuItem1)
                  Fi= "Help1";
            else if (object== wMenuItem2)
                Fi= "Help2";
            else if  (object== wMenuItem3)
                  Fi= "Help3";
            String path= new String();
            path= "";
            OutputFrame out = new OutputFrame("HELP Window...");
            RDF.SimpleRead(path, Fi,out.text );
            out.setSize( 500, 700 );
```

```
            out.setVisible( true );
            }
      }
}

//---------ActionListener---------------Geting and  controling values from
Imputs---------------------------------------
class CSFocusListener implements FocusListener
{
    public void focusGained(FocusEvent e)  {}
    public void focusLost(FocusEvent e)
      {
            int  aa=0;
          boolean ex=false;
            String Str = ((JTextField)e.getSource()).getText();
            Object object = e.getSource();
            try
            {
                  aa=  Short.valueOf(Str).shortValue();
                  getAppletContext().showStatus("Last Value set to: "+Str);
            }
          catch( NumberFormatException exc)

            {
                   ex= true;
            }
            if ((aa<0)||( ex))
            {
                  ex= true;
                  fc=fc+1; if (fc == 3) fc = 1; if (fc == 1)
                  {
                        JOptionPane.showMessageDialog(wTabbedPane, "Field value
must be a number!", "Warning!", JOptionPane.WARNING_MESSAGE);
                  }
            }
            if (object ==  w2nTextField[0])
            {
                  if(ex)
                      w2nTextField[0].setText(""+ModelAR_ABS.class0);
                  else  if (aa==0)
                  {
                              fc=fc+1; if (fc == 3) fc = 1; if (fc == 1)
                        {
                              JOptionPane.showMessageDialog(wTabbedPane, "Value
must be greater than 0!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    ARABS.class0 =1;
                                    w2nTextField[0].setText(""+ ARABS.class0);
                              }
                  }
                  else
                        ModelAR_ABS.class0 =aa;
            }
             else if (object==  w2nTextField[1] )
             {
                  if(ex)
                        w2nTextField[1].setText(""+ModelAR_ABS.class1);
                  else
                   {
                      ModelAR_ABS.class1 =aa;
```

```
                        if ((aa>=ModelAR_ABS.class2)&&(Task))
                        {
                              fc=fc+1; if (fc == 3) fc = 1; if (fc == 1)
                        {
                              JOptionPane.showMessageDialog(wTabbedPane, "p value
must be less than T!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    ARABS.class2 =ModelAR_ABS.class1 + 1;
                                    w2nTextField[2].setText(""+ ARABS.class2);
                              }
                        }
                  }
             }
              else if (object==  w2nTextField[2] )
              {
                   if(ex)
                        w2nTextField[2].setText(""+ModelAR_ABS.class2);
                   else
                   {
                         ModelAR_ABS.class2 =aa;
                         if ((aa<=ModelAR_ABS.class1)&&(Task))
                        {
                              fc=fc+1; if (fc == 3) fc = 1; if (fc == 1)
                              {
                              JOptionPane.showMessageDialog(wTabbedPane, "p value
must be less than T!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    if (ModelAR_ABS.class2 >0)
                                    {
                                          ARABS.class1 =ModelAR_ABS.class2 - 1;
                                          w2nTextField[1].setText(""+
ARABS.class1);
                                    }
                                    else
                                    {
                                          ARABS.class2 =ModelAR_ABS.class1 + 1;
                                          w2nTextField[2].setText(""+
ARABS.class2);
                                    }
                              }
                         }
                        }
                  }
                   else if (object==  w2nTextField[3] )
                  {
                     if(ex)
                         w2nTextField[3].setText(""+ModelAR_ABS.class3);
                     else if (aa==0)
                     {
                          ModelAR_ABS.class3 =aa;
                        w3Button.setEnabled(false);
                   }
                     else if (aa==1)
                         ModelAR_ABS.class3 =aa;
                     else
                     {
                              fc=fc+1; if (fc == 3) fc = 1; if (fc == 1)
                         {
                              JOptionPane.showMessageDialog(wTabbedPane, "Indicator
value must be 0 or 1!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    ModelAR_ABS.class3 =0;
```

```
                                w2nTextField[3].setText("0");
                            }
                        }
                    }
                    else if (object==  w2nTextField[4] )
                   {
                      if(ex)
                          w2nTextField[4].setText(""+ModelAR_ABS.class4);
                      else if (aa==0)
                      {
                           ModelAR_ABS.class4 =aa;
                          w3Button.setEnabled(false);
                           w2nTextField[5].setEnabled(true);
                           w2nTextField[6].setEnabled(true);
                    }
                      else if (aa==1)
                      {
                          ModelAR_ABS.class4 =aa;
                          w2nTextField[5].setEnabled(false);
                          w2nTextField[5].setEnabled(false);
                          ModelAR_ABS.classK =0;
                          w2nTextField[6].setText("0");
                          w2nTextField[6].setEnabled(false);
                     }
                      else
                      {
                                fc=fc+1; if (fc == 3) fc = 1; if (fc == 1)
                          {
                                JOptionPane.showMessageDialog(wTabbedPane, "Indicator
value must be 0 or 1!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    ModelAR_ABS.class4 =0;
                                    w2nTextField[4].setText("0");
                          }
                    }
               }
               else if (object==  w2nTextField[5] )
                    {
                      if(ex)
                          w2nTextField[5].setText(""+ModelAR_ABS.classM);
                      else if (aa==0)
                      {
                           ModelAR_ABS.classM =aa;
                           w2nTextField[4].setEnabled(true);
                           w3Button.setEnabled(false);
                    }
                      else if (aa==1) {
                          ModelAR_ABS.classM =aa;
                          w2nTextField[4].setEnabled(false);
                    }
                      else
                      {
                                fc=fc+1; if (fc == 3) fc = 1; if (fc == 1)
                          {
                                JOptionPane.showMessageDialog(wTabbedPane, "Indicator
value must be 0 or 1!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    ModelAR_ABS.classM =0;
                                    w2nTextField[5].setText("0");
                          }
                    }
```

```
            }
             else if (object==  w2nTextField[6] )
             {
                  if(ex)
                        w2nTextField[6].setText(""+ModelAR_ABS.classK);
                  else
                   {
                      ModelAR_ABS.classK =aa;
                        if ((aa>100)&&(Task))
                        {
                              fc=fc+1;  if (fc == 3) fc = 1;     if (fc == 1)
                        {
                              JOptionPane.showMessageDialog(wTabbedPane, "It is
recomended k value less than 100!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    ARABS.classK =100;
                                    w2nTextField[6].setText(""+ ARABS.classK);
                              }
                        }
                  }
             }
   }
}

CSFocusListener wTSFocusListener = new CSFocusListener();

class CSFocusListenerD implements FocusListener
{
    public void focusGained(FocusEvent e) {}
    public void focusLost(FocusEvent e)
      {
            int  aa=0;
          boolean ex=false;
            String Str = ((JTextField)e.getSource()).getText();
            Object object = e.getSource();
            try
            {
                  aa=  Short.valueOf(Str).shortValue();
                  getAppletContext().showStatus("Last Value set to: "+Str);
            }
          catch( NumberFormatException exc)

            {
                   ex= true;
            }
            if (((aa<0)||( ex))&(object != w2nTextField[1]))
            {
                  ex= true;
                  fc=fc+1;if (fc == 3) fc = 1;if (fc == 1)
                  {
                        JOptionPane.showMessageDialog(wTabbedPane, "Field value
must be a number!", "Warning!", JOptionPane.WARNING_MESSAGE);
                  }
            }
            if (object ==  w2nTextField[0])
            {
                  if(ex)
                      w2nTextField[0].setText(""+ModelAR_ABS.class0);
                  else  if (aa==0)
                  {
```

```
                              fc=fc+1; if (fc == 3) fc = 1;if (fc == 1)
                        {
                              JOptionPane.showMessageDialog(wTabbedPane, "Value
must be greater than 0!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    ARABS.class0 =1;
                                    w2nTextField[0].setText(""+ ARABS.class0);
                              }
                  }
                  else
                        ModelAR_ABS.class0 =aa;
            }
             else if (object==  w2nTextField[1] )
             {
                   try
                  {
                         double d = Double.parseDouble(Str);
                         if (d < 0) d = d * -1;
                         ARABS.classD = d;
                        if (d > 1)
                        {
                              fc=fc+1; if (fc == 3) fc = 1;if (fc == 1)
                        {
                              JOptionPane.showMessageDialog(wTabbedPane, " value
must be less than 1!", "Warning!", JOptionPane.WARNING_MESSAGE);
                                    ARABS.classD =0.0;
                                    w2nTextField[1].setText(""+ ARABS.classD);
                              }
                        }
               } catch( NumberFormatException exc)

                  {
                        fc=fc+1;  if (fc == 3) fc = 1;      if (fc == 1)
                        JOptionPane.showMessageDialog(wTabbedPane, "Field value
must be a number!", "Warning!", JOptionPane.WARNING_MESSAGE);
                        ARABS.classD =0.0;
                        w2nTextField[1].setText(""+ ARABS.classD);
                  }
            }
              else if (object ==  w2nTextField[2] )
              {
                   if(ex)
                        w2nTextField[2].setText(""+ModelAR_ABS.class2);
                   else
                         ModelAR_ABS.class2 =aa;
                   }
            }
   }
CSFocusListenerD wTSFocusListenerD = new CSFocusListenerD();

//--------------ActionListener--------------- File
Chooser---------------------------------------
class CFActionListener implements ActionListener
{
      public void actionPerformed(ActionEvent e)
      {
            try
            {
                  JFileChooser wFileChooser = new JFileChooser();
                  wFileChooser.setLocation( 250,250);
```

```
                wFileChooser.changeToParentDirectory();
                int returnVal = wFileChooser.showOpenDialog(w1Panel);
                if(returnVal == JFileChooser.APPROVE_OPTION)
                {
                            IFile=wFileChooser.getSelectedFile().getName();
                            Dir=
wFileChooser.getCurrentDirectory().getAbsolutePath()+"/"+IFile;
                            File my = new File(Dir);
                            if (my.exists()== false)
                            {
                               JOptionPane.showMessageDialog(null, "Can't open.
No such file!", "Error!", JOptionPane.ERROR_MESSAGE);
                               IFile="DNumber";
                               Dir="";
                            }
                            w11TextField.setText(IFile);
                }
          }
          finally {}
     }
}
CFActionListener wFActionListener = new CFActionListener();

//---------------------------------------------------------------
class PerformCount extends Thread
{
     ReadDataFile  RDF= new ReadDataFile();
     int prog, inkr=0;
     double sigma;
     boolean Stop = false, red= false;
     double[] TAll =new double[ARABS.lenght];
     double[] PAll =new double[ARABS.lenght];
       private double[] TempPast =new double[ARABS.lenght];
     private double[] TempPast2 =new double[ARABS.lenght];
          private  double[] Mi =new double[ARABS.lenght];
               private  double[] Ma =new double[ARABS.lenght];
                    private  double[] Me =new double[ARABS.lenght];
                     int TempClass2 =ARABS.class2;

   public void run()
     {

              //ppp4'1/2'
               ARABS.pgclass6=  TempClass2/2;
          //ARABS.pgclass6=  TempClass2;
for (int i=1; i< (TempClass2+ARABS.pgclass6+2)*ARABS.class0; i++)
                    {
                            TempPast[i]=ARABS.past[i];
                            TempPast2[i]=ARABS.past[i];
                     }
          Solver = new MyLp();
          Pred= new Prediction();
          double ar=0, rand, nr, opt=90000000;
          String Str = w2nTextField[1].getText();
          ARABS.class1=  Short.valueOf(Str).shortValue();
          int tempP=ARABS.class1;
          int T= ARABS.class2, rg=0;
          prog=0;
```

```
            if (wItVector.size() !=0)
            {
                  wItVector.removeAllElements();
                  wFVector.removeAllElements();
            }
            if((ARABS.class3==0)&&(ARABS.class4==0)&&(ARABS.classM==0))
            {
            prog=10;

                                    wProgressBar.setMaximum(prog);

                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);
      if(   Solver.MyLpSolve(Task))
            {

                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);

                                                      if (Stop) Finish();
                  out.writeString( "  ----------------------Parameters: P = "+tempP
+" ; T = "+T+"----------------------\n");
                  t.append( "  ----------------------Parameters: P = "+tempP+" ; T
= "+T+"----------------------\n");

                                    wProgressBar.setValue(inkr++);
                  for (int i=1; i<= (ARABS.class1*ARABS.class0); i++)
                  {
                        t.append("   a[ "+i+" ]"+ ARABS.Acoef[i]+"\n");
                  out.writeString("   a[ "+i+" ]"+ARABS.Acoef[i]+"\n");
            }

                                    wProgressBar.setValue(inkr++);

                                                      if (Stop) Finish();
                  sigma=Pred.Residuals();
           for (int i=2; i<= ARABS.class2; i++)
                 {
                  out.writeString("Residuals-1 ["+ i +"] "+ ARABS.Residuals[i]
+"\n");
                        ar=ar+Math.abs(ARABS.Residuals[i]);
                  }

                                    wProgressBar.setValue(inkr++);

                                                      if (Stop) Finish();
                  out.writeString("The average of residuals for known data-1: "+ar/
(ARABS.class2-1)+"\n");
                  out.writeString("The variance of residuals for known data:
"+sigma+"\n");
                  ar=Pred.Predict();
                  t.append("Next predicted: "+ar+"\n");
                  out.writeString("Next predicted: "+ar+"\n");
```

```
                if (Bcompare)
                {
                      t.append("Should be     :
"+ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]+"\n");
                      out.writeString("Should be     :
"+ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]+"\n");
                      t.append("Residual      : "+
(ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]-ar)+"\n");
                      out.writeString("Residual-0      : "+
(ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]-ar)+"\n");
                }
                else
                {
                      t.append("Should be     : unnown\n");
                      out.writeString("Should be     : unnown\n");
                      t.append("Residual      : unnown\n");
                      out.writeString("Residual      : unnown\n");
                }
          }
          else
                      {
                            t.append("Lp_solve hasn't found the solution to the
problem.\n");
                            out.writeString("Lp_solve hasn't found the solution
to the problem.\n");
                      }

                                        wProgressBar.setValue(prog);
          }
//=============================================================================
=======================//
          else  if ((ARABS.class3==0)&&(ARABS.class4==1))
          {
                prog=10;

                                  wProgressBar.setMaximum(prog);

                                  wProgressBar.setValue(inkr++);

                                  wProgressBar.setValue(inkr++);
          if(   Solver.MyLpSolve(Task))
          {

                                  wProgressBar.setValue(inkr++);

                                  wProgressBar.setValue(inkr++);

                                  wProgressBar.setValue(inkr++);

                                  wProgressBar.setValue(inkr++);

                                                if (Stop) Finish();
                for (int i=1; i<= (ARABS.class1*ARABS.class0); i++)
                {
                out.writeString("   a[ "+i+" ]"+ARABS.Acoef[i]+"\n");

}

                                                if (Stop) Finish();
```

```
                  TAll= Pred.PredictMulti();

                                    wProgressBar.setValue(inkr++);
                  sigma=Pred.Residuals();
                for (int i=2; i<= ARABS.class2; i++)
                {
                  out.writeString("Residuals-2 ["+ i +"] "+ ARABS.Residuals[i]
+"\n");
                        ar=ar+Math.abs(ARABS.Residuals[i]);
                  }
                  out.writeString("The average  of residuals for known dat-2 :
"+ar/(ARABS.class2-1)+"\n");
                  out.writeString("The variance  of residuals for known data :
"+sigma+"\n");


//System.out.println(" \nParameter p= "+P+"\n");
System.out.println("predicted actual");
for (int i=1; i<= TempClass2/2; i++)
 {
                                    rand= Math.round(100000*TAll[i]);
                                    rand=rand/100000;
                                    out.writeString(" " +(i+ARABS.class2)+"
"+ rand+  "                  "+
ARABS.past[(i-1)*ARABS.class0+1 +ARABS.class2*ARABS.class0]+"                "+
(ARABS.past[(i-1)*ARABS.class0+1 +ARABS.class2*ARABS.class0]-TAll[i])+    "\n");
                                    ar= ar+
Math.abs(ARABS.past[(i-1)*ARABS.class0+1 +ARABS.class2*ARABS.class0]-TAll[i]);
System.out.println("  "+  rand+ "     "+ARABS.past[(i-1)*ARABS.class0+1
+ARABS.class2*ARABS.class0]+ "   " );
//ppp3
//out.writeString("   a[ "+i+" ]"+ARABS.Acoef[i]+"\n");
                             }


wProgressBar.setValue(inkr++);

                                                  if (Stop) Finish();
            out.writeString(" \n Parameter p= "+tempP+"\n");
                  t.append("\n Parameter p= "+tempP+"\n");


out.writeString("_____________________________________________________________
___\n\n");
                  out.writeString(" t.         Predicted              Should be
Residual           \n");


out.writeString("_____________________________________________________________
___\n");


t.append("_____________________________________________________________\n\
n");
                  t.append(" t.        Predicted         Should be
Residual           \n");


t.append("_____________________________________________________________\n"
```

```
);
                ar=0;

                                wProgressBar.setValue(inkr++);

                for (int i=1; i<= ARABS.pgclass6; i++)
                {
                      rand= Math.round(100000*TAll[i]);
                      rand=rand/100000;
                      nr= ARABS.past[(i-1)*ARABS.class0+1
+ARABS.class2*ARABS.class0]-TAll[i];
                      out.writeString(" " +  (i+ARABS.class2)+"       "+ rand+  "
"+ ARABS.past[(i-1)*ARABS.class0+1 +ARABS.class2*ARABS.class0] +"
"+(nr)+    "\n");
                      t.append(" " +(i+ARABS.class2)+"       "+ rand+  "
"+ ARABS.past[(i-1)*ARABS.class0+1 +ARABS.class2*ARABS.class0]+"                 "+
(nr)+    "\n");
                      ar= ar+ Math.abs(nr);
            }


out.writeString("_____
___\n\n");


t.append("_____\n"
);
                out.writeString("Sum of all residuals: "+ar+"\n");
                t.append("Sum of all residuals: "+ar+"\n");
                out.writeString("The average  of residuals:
"+ar/ARABS.pgclass6+"\n");
                t.append("The average  of residuals: "+ar/ARABS.pgclass6+"\n");
            }
            else
            {
                        t.append("Lp_solve hasn't found the solution to the
problem.\n");
                        //out.writeString( "  ----------------------
Parameters: P = "+P+" ; T = "+T+"----------------------\n");
                        out.writeString("Lp_solve hasn't found the solution
to the problem.\n");
            }

                                    wProgressBar.setValue(prog);
      }
//=============================================================================//

            else  if ((ARABS.class3==1)&&(ARABS.class4==0)&&(ARABS.classM==0))
            {
                  int temp =ARABS.class2;
            prog=3*ARABS.class1+2;

                                wProgressBar.setMaximum(prog);

                                wProgressBar.setValue(inkr++);
                wGFrame = new JFrame("Residuals graphic");
                MeCanvas mCanvas = new MeCanvas(Iterations, wItVector, wFVector);
                wGFrame.getContentPane().add( mCanvas );
                t.append("\n Parameter T= "+ARABS.class2+"\n");
```

```
t.append("_____________________________________________________________________\n\
n");
                t.append(" p.        Predicted         Should be
Residual           \n");


t.append("_____________________________________________________________________\n"
);
                int have=0;
                for (int P=1; P<=tempP; P++)
                {

                if (Stop) {Finish(); ARABS.class1= tempP;}
                      ARABS.class1=P;

                                  wProgressBar.setValue(inkr++);
                      if (Solver.MyLpSolve(Task))
                      {

                if (Stop) {Finish(); ARABS.class1= tempP;}
                            out.writeString( "  ----------------------Parameters:
P = "+P+" ; T = "+T+"----------------------\n");
                      for (int i=1; i<= ARABS.class0*ARABS.class1; i++)
                            {
                                  out.writeString("   a[ "+i+" ]"+ARABS.Acoef[i]
+"\n");

                            }

                if (Stop) {Finish(); ARABS.class1= tempP;}
                            sigma=Pred.Residuals();

                                  wProgressBar.setValue(inkr++);

                            ar=0;
                            for (int i=2; i<= ARABS.class2; i++)
                            {
                                  //out.writeString("Residuals-3 ["+ i +"] "+
ARABS.Residuals[i]+"\n");
                                  ar=ar+Math.abs(ARABS.Residuals[i]);
                            }
                            out.writeString("The average  of residuals for known
data-3 : "+ar/ARABS.class2+"\n");
                            out.writeString("The variance  of residuals for known
data : "+sigma/ARABS.class2+"\n");
                      TAll[P]=Pred.Predict();
                            wItVector.addElement(new Integer(P));
                      wFVector.addElement(new
Double(Math.abs(ARABS.past[ARABS.class2*ARABS.class0+1]-TAll[P])));
                            mCanvas.repaint();
                            out.writeString("Next predicted: "+TAll[P]+"\n");

                                  wProgressBar.setValue(inkr++);
                            rand= Math.round(100000*TAll[P]);
                           rand=rand/100000;
                if (Bcompare)
                {
                            out.writeString("Should be     :
```

```
"+ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]+"\n");
                              out.writeString("Residual      : "+
(ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]-TAll[P])+"\n");
                              t.append(" " +(P)+"       "+ rand+  "
"+ ARABS.past[ARABS.class2*ARABS.class0+1]+"                "+
(ARABS.past[ARABS.class2*ARABS.class0+1]-TAll[P])+    "\n");
                  }
                  else
                  {
                              out.writeString("Should be     : unnown\n");
                              out.writeString("Residual      : unnown\n");
                              t.append(" " +(P)+"       "+ rand+  "
unnown                  unnown\n");
                  }

                  if (Stop) {Finish();; ARABS.class1= tempP;}
                        }
                        else
                        {
                              t.append("Lp_solve hasn't found the solution to the
problem.\n");
                              out.writeString( "  ----------------------Parameters:
P = "+P+" ; T = "+T+"----------------------\n");
                              out.writeString("Lp_solve hasn't found the solution
to the problem.\n");
                        }
                  }
                  out.writeString(" \nParameter T= "+ARABS.class2+"\n");


out.writeString("_____________________________________________________________________
___\n\n");
                  out.writeString(" p.         Predicted               Should be
Residual           \n");


out.writeString("_____________________________________________________________________
___\n");
                  ar=0;    int no=0;     rg=0;   opt=90000000;
                  for (int i=1; i<= ARABS.class1; i++)
                  {
                        rand= Math.round(100000*TAll[i]);
                        rand=rand/100000;
                        if (Bcompare)
                        {
                              nr=ARABS.past[ARABS.class2*ARABS.class0+1]-TAll[i];
                              if (TAll[i] != 0)
                              {
                              ar= ar+Math.abs(nr);
                              no++;
                              }
                              if (opt > Math.abs(nr))
                              {
                                    opt= Math.abs(nr);
                                     rg=i;
                              }
                                    out.writeString(" " +(i)+"       "+ rand+  "
"+ ARABS.past[ARABS.class2*ARABS.class0+1]+"                "+(nr)+    "\n");
```

```
                                t.append(" " +(i)+"       "+ rand+  "
"+ ARABS.past[ARABS.class2*ARABS.class0+1]+"                  "+
(ARABS.past[ARABS.class2*ARABS.class0+1]-TAll[i])+    "\n");
                          }
                          else
                          {
                                out.writeString(" " +(i)+"       "+ rand+  "
unnown                  unnown\n");
                          }
                  }


out.writeString("_____________________________________________________________________
___\n\n");


t.append("_____________________________________________________________________\n"
);
                  if (Bcompare)
                  {
                  out.writeString("Sum of all residuals: "+ar+"\n");
                  t.append("Sum of all residuals: "+ar+"\n");
                  out.writeString("The average  of residuals-1: "+ar/no+"\n");
                  t.append("The average  of residuals-2: "+
(ARABS.past[ARABS.class2*ARABS.class0+1]-TAll[rg])+"\n");
                  out.writeString("The optimal-1 p  parameter: "+rg+"\n");
                  t.append("The optimal-2 p  parameter: "+rg+"\n");
                  }

                                          wProgressBar.setValue(prog);
                  ARABS.class1= tempP;
                  mCanvas.repaint();
            }
//============================================================================//

            else  if
((ARABS.class3==1)&&(ARABS.class4==1))
            {

prog=3*ARABS.class1+2;
                   ARABS.pgclass6=ARABS.class2;

                                  wProgressBar.setMaximum(prog);

                                  wProgressBar.setValue(inkr++);
                  int optimalp=1;
                  double detail=0;
                  wGFrame = new JFrame("Residuals graphic");
                  MeCanvas mCanvas = new MeCanvas(Iterations, wItVector, wFVector);
                  wGFrame.getContentPane().add( mCanvas );
                  t.append("\n The optimization-1 of p parameter:\n");

      t.append("_____________________________________________\n\n");
                  t.append(" p.          The average of residuals           \n");
                  t.append("_____________________________________________\n");

             for (int P=1; P<=tempP; P++)
                  {
                        ARABS.class1=P;
```

```
                  if(   Solver.MyLpSolve(Task))
{
                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);

                                    if (Stop) {Finish();; ARABS.class1= tempP;}

                               TAll= Pred.PredictMulti();

                  if (Stop) {Finish();; ARABS.class1= tempP;}
                        out.writeString(" \nParameter p= "+P+"\n");


out.writeString("_____________________________________________________________________
___\n\n");
                               out.writeString(" T.         Predicted
Should be                Residual            \n");


out.writeString("_____________________________________________________________________
___\n");
                               ar=0;
//ppp7
//for (int i=1; i<= ARABS.pgclass6-4; i++)
//for (int i=1; i<= ARABS.pgclass6; i++) //orig
//aist1

TAll= Pred.PredictMulti();

      wProgressBar.setValue(inkr++);
sigma=Pred.Residuals();
for (int i=2; i<= ARABS.class2; i++)
                {
                  out.writeString("Residuals-4 ["+ i +"] "+ ARABS.Residuals[i]
+"\n");
                        ar=ar+Math.abs(ARABS.Residuals[i]);
                  }
                  out.writeString("The average  of residuals for known dat-2 :
"+ar/(ARABS.class2-1)+"\n");


System.out.println(" \nParameter p= "+P+"\n");
System.out.println("predicted actual");
for (int i=1; i<= TempClass2/2; i++)
 {
                                    rand= Math.round(100000*TAll[i]);
                                    rand=rand/100000;
                                    out.writeString(" " +(i+ARABS.class2)+"
"+ rand+  "                  "+
ARABS.past[(i-1)*ARABS.class0+1 +ARABS.class2*ARABS.class0]+"                "+
(ARABS.past[(i-1)*ARABS.class0+1 +ARABS.class2*ARABS.class0]-TAll[i])+    "\n");
                                    ar= ar+
Math.abs(ARABS.past[(i-1)*ARABS.class0+1 +ARABS.class2*ARABS.class0]-TAll[i]);
System.out.println("  "+  rand+ "     "+ARABS.past[(i-1)*ARABS.class0+1
+ARABS.class2*ARABS.class0]+ "   " );
//ppp3
//out.writeString("   a[ "+i+" ]"+ARABS.Acoef[i]+"\n");
```

```
                              }


out.writeString("_____________________________________________________________________
___\n\n");


//ppp2
for (int j=1; j<= ARABS.class0*ARABS.class1; j++)
                              {
                                    out.writeString("   a[ "+j+" ]"+ARABS.Acoef[j]
+"\n");
                              }
wProgressBar.setValue(inkr++);
                              out.writeString("Sum of all residuals: "+ar+"\n");
//ppp8'1/2'
out.writeString("The average  of residuals-pall-1: "+ar*2/TempClass2+"\n");
                              PAll[P]=ar;
                              t.append(" " +(P)+"       "+

PAll[P]/ARABS.pgclass6+  "\n");
                              wItVector.addElement(new Integer(P));
                        wFVector.addElement(new
Double(Math.abs(PAll[P]/ARABS.pgclass6)));
                              mCanvas.repaint();
                  }
                  else
                  {
                              t.append("Lp_solve hasn't found the solution to the
problem.\n");
                              out.writeString( "  ----------------------Parameters:
P = "+P+" ----------------------\n");
                              out.writeString("Lp_solve hasn't found the solution
to the problem.\n");
                  }
            }

                  if (Stop) {Finish(); ARABS.class1= tempP;}
                  ar= PAll[1];
            out.writeString("\n The optimization-2 of p parameter:\n");

      out.writeString("_____________________________________________________\n\n");
                  out.writeString(" p.          The average of residuals
\n");

      out.writeString("_____________________________________________________\n");
                  int no=0;
                  for (int i=1; i<= ARABS.class1; i++)
                  {
//ppp9
out.writeString("opt-2 " +(i)+"       "+ PAll[i]*2/TempClass2+     "\n");
//ppp2
//out.writeString("   a[ "+i+" ]"+ARABS.Acoef[i]+"\n");
if   (PAll[i] != 0)
                        {
                              no++;
                        if   (PAll[i]<ar)
                        {
                              ar=PAll[i];
```

```
                              optimalp=i;
                        }
                        detail=detail+PAll[i]/ARABS.pgclass6;
                        }
                  }

      out.writeString("_____________________________________________________\n\n");

      t.append("_____________________________________________________\n\n");
//ppp1
//for (int j=1; j<= ARABS.class0*ARABS.class1; j++)
//                              {
//                                    out.writeString("   a[ "+j+" ]"+ARABS.Acoef[j]
+"\n");
//                              }


if (Stop) {Finish(); ARABS.class1= tempP;}
            out.writeString("The optimal-3 p parameter: "+optimalp+"\n");
                  out.writeString("The optimal-4 average :
"+PAll[optimalp]*2/TempClass2+"\n");
                  t.append("The optimal-5 p parameter: "+optimalp+"\n");

                                          wProgressBar.setValue(prog);
                  ARABS.class1= tempP;
                  mCanvas.repaint();
       }
       else  if ((ARABS.class3==0)&&(ARABS.classM==1))
            {
                        int M =   ARABS.class0;

                         prog=10;

                              wProgressBar.setMaximum(prog);

                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);
      if(   Solver.MyLpSolve(Task))
            {

                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);

                                    wProgressBar.setValue(inkr++);

                                                      if (Stop) Finish();

                              wProgressBar.setValue(inkr++);
                  for (int i=1; i<= (ARABS.class1*ARABS.class0); i++)
                  {
                        out.writeString("   a[ "+i+" ]"+ARABS.Acoef[i]+"\n");
            }

                                    wProgressBar.setValue(inkr++);
```

```
                                                    if (Stop) Finish();
                  sigma=Pred.Residuals();
          for (int i=2; i<= ARABS.class2; i++)
                {
                  out.writeString("Residuals-5 ["+ i +"] "+ ARABS.Residuals[i]
+"\n");
                        ar=ar+Math.abs(ARABS.Residuals[i]);
                  }

                                  wProgressBar.setValue(inkr++);

                                                    if (Stop) Finish();
                  out.writeString("The average of residuals for known data-4: "+ar/
(ARABS.class2-1)+"\n");
                  out.writeString("The variance of residuals for known data:
"+sigma+"\n");
                  out.writeString(" \nParameter p= "+tempP+"\n");


out.writeString("_____________________________________________________________________
________\n\n");
                  out.writeString(" T.      Pred.min          Pred.max
Pred.mean         Residual            \n");


out.writeString("_____________________________________________________________________
________\n");
                  t.append(" \nParameter p= "+tempP+"\n");


t.append("_____________________________________________________________________
_\n\n");
                  t.append(" T.      Pred.min          Pred.max           Pred.mean
Residual            \n");


t.append("_____________________________________________________________________
_\n");
                  for (int i=1; i< (TempClass2+ARABS.pgclass6+1)*M; i++)
                      {
                              Mi[i]= ARABS.past[i];
                              Ma[i]= ARABS.past[i];
                        }
                         sigma=Pred.Residuals();
                  ARABS.class2=TempClass2;
                  for (int i=0; i< (ARABS.pgclass6); i++)
                  {
                    double min=100000;
                    ARABS.class2=TempClass2+i;
                    ar=Pred.Predict();
                   // sigma=Pred.Residuals1();
                    for (int k=1; k<= (ARABS.classK); k++){
                                TAll[k]=ar+Pred.Noise()*sigma;
                               if  (TAll[k]<min)
                                       min=TAll[k];
                  }
//ppp0301
//System.out.println("  sigma  ar min  "+  sigma+"   "+ar+ "   "+ min  );
```

```
                    ARABS.past[(TempClass2+i)*M+1]=min;
                    Mi[TempClass2+i+1]= min;
                   }
                    for (int j=1; j< (TempClass2+ARABS.pgclass6+2)*M; j++)
                       {
                               System.out.println("j1  "+(j)+" ARABS.past "  +
ARABS.past[j]);
                               ARABS.past[j]=TempPast[j];
}
                    ARABS.class2=TempClass2;
                    for (int i=0; i< (ARABS.pgclass6); i++)
                   {
                      double  max=0.0000001;
                      ARABS.class2=TempClass2+i;
                      ar=Pred.Predict();
                      //sigma=Pred.Residuals1();
                      for (int k=1; k<= (ARABS.classK); k++){
                                  TAll[k]=ar+Pred.Noise()*sigma;
                                  if  (TAll[k]>max)
                                         max=TAll[k];
                   }
                   ARABS.past[(TempClass2+i)*M+1]=max;
                    Ma[TempClass2+i+1]= max;

                   }
                   for (int i=1; i< (TempClass2+ARABS.pgclass6+2)*M; i++)
                       {
                               ARABS.past[i]=TempPast[i];
                         }
                         ARABS.class2=TempClass2;
                         for (int i=0; i< (ARABS.pgclass6); i++)
                         {
                         double  mean=0;
                         ARABS.class2=TempClass2+i;
                         ar=Pred.Predict();
                         //sigma=Pred.Residuals1();
                        for (int k=1; k<= (ARABS.classK); k++){
                                  TAll[k]=ar+Pred.Noise()*sigma;
                                  mean+=(TAll[k]); }
                         mean/=ARABS.classK;
                         System.out.println("pall-2 ar  sigma mean  "+  ar+ "
"+sigma+ "   "+ mean );
                         PAll[i]=TempPast[ (TempClass2+i)*M+1]-mean;
                         out.writeString(" "+(ARABS.class2+1)+ "     "+
(Math.round(100000*Mi[ARABS.class2+1])/100000)+ "     "+
(Math.round(100000*Ma[ARABS.class2+1])/100000)+ "     "+
(Math.round(100000*mean)/100000)+ "     "+ (Math.round(100000*PAll[i])/100000)+"
"+TempPast[ (TempClass2+i)*M+1]+"\n");
                       t.append(" "+(ARABS.class2+1)+ "     "+
(Math.round(100000*Mi[ARABS.class2+1])/100000)+ "     "+
(Math.round(100000*Ma[ARABS.class2+1])/100000)+ "     "+
(Math.round(100000*mean)/100000)+ "     "+ (Math.round(100000*PAll[i])/100000)+"
"+TempPast[ (TempClass2+i)*M+1]+"\n");
                     ARABS.past[(TempClass2+i)*M+1] =mean;

                }
//              ARABS.past =TempPast;
                   ARABS.class2=TempClass2;
```

```
out.writeString("_____
________\n\n");

t.append("_____
_\n\n");
                  sigma=0;
                  for (int i=0; i< (ARABS.pgclass6); i++) {
                                    sigma+=Math.abs(PAll[i]);
//aistpr
System.out.println("i  "+(TempClass2+i)+" min "  + Mi[TempClass2+i]);
                                 // System.out.println("i  "+(TempClass2+i)+" max "  +
Ma[TempClass2+i]);
                  }
                  sigma/= ARABS.pgclass6;
                  out.writeString(" Avr progn err: "+ sigma);
                   t.append( " Avr progn err: "+ sigma);
      /*          if (Bcompare)
                  {
                        t.append("Should be     :
"+ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]+"\n");
                        out.writeString("Should be     :
"+ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]+"\n");
                        t.append("Residual      : "+
(ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]-ar)+"\n");
                        out.writeString("Residual      : "+
(ModelAR_ABS.past[(ARABS.class2*ARABS.class0+1)]-ar)+"\n");
                  }
                  else
                  {
                        t.append("Should be     : unnown\n");
                        out.writeString("Should be     : unnown\n");
                        t.append("Residual      : unnown\n");
                        out.writeString("Residual      : unnown\n");
                  }
            }*/

            }
            else
                        {
                              t.append("Lp_solve hasn't found the solution to the
problem.\n");
                              out.writeString("Lp_solve hasn't found the solution
to the problem.\n");
                        }
       ARABS.class2 =TempClass2;

                                    wProgressBar.setValue(prog);
            }
       wItVector.addElement(new Integer(0));
       wFVector.addElement(new Double(0));
     Finish();
      }
      void Finish()
      {
            w2Button.setEnabled(false);
            w1Button.setEnabled(true);
            w4Button.setEnabled(true);
            if(ARABS.class3==1)
                  w3Button.setEnabled(true);
```

```
          wTabbedPane.setEnabledAt(0, true);
          wTabbedPane.setEnabledAt(1, true);
            stop();
      }
      public void StopCount()
      {
            Stop = true;
      }
}
//---------PerformDiagnosis
class PerformDiagnosis extends Thread
{
    boolean Stop = false;
      double avr;
      String str="";
      private void route (double[][] Probability )
      {
            int counter=0;
              t.append("Rounded rezults:  \n");
              out.writeString("  \n");
              out.writeString("Rounded rezults:  \n");
             for (int i=1; i<=ARABS.class5; i++ )
              {
                        t.append("---------");
                    out.writeString   ("---------");
              }
              t.append("----------------------------\n");
          out.writeString     ("----------------------------\n");
              t.append("|                                    Class
| Scould be | Calculated |   \n");
              out.writeString   ("|                                          Class
| Scould be | Calculated |   \n");
              for (int i=1; i<=ARABS.class5; i++ )
              {
                        t.append("---------");
                    out.writeString   ("---------");
              }
              t.append("----------------------------\n");
          out.writeString     ("----------------------------\n");
              for (int i=1; i<=ARABS.class5; i++ )
            {
                  t.append("      "+ARABS.ClassAll[i]);
            out.writeString   ("      "+ARABS.ClassAll[i]);
            }
            t.append("  \n");
            out.writeString("  \n");
             int det=0;
        for (int j=1; j<=ARABS.class6; j++)
             {
                  det=0;
                  for (int i=1; i<=ARABS.class5; i++ )
                  {
                        if ((Probability[j][i]) >= (1-ARABS.classD/2))
                        {
                              out.writeString("   "+1.0);
                              t.append("   "+1.0);
                              str = ARABS.ClassAll[i];
                              det++;
                        }
```

```
                        else if (Probability[j][i] <= (0+ARABS.classD/2))
                        {
                              out.writeString("   "+0.0);
                              t.append("   "+0.0);
                        }
                        else
                        {
                              out.writeString("   "+ "No.");
                              t.append("   "+ "No.");
                   }
                  if (det!=1)
                  str="NoDecision";
                  }
                  if  ((str).compareTo(ARABS.newsAll[j])==0)
                        counter++;
                  out.writeString   (" |            "+ARABS.newsAll[j]+ "
"+str);
                  t.append(" |            "+ARABS.newsAll[j]+ "
"+str);
                  t.append("  \n");
                  out.writeString("  \n");
              }
               for (int i=1; i<=ARABS.class5; i++ )
               {
                        t.append("---------");
                  out.writeString   ("---------");
               }
               t.append("----------------------------\n");
          out.writeString     ("----------------------------\n");
               t.append("Number of  examples: "+ ARABS.class6+ " \n");
               out.writeString("Number of examples: "+ ARABS.class6+ " \n");
               t.append("Number of rigth clasified examples: "+ counter+ " \n");
               out.writeString("Number of rigth clasified examples: "+ counter+
" \n");
               wProgressBar.setValue(ARABS.class6+3);
      }
      public void run()
      {
            Dig = new Diagnosis();
            int prog=ARABS.class6+3, inkr=0;
          double prob=0;
            double[][] Temp = new double[40][40];
            wProgressBar.setMaximum(prog);
            wProgressBar.setValue(inkr++);
            wProgressBar.setValue(inkr++);
            w2Button.setEnabled(true);
            out.text.setText("");
            if (Stop) {Finish();}
            Dig.Distinct();
            for (int i=1; i<=ARABS.class5; i++ )
                  out.writeString   ("                        "+ARABS.ClassAll[i]);
            out.writeString("   \n");
            out.writeString("Not normalized values:   \n");
            t.append("With not normalized values   \n");
            wProgressBar.setValue(inkr++);
            if (Stop) {Finish();}
            for (int i=1; i<=ARABS.class5; i++ )
            {
                  Solver= new MyLp();
```

```
                  //System.out.println("Klase::::::::"+ i);
                  //System.out.println("ARABS.class5:"+ ARABS.class5+ "
ARABS.class2" +ARABS.class2 );
                  Dig.ChangeClass(ARABS.ClassAll[i]);
                  if (Solver.MyLpSolve(false))
                        for (int j=1; j<=ARABS.class0; j++)
                          ARABS.ADcoef[i][j]=ARABS.Acoef[j];
                        else
                  for (int j=1; j<=ARABS.class0; j++)
                          ARABS.ADcoef[i][j]= -100;
            }
            for (int j=1; j<=ARABS.class6; j++)
             {
                  for (int i=1; i<=ARABS.class5; i++ )
                  {
                  if (Stop) {Finish();}
                        prob=Dig.Diagnose(ARABS.class0*(j-1), i);
                        ARABS.Probability[j][i]= prob;
                        out.writeString("  "+ARABS.Probability[j][i]);
                  }
                  Dig.norm(j, Temp);
                  if (Stop) {Finish();}
                  wProgressBar.setValue(inkr++);
                  out.writeString("  \n");
             }
             route(ARABS.Probability);
             out.writeString("Normalized:  \n");
             t.append("With normalized values   \n");
             for (int j=1; j<=ARABS.class6; j++)
             {
                  for (int i=1; i<=ARABS.class5; i++ )
                  {
                  out.writeString("    "+Temp[j][i]);
                  }
                out.writeString("  \n");
            }
          route(Temp);
              out.writeString("Residual: "+ Dig.Resid()+"\n");
              Finish();
             }

      void Finish()
      {
            w2Button.setEnabled(false);
            w1Button.setEnabled(true);
            w4Button.setEnabled(true);
          wTabbedPane.setEnabledAt(0, true);
          wTabbedPane.setEnabledAt(1, true);
            stop();
      }
      public void StopCount()
      {
            Stop = true;
      }
}
//---------ActionListener----------------
Grafic---------------------------------------

public class Grafic implements ActionListener
```

```
{
      public void actionPerformed(ActionEvent e)
      {
            int X = 210, Y = 10, Xx = 350, Yy = 450;
            if (wGFrame != null)
            {
                  wGFrame.dispose();
            }
            wGFrame.setLocation(X, Y);
            wGFrame.setSize(Xx, Yy);
            wGFrame.setVisible(true);
            }
 }
//---------ActionListener----------------For Simple File
read---------------------------------------

public class PicFile implements ActionListener
{
      ReadDataFile  RDF= new ReadDataFile();

      public void actionPerformed(ActionEvent e)
      {
            t.setText("");
            RDF.SimpleRead(Dir, IFile,t);
            getAppletContext().showStatus("Reading file: "+IFile);
      }
 }
//-------------------------------------------------------------------------------
----------------------
public class Stop implements ActionListener
{
      ReadDataFile  RDF= new ReadDataFile();
      ReadData1   RDT = new ReadData1();
    String name;
      private void routine()
            {
                  if ( ARABS.class3 ==1 )
                     {
                         Iterations =ARABS.class1;
                       w3Button.setEnabled(true);
                        }
                        w1Button.setEnabled(false);
                        w4Button.setEnabled(false);
                        w2Button.setEnabled(true);
                        wTabbedPane.setEnabledAt(0, false);
                        wTabbedPane.setEnabledAt(1, false);
                   try
                     {
                      if (Task)
                          {
                                PC= new PerformCount();
                                PC.setPriority(PC.MIN_PRIORITY);
                                PC.start();
                             }
                        else
                           {
                                ARABS.class0= ARABS.class0-1;
                                PD= new PerformDiagnosis();
                                Red = new ReadData1();
```

```
                              ARABS.class6 =Red.ReadDataArea1()/ARABS.class0;
                              PD.setPriority(PD.MIN_PRIORITY);
                        PD.start();
                              w1Button.setEnabled(true);
                              w2Button.setEnabled(false);
                              w4Button.setEnabled(false);
                              wTabbedPane.setEnabledAt(0, true);
                              wTabbedPane.setEnabledAt(1, true);
                             }
                     }
                  catch (Exception ex)
                  {
                        w1Button.setEnabled(true);
                        w2Button.setEnabled(false);
                        w4Button.setEnabled(false);
                        wTabbedPane.setEnabledAt(0, true);
                        wTabbedPane.setEnabledAt(1, true);
                  }
          }
      public void actionPerformed(ActionEvent e)
      {
            int acount=0, sel1, sel2, p;
            if (Task==false)
            {
                  String Str =w2nTextField[0].getText();
                  ARABS.class0=  Short.valueOf(Str).shortValue()+1;
                  Str =w2nTextField[2].getText();
                  ARABS.class2=  Short.valueOf(Str).shortValue();
            }
            Object wObject = e.getSource();
           if ( wObject == w1Button)
            {
                  if (wGFrame != null)
                        wGFrame.dispose();
                        if ( rb1.isSelected())
                        {
                              if (Task)
                                    acount= RDF.ReadData(Dir, IFile);
                              else
                                  acount= RDF.ReadDataforDiagnosis(Dir, IFile);
                        }
                        else
                        {    //System.out.println(""+t1.getText());
                         t.setText("");
                               if (Task)
                                    acount= RDT.ReadAreaProgn();
                               else
                                    acount= RDT.ReadAreaDiag();
                        }
                  p=  ARABS.class1;
                  getAppletContext().showStatus("Calculation started");
                  if (acount<2)
                  {
                              JOptionPane.showMessageDialog(wTabbedPane, "Invalid
data format!", "Warning!", JOptionPane.WARNING_MESSAGE);
                  }
                  else
if(acount<(ARABS.class2*(ARABS.class0)+ARABS.class2/2*ARABS.class0*ARABS.class4))
                  {
```

```
                        acount=((acount/ARABS.class0)/(1+ARABS.class4));
                       if (acount<1)
                       {
                             JOptionPane.showMessageDialog(wTabbedPane, "Number of
factors (M) exceeds data!", "Warning!", JOptionPane.WARNING_MESSAGE);
                       }
                       else
                       {
                       Object[] options1 = { "  Ok  " };
                  sel1 = JOptionPane.showOptionDialog(
                  wTabbedPane, "File or TextArea doesn't contain as much data as
needed", "Warning!", JOptionPane.DEFAULT_OPTION,
                  JOptionPane.WARNING_MESSAGE, null, options1, options1[0]);
                       if ( options1[sel1] == "  Ok  " )
                       {
                             if (acount<=p)
                              {
                                   p=  acount-1;
                              }
                              Object[] options2 = { "  Yes  ", " Cancel " };
                       sel2 = JOptionPane.showOptionDialog(
                             wTabbedPane, "Ajust parameters acording to data file
into:   p = "+p+ " ,  T = "+acount +"  and  continue calculation? ", "Warning!",
JOptionPane.CLOSED_OPTION,
                       JOptionPane.WARNING_MESSAGE, null, options2, options2[0]);
                             if ( options2[sel2] == "  Yes  " )
                             {
                                ARABS.class1=p;
                                w2nTextField[1].setText(""+ p);
                                ARABS.class2=acount;
                                w2nTextField[2].setText(""+acount);
                                routine();
                            }
                      }           }          }
                       else
                       routine();
            }
            else if ( wObject == w2Button)
            {
                System.out.println("Gavau stabdyma!!!!      ");
                  getAppletContext().showStatus("Calculation stopped");
                  if (Task)
                        PC.StopCount();
                  else
                        PD.StopCount();
                  w2Button.setEnabled(false);
            }
      }
 }
//=============================================================================Read
DataFile classs
public class ReadDataFile
{   //-------------------------------------------------ReadData- reads inicial data
      public int ReadData( String Dir, String IFile )
      {
            File DM = new File(Dir);
            int i=1, ret=1;
            String tem =new String();
          Bcompare = false;
```

```
            try
            {
                  URL ur2 = new URL(getDocumentBase(),IFile);
                  if ( Dir.compareTo("")!= 0)
                  {
                        ur2=DM.toURL();
                  }
                  if (IFile.compareTo("call.data")==0)
                          FileIdent=1000;
                  else
                          FileIdent=1;
                  t.setText(ur2+"\n");
                  InputStream is = ur2.openStream();
                  BufferedReader in = new BufferedReader(   new
InputStreamReader(is));
                  String line="";
                  while ((line = in.readLine()) != null)
                  {
                        if (i <=
(ARABS.class2*(ARABS.class0)+ARABS.class2/2*ARABS.class0*(ARABS.class4+ARABS.classM
)+1))
                        {
                              StringTokenizer line1 = new StringTokenizer(line);
                              int m=1;
                              if (IFile.compareTo("call.data")==0)
                              {
                                    tem=line1.nextToken();
                                    tem=line1.nextToken();
                                    tem=line1.nextToken();
                                    tem=line1.nextToken();
                              }
                              while (line1.hasMoreTokens())
                             {
                                    if(m <= (ModelAR_ABS.class0))
                                   {
                                          ModelAR_ABS.past[i]=
Double.valueOf(line1.nextToken()).doubleValue();
                                          t.append("w[ "+i+" ]
"+ModelAR_ABS.past[i]+ "\n");
                                          i++;
                                          m++;
                                          ret=i;
                                    }
                                    else
                                     tem= line1.nextToken();
                              }
                        }
                        else
if((i==(  ModelAR_ABS.class2*(ModelAR_ABS.class0)+1))&&(ModelAR_ABS.class4==0))
                        {
                              StringTokenizer line1 = new StringTokenizer(line);
                              if (IFile.compareTo("call.data")==0)
                              {
                                    tem=line1.nextToken();
                                    tem=line1.nextToken();
                                    tem=line1.nextToken();
                                    tem=line1.nextToken();
                              }
                              int m=0;
```

```
                              while (line1.hasMoreTokens())
                             {
                                    if(m <= (ModelAR_ABS.class0))
                                   {
                                          ModelAR_ABS.past[i]=
Double.valueOf(line1.nextToken()).doubleValue();
                                          i++;
                                          m++;
                                          Bcompare = true;
                                    }
                                    else
                                    tem= line1.nextToken();
                              }
                        }
                   }
                   is.close();
                   Bcompare = true; // istrinti!!!!!!!
                  }catch(Exception ex)
                  {
                  JOptionPane.showMessageDialog(wTabbedPane, "Errors occured while
reading "+IFile+ " file ", "Warning!", JOptionPane.WARNING_MESSAGE);
                        System.out.println("Error! "+ex);
                  }
            return ret-1;
      }

public int ReadDataforDiagnosis ( String Dir, String IFile )
      {
            File DM = new File(Dir);
            int i=1, ret=1, d=1;
            String tem =new String();
          Bcompare = false;
            try
            {
                  URL ur2 = new URL(getDocumentBase(),IFile);
                  if ( Dir.compareTo("")!= 0)
                  {
                        ur2=DM.toURL();
                  }
                  t.setText(ur2+"\n");
                  InputStream is = ur2.openStream();
                  BufferedReader in = new BufferedReader(   new
InputStreamReader(is));
                  String line="";
                  //System.out.println("ModelAR_ABS.class0 "+ModelAR_ABS.class0+
"\n");
                  while ((line = in.readLine()) != null)
                  {
                        if (i <= (ARABS.class2*(ARABS.class0)))
                        {
                              StringTokenizer line1 = new StringTokenizer(line);
                              int m=1;
                              while (line1.hasMoreTokens())
                             {
                                    if(m <= (ModelAR_ABS.class0))
                                   {
                                          if (m==1)
                                          {
                                             ModelAR_ABS.All[d]= line1.nextToken();
```

```
                                          ModelAR_ABS.past[i]=0;
                                        t.append("w[ "+i+" ]
"+ModelAR_ABS.past[i]+ "\n");
                                         // System.out.println("Skaitom
"+ModelAR_ABS.past[i]+ "\n");
                                         // System.out.println("Skaitom!!!
"+ModelAR_ABS.All[d]+ "\n");
                                          d++;
                                        }
                                        else
                                        {
                                             ModelAR_ABS.past[i]=
Double.valueOf(line1.nextToken()).doubleValue();
                                             //System.out.println("Skaitom
"+ModelAR_ABS.past[i]+ "\n");
                                             t.append("w[ "+i+" ]
"+ModelAR_ABS.past[i]+ "\n");

                                        }
                                        i++;
                                        m++;
                                        ret=i;
                                   }
                                   else
                                    tem= line1.nextToken();
                              }
                         }
             }
                    is.close();
                   }catch(Exception ex)
                   {
                   JOptionPane.showMessageDialog(wTabbedPane, "Errors occured while
reading "+IFile+ " file ", "Warning!", JOptionPane.WARNING_MESSAGE);
                         System.out.println("Error! "+ex);
                   }
            return ret-1;
      }
public void SimpleRead( String Dir, String IFile, JTextArea t )
      {   //-----------------------------------------------------------
SimpleRead- reads any File as text
            File DM = new File(Dir);
            try
            {
                  URL ur2 = new URL(getDocumentBase(),IFile);
                  if ( Dir.compareTo("")!= 0)
                  {
                        ur2=DM.toURL();
                  }
                  t.append(ur2+"\n");
                  InputStream is = ur2.openStream();
                  BufferedReader in = new BufferedReader(   new
InputStreamReader(is));
                  String line="";
                  while ((line = in.readLine()) != null)
                  {
                        t.append(line+"\n");
                  }
                   is.close();
            } catch(Exception ex)
```

```
            {
                 JOptionPane.showMessageDialog(wTabbedPane, "Errors occured while
reading "+IFile+ " file ", "Warning!", JOptionPane.WARNING_MESSAGE);
                  System.out.println("Error! "+ex);
            }
      }
//---------------------------------------------------------------------

}
//*************************************************************/
//*************************************************************/
public void init ()
{ //2
      // Force SwingApplet to come up in the System L&F
      String laf = UIManager.getCrossPlatformLookAndFeelClassName();
      String laf1 = UIManager.getSystemLookAndFeelClassName();
      try
      { //3
            //System.out.println("");
            //System.err.println("Started");
          UIManager.setLookAndFeel(laf);
          System.err.println("Found LookAndFeel: " + laf);
          //System.out.println("");
      }  //3
      catch (UnsupportedLookAndFeelException exc)
      { //4
          System.err.println("Warning: UnsupportedLookAndFeel: " + laf);
      }  //4
      catch (Exception exc)
      { //5
          System.err.println("Error loading " + laf + ": " + exc);
      } //5
      t2.setText(" 1  433        66        -204        -20        146        -66
-229        288        -30        113        -40        53        -65        54\n"+
                         "1   414        -25        -200        1        122
-119        -204        257        -35        122        -47        32        -20
14\n"+
"2  541        -27        -7        -103        -104        -143        -104
375        -43        91        -32        22        -104        0\n"+
"2  460        10        -1        -62        -118        -89        -85        315
-41        74        -36        44        -56        25\n"+
"2  403        -1        19        -25        -96        -49        -89        296
-25        79        -35        85        -75        39\n"+
"3  414        -152        -13        -86        -136        38        -114
255        0        33        -39        20        18        4\n"+
"3  427        -165        -23        -94        -111        24        -110
267        -8        44        -36        18        7        1\n"+
"3  420        -146        -36        -45        -133        42        -101
245        -6        38        -39        15        9        0\n"+
"4  492        -66        -264        125        -39        -76        -103
328        -69        -145        56        -91        13        0\n"+
"4  465        -79        -269        105        -97        -20        -59
317        -53        -128        49        -86        28        17\n"+
"4  463        -97        -266        107        -184        11        39
329        -52        -138        58        -73        18        -7\n"+
"5 555        -9        -23        -5        -169        -20        -148        318
55        15        -52        54        36        35\n"+
"5  557        -7        6        -34        -168        -57        -151        320
70        12        -43        43        29        42\n"+
```

```
"5   586        10        40        -39        -155        8        -113        322
68        10        -44        64        40        28\n"+
"6  507        109        26        5        -164        -65        -132        299
-99        -9        -5        26        -41        -40\n"+
"6  537        76        22        9        -139        -82        -144        308
-78        -11        -11        32        -6        -65\n"+
"6  521        86        36        14        -126        -49        -129        312
-78        -26        4        37        -15        -49\n"+
"7  378        -88        -129        21        48        -15        -138
236        96        88        80        16        32        20\n"+
"7  384        -111        -112        37        43        -8        -133
245        95        66        80        6        15        12\n"+
"7  421        -134        -124        48        16        -38        -129
263        115        94        90        25        31        26\n"+
"8  566        57        47        -78        -188        -111        -81
424        -12        -24        -43        42        16        -62\n"+
"8  571        32        35        -119        -182        -165        -42
440        -53        -20        -63        48        -45        -30\n"+
"8  588        41        49        -21        -206        -77        -174
457        -33        1        -49        45        -7        -53\n"+
"9  553        -57        44        -92        -133        -70        -149
330        67        -32        -44        52        -18        -60\n"+
"9  544        -58        43        -81        -111        -51        -129
339        71        -24        -49        40        4        -44\n"+
"9  551        -63        47        -104        -121        -75        -113
352        56        -38        -50        43        -30        -76" );

//-------------------------------------------------------------
      out = new OutputFrame("Results...");
      getContentPane().setLayout(new BorderLayout());
      setSize(700, 450);
      ParamWidth = 700;
      ParamHeight = 450;

      ModelAR_ABS = new ARABS();
      Model = "AR-ABS";
      IFile=DefaultFile;
      String[] Inputfieldnames = new String[ModelAR_ABS.InputsNumber];
      Inputfieldnames[0] = " Number of factors (M) : ";
      Inputfieldnames[1] = " AR parameter (p) :";
      Inputfieldnames[2] = " Number of data entrees (T) :";
      Inputfieldnames[3] = " Indikator for one p evaluation (0), increasing
p(1) :";
      Inputfieldnames[4] = " Indikator for one step evaluation(0), multistep(1) :";
      Inputfieldnames[5]= "  Indikator for multi days prognose (1) :";
      Inputfieldnames[6]=  " Number of multi days prognose repetitions K :";

//--------------- MODEL -------------------------
                int l = 25;
                g.add(rb1); g.add(rb2);
                w1Panel = new JPanel();
                w1Panel.setLayout(null);

                w11Label = new JLabel("Model to expirement and apply-    AR-
ARS");
                w11Label.setBounds(15, 25, ParamWidth-200, height);
                w11Label.setOpaque(true);
                w11Label.setBackground(new Color(161, 161, 161));
                w11Label.setForeground(new Color(66, 75, 145));
```

```
w1Panel.add(w11Label);

Left = ParamWidth-200+30;

w12Label = new Choice ();
w12Label.addItem("Prognosis");
w12Label.addItem("Diagnosis");
w12Label.addItemListener(new IL1());
w12Label.setBounds(Left, 25, width, height);
w1Panel.add(w12Label);
rb1.setBounds(25, (33*1)+l+3,ParamWidth-200, height);
rb1.setOpaque(true);
rb1.addActionListener(new  ActionListener()
{
      public void actionPerformed(ActionEvent e)
      {
            if (e.getSource()==rb1)
            {
                  w11TextField.setEnabled(true);
                w11Button.setEnabled(true);
                  w12Button.setEnabled(true);
                  t1.setEditable(false);
                  t1.setVisible(false);
                  //wPopupMenut1.setOpaque(false);
            }
      }
});
w1Panel.add(rb1);

rb2.setBounds(25, (33*2)+l-3,ParamWidth-200, height);
rb2.setOpaque(true);
rb2.addActionListener(new  ActionListener()
{
      public void actionPerformed(ActionEvent e)
      {
            if (e.getSource()==rb2)
            {
                  w11TextField.setEnabled(false);
                w11Button.setEnabled(false);
                  w12Button.setEnabled(false);
                  t1.setEditable(true);
                  t1.setVisible(true);
                  //wPopupMenut1.setEnabled(true);
            }
      }
});
w1Panel.add(rb2);

w13Label = new JLabel("  Select a data file ");
w13Label.setBounds(15, (33*3)+l,ParamWidth-200, height);
w13Label.setOpaque(true);
w13Label.setBackground(new Color(161, 161, 161));
w13Label.setForeground(new Color(66, 75, 145));
w1Panel.add(w13Label);

w11Button = new JButton("Default");
w11Button.setToolTipText("Default file - armatest");
w11Button.setBounds(Left,(33*3)+l, width, height);
w11Button.addActionListener(new  ActionListener()
```

```
                {
                      public void actionPerformed(ActionEvent e)
                      {
                            if (e.getSource()==w11Button)
                            {
                                  w11TextField.setText(DefaultFile);
                                  IFile=DefaultFile;
                                  Dir= "";
                                  getAppletContext().showStatus(IFile);
                            }
                      }
                });
                w1Panel.add(w11Button);

                w11TextField = new JTextField(DefaultFile);
          w11TextField.setBounds(15, (33*4)+l, ParamWidth-200, height);
                w11TextField.addFocusListener(new FocusListener()
                {
                      public void focusGained(FocusEvent e)
                {
                }
                public void focusLost(FocusEvent e)
                    {
                             IFile = ((JTextField)e.getSource()).getText();
                             w11TextField.setText(IFile);
                             Dir= "";
                             getAppletContext().showStatus(IFile);
                    }
               } );
                w1Panel.add(w11TextField);

                w12Button = new JButton("Browse...");
                w12Button.setBounds(Left,(33*4)+l, width, height);
                w12Button.addActionListener(wFActionListener);
                w1Panel.add(w12Button);
                w1ScrollPane = new
JScrollPane(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
                w1ScrollPane.setBounds(15, 33*5+l+2, ParamWidth-35,
ParamHeight-260);
                w1ScrollPane.setViewportView(t1);
                w1ScrollPane.setAutoscrolls(true);
                t1.setEditable(false);
                t1.setVisible(false);
                w1Panel.add(w1ScrollPane);

//------- Inputs ------------------------------------------------------
                w2Panel = new JPanel();
                w2Panel.setLayout(null);

                w21Label = new JLabel("  Property ");
                w21Label.setBounds(15, 25, ParamWidth-200, height);
                w21Label.setOpaque(true);
                w21Label.setBackground(new Color(161, 161, 161));
                w21Label.setForeground(new Color(66, 75, 145));
                w2Panel.add(w21Label);

                Left = ParamWidth-200+30;
                w22Label = new JLabel("  Value ");
```

```
                  w22Label.setBounds(Left, 25, width, height);
                  w22Label.setOpaque(true);
                  w22Label.setBackground(new Color(161, 161, 161));
                  w22Label.setForeground(new Color(66, 75, 145));
                  w2Panel.add(w22Label);
                  w200Label = new JLabel("Confidence value: ");
                  w200Label.setBounds(25, (33*(1+1))+l, ParamWidth-200, height);
                  w200Label.setVisible(false);
                w2Panel.add(w200Label);
                  w20Label = new JLabel("Enter examples to classify below: ");
                  w20Label.setBounds(25, (33*(3+1))+l, ParamWidth-200, height);
                  w20Label.setVisible(false);
                  w2Panel.add(w20Label);
                  w2nTextField[0] = new JTextField(""+ModelAR_ABS.class0);
                  w2nTextField[1] = new JTextField(""+ModelAR_ABS.class1);
                  w2nTextField[2] = new JTextField(""+ModelAR_ABS.class2);
                w2nTextField[3] = new JTextField(""+ModelAR_ABS.class3);
                  w2nTextField[4] = new JTextField(""+ModelAR_ABS.class4);
                  w2nTextField[5] = new JTextField(""+ModelAR_ABS.classM);
                  w2nTextField[6] = new JTextField(""+ModelAR_ABS.classK);
                  for (int i=0; i<7; i++)
                {
                        w2nLabel[i] = new JLabel(Inputfieldnames[i]);
                        w2nLabel[i].setBounds(25, (33*(i+1))+l, ParamWidth-200,
height);
                        w2Panel.add(w2nLabel[i]);
                        w2nTextField[i].setBounds(Left,(33*(i+1))+l, width,
height);
                w2nTextField[i].addFocusListener(wTSFocusListener);
                  w2Panel.add(w2nTextField[i]);
                  }
                  w2nTextField[0].setToolTipText("M value must be 1 or greater ");
                  w2nTextField[1].setToolTipText("p value must be less than T ");
                  w2nTextField[2].setToolTipText("T value must be greater than p");
            w2nTextField[3].setToolTipText("1-p increasing; 0-p constant");
                  w2nTextField[4].setToolTipText("1-multi step; 0-one step ");
                  w2ScrollPane = new
JScrollPane(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
                  w2ScrollPane.setBounds(15, 33*5+l+2, ParamWidth-35,
ParamHeight-260);
                  w2ScrollPane.setViewportView(t2);
                  w2ScrollPane.setAutoscrolls(true);
                  t2.setEditable(true);
                  w2ScrollPane.setVisible(false);
                  w2Panel.add(w2ScrollPane);
 //--------------- COUNT ----------------------------
            w3Panel = new JPanel();
                  w3Panel.setLayout(null);

                  w31Label = new JLabel(Model);
                  w31Label.setBounds(15, 25, ParamWidth-35, height);
                  w31Label.setOpaque(true);
                  w31Label.setBackground(new Color(161, 161, 161));
                  w31Label.setForeground(new Color(66, 75, 145));
                  w3Panel.add(w31Label);

                  int LeftWidth = (ParamWidth-45)/4;
                  w1Button = new JButton("Calculate");
```

```
                w1Button.setToolTipText("Start Counting ");
                w1Button.setBounds(25, ParamHeight-80, LeftWidth-10, height);
                w1Button.addActionListener(new Stop());
                w1Button.setEnabled(true);
                w3Panel.add(w1Button);
                w2Button = new JButton("Stop");
                w2Button.setToolTipText("Stop Counting ");
                w2Button.setBounds(25+LeftWidth, ParamHeight-80, LeftWidth-10,
height);
                w2Button.addActionListener( new Stop());
                w2Button.setEnabled(false);
                w3Panel.add(w2Button);
          w3Button = new JButton("Graphic");
                w3Button.setToolTipText("View Graphic ");
                w3Button.setBounds(25+LeftWidth*2, ParamHeight-80, LeftWidth-10,
height);
                w3Button.setEnabled(false);
                w3Button.addActionListener(new Grafic());
                w3Panel.add(w3Button);
                w4Button = new JButton("Detaled rezults");
                 w4Button.setToolTipText("View detaled rezults");
                w4Button.setBounds(25+LeftWidth*3, ParamHeight-80, LeftWidth-10,
height);
                w4Button.setEnabled(true);
                w4Button.addActionListener(new  ActionListener()
                {
                      public void actionPerformed(ActionEvent e)
                      {
                            if (e.getSource()==w4Button)
                            {
                                  if (Task)
                                  {
                                        out.setSize( 400, 450 );
                                        out.setVisible( true );
                            }
                                  else
                                  {
                                        out.setSize( 400, 450 );
                                        out.setVisible( true );
                                  }
                            }
                                  /*t.setText("");
                                  if ( rb1.isSelected())
                                  {
                                        ReadDataFile  RDF= new ReadDataFile();
                                        RDF.SimpleRead(Dir, IFile,t);
                                        getAppletContext().showStatus(IFile);
                                  }
                                  else if ( rb2.isSelected())
                                  {
                                        t.setText(t1.getText());
                                  }*/
                            }
                });

                w3Panel.add(w4Button);
                //--- COUNT ScrollPane
----------------------------------------
                wScrollPane = new
```

```
JScrollPane(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
                  wScrollPane.setBounds(15, 33+l, ParamWidth-35, ParamHeight-160);
                  wScrollPane.setViewportView(t);
                  wScrollPane.setAutoscrolls(true);

                  //wScrollPane.ScroolBar.setEnabled(true);
                  t.setEditable(false);
                  w3Panel.add(wScrollPane);
            //--- COUNT ProgressBar ----------------------------------------

                  wProgressBar.setBounds(15, ParamHeight-98, ParamWidth-35, 10);
                  wProgressBar.setValue(0);
                  wProgressBar.setBorderPainted(true);
                  w3Panel.add(wProgressBar);
            System.err.println(" Added Count");
//---------------------------------------------------------------

      wTabbedPane.addTab(Name1, w1Panel);
      wTabbedPane.addTab(Name2, w2Panel);
      wTabbedPane.addTab(Name3, w3Panel);

            wPopupMenut1 = new JPopupMenu();
          //wMenuItemt1 = new JMenuItem();
          //wMenuItemt1.setText("Copy");
            //wPopupMenut1.add(wMenuItemt1);
            //wMenuItemt2 = new JMenuItem();
            //wMenuItemt2.setText("Paste");
        //wPopupMenut1.add(wMenuItemt2);
            //wMenuItemt3 = new JMenuItem();
            //wMenuItemt3.setText("Delete");
            //wPopupMenut1.add(wMenuItemt3);
            //wPopupMenut1.addSeparator();
            wMenuItemt4 = new JMenuItem();
            wMenuItemt4.setText("Select All");
        wPopupMenut1.add(wMenuItemt4);
            wMenuItemt4.addActionListener(new Popup1());

            wPopupMenu1 = new JPopupMenu();
          wMenuItem1 = new JMenuItem();
          wMenuItem1.setText("Hellp..");
        wPopupMenu1.add(wMenuItem1);
        wMenuItem1.addActionListener(new Popup1());

          wPopupMenu2 = new JPopupMenu();
          wMenuItem2 = new JMenuItem();
          wMenuItem2.setText("Hellp..");
        wPopupMenu2.add(wMenuItem2);
        wMenuItem2.addActionListener(new Popup1());

            wPopupMenu3 = new JPopupMenu();
          wMenuItem3 = new JMenuItem();
          wMenuItem3.setText("Hellp..");
        wPopupMenu3.add(wMenuItem3);
        wMenuItem3.addActionListener(new Popup1());
        t.addMouseListener(wPMouseAdapter);
            t1.addMouseListener(wPMouseAdapter);
            w1Panel.addMouseListener(wPMouseAdapter);
            w2Panel.addMouseListener(wPMouseAdapter);
```

```
            w3Panel.addMouseListener(wPMouseAdapter);
          getContentPane().add(wTabbedPane);
        System.err.println("Finished");
  }
}
```