Routen-Ersteller Quellcode:

GUI.java:

```
import java.awt.EventQueue;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.math.RoundingMode;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Iterator;
import javax.swing.DefaultListModel;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JList;
import javax.swing.JOptionPane;
import javax.swing.ListSelectionModel;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.JEditorPane;
import java.awt.Font;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.JPopupMenu;
import java.awt.Component;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

public class GUI {

    static ArrayList<Tankstelle> alleTankstellen=new ArrayList<Tankstelle>();
    static DefaultListModel<Tankstelle> model=new DefaultListModel<Tankstelle>();
    DefaultListModel<Tankstelle> alle=new DefaultListModel<Tankstelle>();
    DefaultListModel<Tankstelle> suche=new DefaultListModel<Tankstelle>();
    DefaultListModel<Tankstelle> letzte=new DefaultListModel<Tankstelle>();
```

```
	static JList<Tankstelle> list = new JList<Tankstelle>();
	private JFrame frame;
	static Route route=new Route();
	private JTextField textField;
	ArrayList<String> date=new ArrayList<String>();
	private JTextField txtJahr;
	private JTextField txtMonat;
	private JTextField txtTag;
	private JTextField txtStunde;
	private static JTextField txtFile;

	/**
	 * Launch the application.
	 * @throws IOException
	 */
	static void laden() {
		route.getRoute().clear();
		FileReader fr=null;
		BufferedReader br=null;
		try {
			String path=new File("").getAbsolutePath();
			fr=new
FileReader(path+"/Eingabedaten/Fahrzeugrouten/"+txtFile.getText()+".csv");
			br=new BufferedReader(fr);
			String s=null;
			String tmpS="";
			while((s=br.readLine()) !=null){
				tmpS+=s+";";
			}
			String[] parts =tmpS.split(";");
			for (int i=1; i<=parts.length; i++) {
				if(i%2==0) {

route.getRoute().add(alleTankstellen.get(Integer.parseInt(parts[i])-1));
				}
			}
			JOptionPane.showMessageDialog(null, "Route geladen!");
			list.setModel(model);
			Iterator<Tankstelle> it=route.getRoute().iterator();
			model.clear();
			while(it.hasNext()) {
				model.addElement(it.next());
			}


		} catch (FileNotFoundException e) {
```

```
                JOptionPane.showMessageDialog(null, "Route nicht gefunden!");
                System.err.println("File not found.");
        } catch (IOException e) {
                JOptionPane.showMessageDialog(null, "Fehler!");
                System.err.println("Error!");
                try {
                        br.close();
                        fr.close();
                } catch (IOException e1) {
                }
        }
    }


    static void einlesen() {
        FileReader fr=null;
        BufferedReader br=null;
        try {
                String path=new File("").getAbsolutePath();
                fr=new FileReader(path+"/Eingabedaten/Tankstellen.csv");
                br=new BufferedReader(fr);
                String s=null;
                while((s=br.readLine()) !=null){
                        String[] parts=s.split(";");
                        alleTankstellen.add(new Tankstelle(Integer.parseInt(parts[0]),
parts[1], parts[2], parts[3]+" "+parts[4]+" "+parts[5]+" "+parts[6],
Double.parseDouble(parts[7]), Double.parseDouble(parts[8])));
                }

        } catch (FileNotFoundException e) {
                System.err.println("File not found.");
        } catch (IOException e) {
                System.err.println("Error!");
                try {
                        br.close();
                        fr.close();
                } catch (IOException e1) {
                }
        }
    }

    void änderDatum(double zeit) {
        int j=Integer.parseInt(date.get(0));
        int m=Integer.parseInt(date.get(1));
        int t=Integer.parseInt(date.get(2));
        int std=Integer.parseInt(date.get(3));
```

```
            int min=Integer.parseInt(date.get(4));
            int sek=Integer.parseInt(date.get(5));

            sek+=(zeit*3600);

            if(sek>=60) {min+=sek/60; sek=sek%60;}

    //    min+=(zeit*60);

            if(min>=60) {std+=min/60; min=min%60;}

    //    std+=zeit;

            if(std>=24) {t+=std/24; std=std%24;}
            int intm=(int) m;
            t+=zeit/24;
            if((intm==1||intm==3||intm==5||intm==7||intm==8||intm==10||intm==12)&&t>31)
{m+=t/31; t=t%31;}
            else if((intm==4||intm==6||intm==9||intm==11)&&t>30)
{m+=t/30; t=t%30;}
            else if(intm==2 && t>28 &!(j%4==0&&j%100!=0&&j%400==0))       {m+=t/28;
t=t%28;}
            else if (intm==2 && t>29&&(j%4==0&&j%100!=0&&j%400==0))      {m+=t/29;
t=t%29;}

            int tmp=0;

            if(intm==1||intm==3||intm==5||intm==7||intm==8||intm==10||intm==12) {tmp=31;}
            else if(intm==4||intm==6||intm==9||intm==11) {tmp=30;}
            else if(intm==2 &!(j%4 == 0 && (j%100 != 0 || j%400 == 0))) {tmp=28;}
            else if (intm==2&&(j%4 == 0 && (j%100 != 0 || j%400 == 0))) {tmp=29;}
            m+=zeit/(24*tmp);
            if(m>12) {j+=m/12; m=m%12;}

            int tag=(int) t;
            int jahr=(int) j;
            int monat=(int) m;
            int stunde=(int) std;
            int minute=(int) min;
            int sekunde=(int) sek;

            date.clear();
            date.add(Integer.toString(jahr));
            date.add(Integer.toString(monat));
            date.add(Integer.toString(tag));
            date.add(Integer.toString(stunde));
```

```
		date.add(Integer.toString(minute));
		date.add(Integer.toString(sekunde));

	}

	void auslesen() {
		route.berechneZeit();
		FileWriter fw=null;
		BufferedWriter bw=null;
		String s="3\n";
		Iterator<Tankstelle> i=route.getRoute().iterator();
		Tankstelle tmp=null;
		ArrayList<Double> zeit=route.getZeiten();
		double[] zeiten=new double[zeit.size()];
		for(int b=0; b<zeiten.length; b++) {
			zeiten[b]=zeit.get(b);
		}
		int b=0;
		double n=zeiten[0];
		while(i.hasNext()){
			änderDatum(n);
			b++;
			try {
			n=zeiten[b];
			}catch(Exception e) {
				System.err.println("err");
			}
			tmp=i.next();
			String m=date.get(1);
			String t=date.get(2);
			String std=date.get(3);
			String min=date.get(4);
			String sek=date.get(5);
			if(Integer.parseInt(m)<10) {
				m="0"+m;
			}
			if(Integer.parseInt(t)<10){
				t="0"+t;
			}
			if(Integer.parseInt(std)<10) {
				std="0"+std;
			}
			if(Integer.parseInt(min)<10) {
				min="0"+min;
			}
			if(Integer.parseInt(sek)<10) {
```

```
                        sek="0"+sek;
                }
                s+=date.get(0)+"-"+m+"-"+t+" "+std+":"+min+":"+sek+"+00;"+tmp.getId()+"\n";
        }
        try {
                String path=new File("").getAbsolutePath();
                fw=new FileWriter(path+"/Eingabedaten/Fahrzeugrouten/"+
txtFile.getText()+".csv");
                bw=new BufferedWriter(fw);
                bw.write(s);
                JOptionPane.showMessageDialog(null, "Route gespeichert.");

        } catch (IOException e) {JOptionPane.showMessageDialog(null, "Ein Fehler ist
aufgetreten");}
        finally {
                try {
                        if(bw!=null){
                                bw.flush();
                                bw.close();
                        }
                        if(fw!=null) {
                                fw.close();
                        }
                }catch(IOException ex) {
                        JOptionPane.showMessageDialog(null, "Ein Fehler ist aufgetreten");
                }
        }

   }



   public static void main(String[] args) {
        einlesen();
        EventQueue.invokeLater(new Runnable() {
                public void run() {
                        try {
                                GUI window = new GUI();
                                window.frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
   }
```

```
/**
 * Create the application.
 */
public GUI() {
	initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
	frame = new JFrame();
	frame.setResizable(false);
	frame.setBounds(100, 100, 1024, 701);
	frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	frame.getContentPane().setLayout(null);

	JScrollPane scrollPane = new JScrollPane();
	scrollPane.setBounds(10, 11, 260, 592);
	frame.getContentPane().add(scrollPane);

	list.setFont(new Font("Arial", Font.PLAIN, 11));
	list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
	list.setVisibleRowCount(15300);
	scrollPane.setViewportView(list);
	Iterator<Tankstelle> i=alleTankstellen.iterator();
	while(i.hasNext()) {
		alle.addElement(i.next());
	}
	list.setModel(alle);

	JScrollPane scrollPane_1 = new JScrollPane();
	scrollPane_1.setBounds(278, 475, 720, 155);
	frame.getContentPane().add(scrollPane_1);

	JEditorPane editorPane = new JEditorPane();
	scrollPane_1.setViewportView(editorPane);
	editorPane.setFont(new Font("Calibri", Font.PLAIN, 18));
	editorPane.setEditable(false);

	JButton btnAddToRoute = new JButton("Hinzuf\u00FCgen");
	btnAddToRoute.addActionListener(new ActionListener() {
		public void actionPerformed(ActionEvent arg0) {
			//Nur benutzen wenn eine Tankstelle nur einmal vorkommen soll
			if(route.getRoute().contains(list.getSelectedValue())) {
```

```
                                JOptionPane.showMessageDialog(null, "Tankstelle ist bereits
in der Route vorhanden.");
                        }else if(list.getSelectedIndex()!=-1){
                                route.addTankstelle(list.getSelectedValue());
                                if(!route.checkDistanzen())
{JOptionPane.showMessageDialog(null, "Die Tankstellen liegen zu weit auseinander und
sind nicht erreichbar!"); route.removeTankstelle(list.getSelectedValue());}
                                else {JOptionPane.showMessageDialog(null, "Tankstelle
wurde zur Route hinzugefügt!");}
                        }
                        else {JOptionPane.showMessageDialog(null, "Bitte eine Tankstelle
aus der Liste wählen!");}

                }
        });
        btnAddToRoute.setBounds(280, 9, 114, 23);
        frame.getContentPane().add(btnAddToRoute);

        JButton btnShowRoute = new JButton("Zeige Route");
        btnShowRoute.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        list.setModel(model);
                        Iterator<Tankstelle> it=route.getRoute().iterator();
                        model.clear();
                        while(it.hasNext()) {
                                model.addElement(it.next());
                        }
                }
        });
        btnShowRoute.setBounds(280, 43, 114, 23);
        frame.getContentPane().add(btnShowRoute);

        JButton btnAnalyse = new JButton("Analyse");
        btnAnalyse.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        DecimalFormat df=new DecimalFormat("#.###");
                        df.setRoundingMode(RoundingMode.CEILING);
                        double distanz=route.berechneDistanzGesamt();
                        double verbrauch=route.berechneVerbrauchGesamt();
                        String anzeige="Distanz: "+df.format(distanz)+"km\n"+"Verbrauch:
"+df.format(verbrauch)+" Liter\n";
                        ArrayList<Double> d=route.berechneDistanzen();
                        Iterator<Double> it=d.iterator();
                        Iterator<Tankstelle> t=route.getRoute().iterator();
                        Tankstelle ta;
                        if(t.hasNext()) {
```

```
                        ta=t.next();
                    }else {ta=null;}
                    Tankstelle ta2;
                    while(it.hasNext()&&t.hasNext()) {
                        ta2=ta;
                        ta=t.next();
                        double tmp=it.next();
                        anzeige+="Von "+ta2.getId()+" bis "+ta.getId()+": Distanz:
"+df.format(tmp)+"km, Verbrauch: "+df.format((tmp/100)*5.6)+" Liter\n";
                    }

                    editorPane.setText(anzeige);
                }
            });
            btnAnalyse.setBounds(280, 77, 238, 23);
            frame.getContentPane().add(btnAnalyse);

            JButton btnNewButton = new JButton("Entfernen");
            btnNewButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    route.getRoute().remove(list.getSelectedValue());
                    JOptionPane.showMessageDialog(null, "Tankstelle aus der Route
entfernt!");
                }
            });
            btnNewButton.setBounds(404, 9, 114, 23);
            frame.getContentPane().add(btnNewButton);

            JButton btnNewButton_1 = new JButton("Route l\u00F6schen");
            btnNewButton_1.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    route.getRoute().clear();
                    JOptionPane.showMessageDialog(null, "Route gelöscht!");
                    list.setModel(alle);
                }
            });
            btnNewButton_1.setBounds(280, 441, 132, 23);
            frame.getContentPane().add(btnNewButton_1);

            JButton btnShowAll = new JButton("Zeige alle");
            btnShowAll.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    list.setModel(alle);
                }
            });
            btnShowAll.setBounds(404, 43, 114, 23);
```

```
		frame.getContentPane().add(btnShowAll);

		textField = new JTextField();
		textField.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent e) {
				textField.selectAll();
			}
		});
		textField.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				if(list.getModel()==suche) {list.setModel(letzte);}
				if(list.getModel()==alle) {
					letzte=alle;
					suche.clear();
					Iterator<Tankstelle> it=alleTankstellen.iterator();
					String s=textField.getText().toUpperCase();
					while(it.hasNext()) {
						Tankstelle tmp=it.next();

if(Integer.toString(tmp.getId()).toUpperCase().contains(s) ||
							tmp.getAdresse().toUpperCase().contains(s) ||
							tmp.getName().toUpperCase().contains(s) ||
							tmp.getFirma().toUpperCase().contains(s)) {
							suche.addElement(tmp);
						}
					}
					list.setModel(suche);
				}else if(list.getModel()==route.getRoute()) {
					letzte=model;
					suche.clear();
					Iterator<Tankstelle> it=route.getRoute().iterator();
					String s=textField.getText().toUpperCase();
					while(it.hasNext()) {
						Tankstelle tmp=it.next();

if(Integer.toString(tmp.getId()).toUpperCase().contains(s) ||
							tmp.getAdresse().toUpperCase().contains(s) ||
							tmp.getName().toUpperCase().contains(s) ||
							tmp.getFirma().toUpperCase().contains(s)) {
							suche.addElement(tmp);
						}
					}
					list.setModel(suche);
				}
```

```
		}
	});
	textField.setBounds(10, 610, 165, 20);
	frame.getContentPane().add(textField);
	textField.setColumns(10);

	JButton btnSearch = new JButton("Search");
	btnSearch.addActionListener(new ActionListener() {
		public void actionPerformed(ActionEvent e) {
			if(list.getModel()==suche) {list.setModel(letzte);}
			if(list.getModel()==alle) {
				letzte=alle;
				suche.clear();
				Iterator<Tankstelle> it=alleTankstellen.iterator();
				String s=textField.getText().toUpperCase();
				while(it.hasNext()) {
					Tankstelle tmp=it.next();

if(Integer.toString(tmp.getId()).toUpperCase().contains(s) ||
						tmp.getAdresse().toUpperCase().contains(s) ||
						tmp.getName().toUpperCase().contains(s) ||
						tmp.getFirma().toUpperCase().contains(s)) {
						suche.addElement(tmp);
					}
				}
				list.setModel(suche);
			}else if(list.getModel()==route.getRoute()) {
				letzte=model;
				suche.clear();
				Iterator<Tankstelle> it=route.getRoute().iterator();
				String s=textField.getText().toUpperCase();
				while(it.hasNext()) {
					Tankstelle tmp=it.next();

if(Integer.toString(tmp.getId()).toUpperCase().contains(s) ||
						tmp.getAdresse().toUpperCase().contains(s) ||
						tmp.getName().toUpperCase().contains(s) ||
						tmp.getFirma().toUpperCase().contains(s)) {
						suche.addElement(tmp);
					}
				}
				list.setModel(suche);
			}

		}
	});
```

```
btnSearch.setBounds(181, 607, 89, 23);
frame.getContentPane().add(btnSearch);

txtJahr = new JTextField();
txtJahr.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent arg0) {
		txtJahr.selectAll();
	}
});
txtJahr.setText("Jahr");
txtJahr.setBounds(892, 44, 86, 20);
frame.getContentPane().add(txtJahr);
txtJahr.setColumns(10);

txtMonat = new JTextField();
txtMonat.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent e) {
		txtMonat.selectAll();
	}
});
txtMonat.setText("Monat");
txtMonat.setBounds(892, 75, 86, 20);
frame.getContentPane().add(txtMonat);
txtMonat.setColumns(10);

txtTag = new JTextField();
txtTag.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseClicked(MouseEvent e) {
		txtTag.selectAll();
	}
});
txtTag.setText("Tag");
txtTag.setBounds(892, 106, 86, 20);
frame.getContentPane().add(txtTag);
txtTag.setColumns(10);

JLabel lblDatum = new JLabel("Datum");
lblDatum.setFont(new Font("Tahoma", Font.BOLD, 13));
lblDatum.setBounds(836, 11, 60, 16);
frame.getContentPane().add(lblDatum);

txtStunde = new JTextField();
txtStunde.addMouseListener(new MouseAdapter() {
```

```
		@Override
		public void mouseClicked(MouseEvent e) {
			txtStunde.selectAll();
		}
	});
	txtStunde.setText("Stunde");
	txtStunde.setBounds(892, 137, 86, 20);
	frame.getContentPane().add(txtStunde);
	txtStunde.setColumns(10);

	JButton btnApply = new JButton("Eintragen");
	btnApply.addActionListener(new ActionListener() {
		public void actionPerformed(ActionEvent arg0) {
			int jahr=0;
			int monat=0;
			int tag=0;
			int stunde=0;
			try {
			jahr=Integer.parseInt(txtJahr.getText());
			monat=Integer.parseInt(txtMonat.getText());
			tag=Integer.parseInt(txtTag.getText());
			stunde=Integer.parseInt(txtStunde.getText());
			}catch(NumberFormatException e)
{JOptionPane.showMessageDialog(null, "Bitte nur Zahlen angeben!");}
			if((1970<=jahr) && (monat>=1 && monat<=12) &&
(stunde>=0&&stunde<=23)) {
				if(tag>=1&&((monat==1&&tag<=31)||(monat==2&& (tag<=28 ||
(tag==29&&(jahr%4 == 0 && (jahr%100 != 0 || jahr%400 ==
0)))))||(monat==3&&tag<=31)||(monat==4&&tag<=30)||(monat==5&&tag<=31)||(monat==6&&
tag<=30)||(monat==7&&tag<=31)||(monat==8&&tag<=31)||(monat==9&&tag<=30)||(monat==
10&&tag<=31)||(monat==11&&tag<=30)||(monat==12&&tag<=31))){
				date.clear();
				date.add(txtJahr.getText());
				date.add(txtMonat.getText());
				date.add(txtTag.getText());
				date.add(txtStunde.getText());
				date.add("00");
				date.add("00");
				date.add("00");
				JOptionPane.showMessageDialog(null, "Datum
gespeichert!");
				}else {JOptionPane.showMessageDialog(null, "Bitte nur
gültige Daten angeben!");}
			}else {JOptionPane.showMessageDialog(null, "Bitte nur gültige Daten
angeben!");}
```

```
		}
	});
	btnApply.setBounds(836, 168, 142, 23);
	frame.getContentPane().add(btnApply);

	JButton btnSpeichern = new JButton("Route speichern");
	btnSpeichern.addActionListener(new ActionListener() {
		public void actionPerformed(ActionEvent e) {
			if(!date.isEmpty()) {
			auslesen();
			}else {JOptionPane.showMessageDialog(null, "Bitte vorher ein Datum
eingeben!");}
		}
	});
	btnSpeichern.setBounds(820, 441, 178, 23);
	frame.getContentPane().add(btnSpeichern);

	txtFile = new JTextField();
	txtFile.addMouseListener(new MouseAdapter() {
		@Override
		public void mouseClicked(MouseEvent e) {
			txtFile.selectAll();
		}
	});
	txtFile.setText("Route");
	txtFile.setBounds(724, 442, 86, 20);
	frame.getContentPane().add(txtFile);
	txtFile.setColumns(10);

	JLabel lblJahr = new JLabel("Jahr");
	lblJahr.setBounds(836, 47, 46, 14);
	frame.getContentPane().add(lblJahr);

	JLabel lblMonat = new JLabel("Monat");
	lblMonat.setBounds(836, 81, 46, 14);
	frame.getContentPane().add(lblMonat);

	JLabel lblTag = new JLabel("Tag");
	lblTag.setBounds(836, 109, 46, 14);
	frame.getContentPane().add(lblTag);

	JLabel lblNewLabel = new JLabel("Stunde");
	lblNewLabel.setBounds(836, 140, 46, 14);
	frame.getContentPane().add(lblNewLabel);

	JLabel lblDateiname = new JLabel("Dateiname");
```

```
		lblDateiname.setBounds(724, 425, 86, 14);
		frame.getContentPane().add(lblDateiname);

		JButton btnNewButton_2 = new JButton("Route laden");
		btnNewButton_2.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				laden();

			}
		});
		btnNewButton_2.setBounds(589, 441, 125, 23);
		frame.getContentPane().add(btnNewButton_2);

		list.addListSelectionListener(new ListSelectionListener() {
			public void valueChanged(ListSelectionEvent arg0) {
				if(list.getSelectedIndex()!=-1) {
					editorPane.setText(list.getSelectedValue().information());

				}
			}
		});


	}
	private static void addPopup(Component component, final JPopupMenu popup) {
		component.addMouseListener(new MouseAdapter() {
			public void mousePressed(MouseEvent e) {
				if (e.isPopupTrigger()) {
					showMenu(e);
				}
			}
			public void mouseReleased(MouseEvent e) {
				if (e.isPopupTrigger()) {
					showMenu(e);
				}
			}
			private void showMenu(MouseEvent e) {
				popup.show(e.getComponent(), e.getX(), e.getY());
			}
		});
	}
}
```

Route.java

```
import java.util.ArrayList;
import java.util.Iterator;

public class Route {
   ArrayList<Tankstelle> route=new ArrayList<Tankstelle>();
   ArrayList<Double> zeiten=new ArrayList<Double>();



   public void addTankstelle(Tankstelle t){
         route.add(t);
   }
   public void removeTankstelle(Tankstelle t) {
         route.remove(t);
   }
   public void clearRoute() {
         route.clear();
   }
   public ArrayList<Tankstelle> getRoute(){
         return this.route;
   }

   double berechneDistanz(Tankstelle a, Tankstelle b) {
        double
d=6378.388*Math.toRadians(Math.acos(Math.sin(a.getLat())*Math.sin(b.getLat())
               +Math.cos(a.getLat())*Math.cos(b.getLat())*Math.cos(b.getLon()-a.getLon())));
         return d;
   }

   boolean checkDistanzen() {
         boolean b=true;
         Iterator<Tankstelle> it=route.iterator();
         double d=0.00;
         Tankstelle t1;
         Tankstelle tmp;
         if(it.hasNext()) {t1=it.next();}
         else {t1=null;}
         while(it.hasNext()) {
               tmp=t1;
               t1=it.next();
               d=berechneDistanz(tmp, t1);
               if(d>((100/5.6)*3)){b=false; break;}
         }
```

```
        return b;
    }
    ArrayList<Double> getZeiten(){
        return this.zeiten;
    }

    void berechneZeit(){
        zeiten.clear();
        ArrayList<Double> distanzen=berechneDistanzen();
        Iterator<Double> i=distanzen.iterator();
        double zeit=0;
        zeiten.add(zeit);
        while(i.hasNext()) {
            zeit=i.next()/60; // Distanz geteilt durch 60km/h um auf die verstrichene Zeit
zu kommen. 60km/h == Durchschnittsgeschwindigkeit
            zeiten.add(zeit);
        }
    }

    ArrayList<Double> berechneDistanzen() {
        ArrayList<Double> distanzen=new ArrayList<Double>();
        Iterator<Tankstelle> it=route.iterator();
        double d=0.00;
        Tankstelle t1;
        Tankstelle tmp;
        if(it.hasNext()) {t1=it.next();}
        else {t1=null;}
        while(it.hasNext()) {
            tmp=t1;
            t1=it.next();
            d=berechneDistanz(tmp, t1);
           distanzen.add(d);
        }
        return distanzen;
    }

    double berechneDistanzGesamt() {

        Iterator<Tankstelle> it=route.iterator();
        double d=0.00;
        Tankstelle t1;
        Tankstelle tmp;
        if(it.hasNext()) {t1=it.next();}
        else {t1=null;}
        while(it.hasNext()) {
            tmp=t1;
```

```
                t1=it.next();
        d+=6378.388*Math.toRadians(Math.acos(Math.sin(tmp.getLat())*Math.sin(t1.getLat())
        +Math.cos(tmp.getLat())*Math.cos(t1.getLat())*Math.cos(t1.getLon()-tmp.getLon())));
        }
        return d;
    }
    double berechneVerbrauchGesamt() {
        double d=(berechneDistanzGesamt()/100)*5.6;
        return d;


    }
}
```

Tankstelle.java:

```
public class Tankstelle {
    int id;
    String name;
    String firma;
    String adresse;
    double lon;
    double lat;


    Tankstelle(int ID, String NAME, String FIRMA, String ADRESSE, double LON, double
LAT){
        this.id=ID;
        this.name=NAME;
        this.firma=FIRMA;
        this.adresse=ADRESSE;
        this.lon=LON;
        this.lat=LAT;
    }

    public String toString() {
        String s=null;
        s=Integer.toString(id)+": "+this.firma;
        return s;
    }

    public String information() {
        String s=null;
```

```
        s="ID: "+this.id+"\n"+"Name: "+this.name+"\n"+"Unternehmen:
"+this.firma+"\n"+"Adresse: "+this.adresse+"\n"+"Koordinaten: LAT: "+this.lat+" LON:
"+this.lon;
        return s;
    }

    double getLat() {
        return this.lat;
    }

    double getLon() {
        return this.lon;
    }

    int getId() {
        return this.id;
    }

    String getAdresse() {
        return this.adresse;
    }

    String getFirma() {
        return this.firma;
    }

    String getName() {
        return this.name;
    }
}
```