

ComPortTester

```
import java.awt.Button;
import java.awt.DisplayMode;
import java.awt.Frame;
import java.awt.GraphicsConfiguration;
import java.awt.GraphicsDevice;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Enumeration;
import javax.comm.*;
public class Main {
//Variablendeklaration
private static final long serialVersionUID = 1L;
private static Button b1;
private static List list;
static Enumeration<?> ports;
private static CommPortIdentifier pID;
private Frame f;
public Main (){
/*
    * Kopiervorgang */
File win32 = new File(System.getProperty("java.home")+"\\bin\\win32com.dll")
;
File prop = new File(System.getProperty("java.home")+"\\lib\\
javax.comm.properties");
if(!win32.isFile()){
System.out.println(copyFile("win32com.dll",System.getProperty("java.home")+"
\\bin\\win32com.dll"));
}
if(!prop.isFile()){
System.out.println(copyFile("javax.comm.properties",System.getProperty(
"java.home")+"\\lib\\javax.comm.properties"));
}
```

```

//fenster erzeugen
f = new Frame("verfügbare Ports");
//was machen wir wenn sich der fensterstatus verändert, also das fenster extern
f.addWindowListener(new ExitWindowListener());
// "Null-Layout" setzen
f.setLayout(null);
//neues listenobjekt
list = new List();
list.setBounds(0, 0, 299, 120); // x, y, breite, höhe
//itemlistener reagiert auf klicks in der liste
list.addItemListener(new ItemListener() {
    @Override public void itemStateChanged(ItemEvent arg0) {
//wenn klick auf einen item
System.out.println(list.getItem(Integer.valueOf(arg0.getItem().toString())));
;
        CommPortIdentiflier ci = null;
try {
//versuche den eintrag in der liste zu lesen
        ci = (CommPortIdentiflier.getPortIdentiflier(list.getItem(Integer.valueOf(
arg0.getItem().toString()))));
//fenster schließen
        ci.dispose();
//lesefenster erzeugen
new ReaderWindow(ci);
} catch (Exception e) {
    e.printStackTrace();
}
}});
/*
 * Fensterschnulli */
b1 = new Button();
b1.setBounds(110,125,80,20);
b1.setLabel("schliessen");
b1.setName("Fenster schliessen");
b1.addActionListener(buttonListener);
f.add(list);
f.add(b1);
DisplayMode dMode = null;
GraphicsConfiguration gc = f.getGraphicsConfiguration();
GraphicsDevice gd = gc.getDevice();
dMode = gd.getDisplayMode();
int width = dMode.getWidth()/2-150;
int height = dMode.getHeight()/2-77;
f.setLocation(width,height);
f.setSize(300,150);
f.setResizable(false);
f.setUndecorated(true);
f.setVisible(true);

```

```

/*
 * ENDE Fensterschnulli */
}
//wenn schließen gedrückt
ActionListener buttonListener = new ActionListener() {
public void actionPerformed(ActionEvent event) {
    dispose();
try{
    SimpleRegedtSerialPort().close();
}catch(Exception ex){}System.exit(0);
}
};
//holen der verfügbaren ports
public static void getPortList(){
    ports= CommPortIdentifier.getPortIdentifiers();
    while (ports.hasMoreElements()) {
        pID= (CommPortIdentifier) ports.nextElement();
        System.out.println("Port " + pID.getName());
        if (pID.getPortType() == CommPortIdentifier.PORT_SERIAL) {
            listadd(pID.getName());
        }
    }
}
//kopieren der dateien
public static boolean copyFile(String fileIn,String fileOut) {
try {
File f1 = new File(fileIn);
File f2 = new File(fileOut);
InputStream in = new FileInputStream(f1);
OutputStream out = new FileOutputStream(f2);
byte[] buf = new byte[1024];
int len;
while ((len = in.read(buf)) > 0) {
    write(buf, 0, len);
}
in.close();
out.close();
return true;
} catch (FileNotFoundException ex) {
return false;
} catch (IOException e) {
return false;
}
}
//die MAIN methode, hier geht alles los
public static void main (String args[]){
new Main ();
getPortLi$;

```

```

}
//wenn externes beenden
class ExitWindowListener extends WindowAdapter{
public void windowClosing(WindowEvent e){
    dfdispose();
try{
    SimpleReader.getSerialPort().close();
}catch(Exception ex){}System.exit(0);
}
}
}
}

```

Die Klasse [ReaderWindow](#) wird dann aufgerufen sobald ein Element in der Liste geklickt wurde. Dann wird [PortReader](#) implementiert.