PortReader

```
import java.io.IOException;
import java.io.InputStream;
import javax.comm.CommPortIdentifier;
import javax.comm.PortInUseException;
import javax.comm.SerialPort;
import javax.comm.SerialPortEvent;
import javax.comm.SerialPortEventListener;
public class PortReader implements SerialPortEventListener {
  //@SuppressWarnings("unchecked")
 // private static Enumeration ports;
  //static CommPortIdentifier pID;
  private InputStream inStream;
  private static SerialPort serPort=null;
  public PortReader(CommPortIdentifier pID) throws Exception{
    ReaderWindowgetTa().append("Lausche auf Port:"+pID.getName()+"\n");
  try{
     serPort= (SerialPort) pID.open("PortReader", 2000);
     inStream= serPort.getInputStream();
     serPortaddEventListener(this);
     serPortnotifyOnDataAvailable(true);
     serPortsetSerialPortParams(9600, SerialPort.DATABITS_8, SerialPort.
STOPBITS_1, SerialPort.PARITY_NONE);
  }catch(PortInUseException piu){
    ReaderWindowgetTa().append("Der Port ist in Benutzung\n");
  }
  }
  public void serialEvent(SerialPortEvent event) {
    switch (event.getEventType()) {
    case SerialPortEvent.BI:
      ReaderWindow.getTa().append("SerialPortEvent.BI occurred\n");
    case SerialPortEvent.OE:
     ReaderWindowgetTa().append("SerialPortEvent.OE occurred\n");
    case SerialPortEvent.FE:
     ReaderWindowgetTa().append("SerialPortEvent.FE occurred\n");
    case SerialPortEvent.PE:
     ReaderWindowgetTa().append("SerialPortEvent.PE occurred\n");
    case SerialPortEvent.CD:
     ReaderWindowgetTa().append("SerialPortEvent.CD occurred\n");
    case SerialPortEvent.CTS:
     ReaderWindowgetTa().append("SerialPortEvent.CTS occurred\n");
    case SerialPortEvent.DSR:
     ReaderWindowgetTa().append("SerialPortEvent.DSR occurred\n");
    case SerialPortEvent.RI:
     ReaderWindowgetTa().append("SerialPortEvent.RI occurred\n");
```

```
    case SerialPortEvent.OUTPUT_BUFFER_EMPTY:
     ReaderWindowgetTa().append(
"SerialPortEvent.OUTPUT_BUFFER_EMPTY occurred\n");
      break;
    case SerialPortEvent.DATA_AVAILABLE:
     ReaderWindowgetTa().append("SerialPortEvent.DATA_AVAILABLE occurred\n")
;
      byte[] readBuffer = new byte[20];
      try {
        while (inStream.available() > 0) {
          int numBytes = inStream.read(readBuffer);
          ReaderWindow.getTa().append("Bytes: "+numBytes+"\n");
        }
        ReaderWindow.getTa().append(new String(readBuffer));
      } catch (IOException ioe) {
       ReaderWindowgetTa().append("Exception " + ioe+"\n");
      }
      break;
    }
  }
public static SerialPort getSerPort() {
return serPort;
}
public void setSerPort(SerialPort serPort) {
 PortReaderserPort = serPort;
}
}
```

Wird von ComPortTester und ReaderWindow implementiert.