

Open Object Rexx (ooRexx)

Ein neues Kind auf der Straße von Linux ...

Prof. Dr. Rony G. Flatscher

Überblick

- Geschichte
- Rexx
 - "Alles ist eine Zeichenkette"
- Open Object Rexx (ooRexx)
 - "Alles ist ein Objekt"
- BSF4Rexx
 - "Java ist Object Rexx" 😊

Übersicht, 1

"**REXX** Eine der ersten ↑ Skriptsprachen, die von Michael Cowlshaw im Jahre 1979 bei ↑ IBM entwickelt wurde. REXX wurde ursprünglich für ↑ Großrechner konzipiert, auch ↑ Endbenutzer können bereits nach relativ kurzer Einarbeitungszeit produktiv damit arbeiten."

Aus: Hansen H.R., Neumann G.:

"Arbeitsbuch Wirtschaftsinformatik - IT-Lexikon, Aufgaben und Lösungen",
7. Auflage, UTB 1281, Lucius & Lucius 2007.

Geschichte Rexx, 1

- 1979 - IBM (Mike F. Cowlshaw, IBM-Fellow)
 - Nachfolger von einer kryptischen Skriptsprache ("EXEC") auf IBM Mainframes
 - Ziel: menschenfreundliche ("human centric") Sprache
 - Interaktiv (Interpreter)
 - REXX Akronym für "REstructured eXtended eXecutor"
 - Mike F. Cowlshaw's Blick auf 25 Jahre Rexx (2004, Rexx Symposium):
<http://www.rexxla.org/Symposium/2004/mikeec.pdf>
- Ab 1987 IBM's "SAA" für "Procedural Language"
 - Strategische Skriptsprache für sämtliche IBM-Plattformen
 - Entwicklung von kommerziellen und freien Versionen der Sprache, defacto für alle Betriebssysteme verfügbar

Geschichte Rexx, 2

- Zahlreiche Rexx-Implementierungen
 - Freie (z.B. Regina, BRexx)
 - Kommerzielle (Z.B. Workstation Group)
 - Betriebssystembestandteil
 - Amiga OS
 - IBM OS/2
- ANSI Rexx Standard in 1996
 - ANSI "Programming Language - REXX", X3.274-1996
 - INCITS (ANSI) J18 Standard 2006
 - <http://j18.incits.org>

REXX, Grundlegende Konzepte (Code 1)

```
#!/usr/bin/rexx

say "Hallo Chemnitz!" /* ergibt: Hallo Chemnitz! */
say 1/3               /* ergibt: 0.333333333 */

numeric digits 25    /* verwende 25 Ziffern bei Berechnungen */
say "1"/'3'         /* ergibt: 0.333333333333333333333333333333 */

"rm -rf *"          /* lösche rekursiv alle Dateien */
```

- "Alles ist eine Zeichenkette (String)"
- Alles wird in Großbuchstaben übersetzt
 - Ausnahme: Zeichenketten, die in " oder ' eingeschlossen sind
- Beliebig genaue dezimale Arithmetik
 - ANSI REXX Definitionen dienen als Grundlage für die Definition der dezimalen Arithmetik für andere Programmiersprachen ("BigDecimal") und werden für die Hardwareimplementierung
- Unbekannte Anweisungen bekommt der Aufrufer

REXX, Stammvariable ("Stem Variables", Code 2)

```
#!/usr/bin/rexx

file.1="max.txt"      /* Variable "FILE.1" erhält Wert zugewiesen */
file.2="pia.txt"     /* Variable "FILE.2" erhält Wert zugewiesen */
file.0=2             /* Variable "FILE.0" erhält Wert zugewiesen */

do i=1 to file.0     /* Schleife mit Laufvariable "I" */
  say file.i        /* "I" wird mit "1" und "2" ersetzt werden */
end
```

Ausgabe:
max.txt
pia.txt

- "Stem": alle Zeichen bis inklusive dem ersten Punkt
 - Ermöglicht assoziative Felder ("associative arrays")
- "Tail": alle Zeichen *nach* dem ersten Punkt
 - Punkte unterteilen Bezeichner
- Nichtinitialisierte Variable
 - Wie ein Literal (Wert: Bezeichner in Großbuchstaben)

REXX, Stammvariable ("Stem Variables", Code 3)

```
#!/usr/bin/rexx

Deutschland.Sachsen=4254000      /* Einwohner von Sachsen (2006) */
Deutschland.Sachsen.Chemnitz=245916 /* Einwohner von Chemnitz (2006) */

say Deutschland.Sachsen.Chemnitz /* ergibt: 245916 */

a="SACHSEN.CHEMNITZ"           /* definiere Wert des "Tails" */
say Deutschland.a              /* ergibt: 245916 */

a="SACHSEN"; b="CHEMNITZ";     /* definiere "Index"-Werte */
say deutschland.a.b           /* ergibt: 245916 */

say DeuTschLanD.SacHseN.ChemNitZ /* ergibt: 245916 */
```

```
Ausgabe:
245916
245916
245916
245916
```

- Nichtinitialisierte Variablen in Stammvariablen
 - Für die Bezeichner SACHSEN und CHEMNITZ werden die Bezeichner selbst als Zeichenkettenwerte eingesetzt
 - Derartige Werte sind immer in Großbuchstaben, da REXX vor dem Ausführen von Anweisungen alles, was außerhalb von Anführungszeichen steht in Großbuchstaben übersetzt

Open Object Rexx



Geschichte IBM Object REXX, 1

- Seit Anfang der 90er
 - auf Initiative der einflußreichen IBM-Benutzervereinigung "SHARE" Entwicklung einer objektorientierten Version von REXX
- "Object-based REXX" a.k.a. "Object REXX"
 - Vollständig kompatibel zum herkömmlichen (prozeduralen) REXX
 - Intern vollständig objektorientiert aufgebaut
 - *Prozedurale Anweisungen werden intern in objektorientierte umgeformt !*
 - Mächtiges Objektmodell (z.B. Metaklassen, Mehrfachvererbung)
 - Weiterhin einfache Syntax

Geschichte IBM Object Rexx, 2

- Nach 1997
 - Entwicklung von IBM Object REXX wird von den USA (Rick McGuire) nach Deutschland (Kurt Märker, verst.) verlegt
- Verfügbarkeit
 - 1997 für OS/2 Warp 4 (frei) und Warp 3 (frei) (mit [SOM](#))
 - 1998 für AIX (Version zum Evaluieren)
 - 1998 Linuxstudie (Kurt Märker, verst.)
 - 1998 für Windows 95 und Windows NT (mit [OLEAutomation/ActiveX](#))
- Kommerzielles Produkt mit Support
 - AIX (€), OS/2 (Betriebssystembestandteil), Windows (€)

Geschichte Open Object Rexx (ooRexx), 1

- 2004
 - Verhandlungen zwischen dem gemeinnützigen Verein "Rexx Language Association" (<http://www.RexxLA.org>) und IBM
 - RexxLA-Verhandlungsführer: Pam Taylor (USA, Managerin), Mark Hessling (Australien, Rexx-Interpreter-Entwickler), Rony G. Flatscher (Österreich, Academy)
 - Im Herbst Übergabe des IBM Object REXX Quellcodes und der Dokumentation an RexxLA !
- April 2005: "Object Rexx goes FOSS !"
 - Erste Veröffentlichung von "*Open Object Rexx (ooRexx) 3.0*" durch die RexxLA (<http://www.ooRexx.org>)

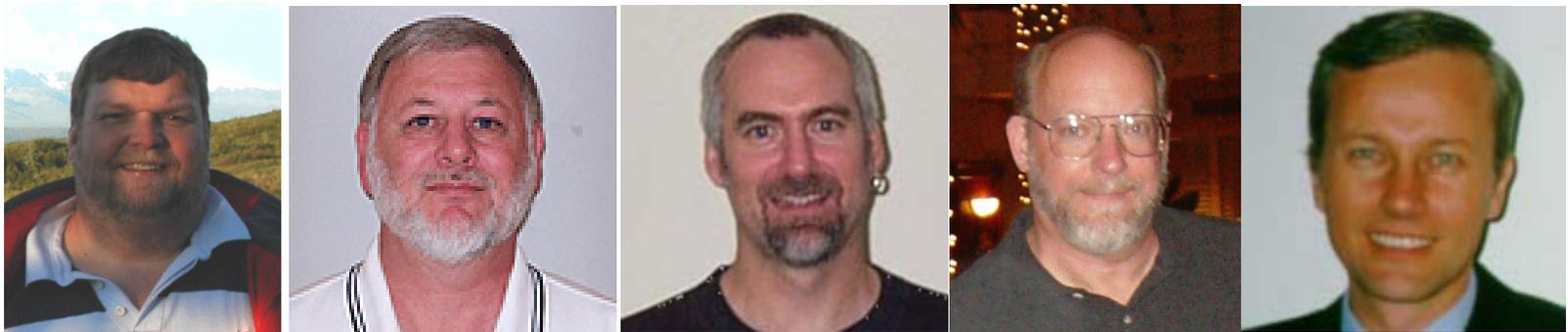


Geschichte Open Object Rexx (ooRexx), 2

- 2006
 - Version 3.1 im Sommer veröffentlicht
 - Dezember 2006 Version 3.1.1 veröffentlicht, Binaries für:
 - AIX
 - Linux
 - MacOSX (René Jansen, Niederlande; enthält BSF4Rexx)
 - Solaris
 - Windows
 - WWW
 - <http://www.ooRexx.org>
 - <http://sourceforge.net/projects/oorexx/>

Geschichte Open Object Rexx (ooRexx), 3

- Entwicklerteam (Committer)
 - Rick McGuire (Chefentwickler, "Vater" von Object Rexx, USA)
 - David Ashley (Dokumentation, USA)
 - Mark Hessling (Projektmanagement, Australien)
 - Mark Miesfeld (Entwickler, seit Anfang Oktober 2006, USA)
 - Rony G. Flatscher (Tests, Österreich)



Open Object Rexx, Grundlegende Konzepte, 1

- Interpreter, vollständig rückwärtskompatibel zu Rexx
- Von Smalltalk beeinflusst
- Neben anderen Dingen
 - Metaklassen
 - Reflection
 - One-off-Objekte
 - ...
- Expliziter Nachrichtenoperator (~, Tilde, "twiddle")
 - Kaskadierende Nachrichten möglich (~ ~)

Open Object Rexx, Grundlegende Konzepte, 2

- "Alles in Object Rexx ist ein Objekt"
 - Konzeptionell "lebt" ein Objekt
- Programmieren wird so verändert, dass einfach
 - Klassen und Methoden definiert werden können
 - Instanzen (= Objekte) von Klassen gebildet werden
 - Interaktion wie unter Menschen möglich, indem man Objekten (ausschließlich!) Nachrichten schickt
 - Objekte nehmen die Nachrichten entgegen und gehen auf die Suche nach einer Methode mit passenden Namen, aktivieren diese und geben ein allfälliges Resultat zurück

Open Object Rexx, Klassenhierarchie

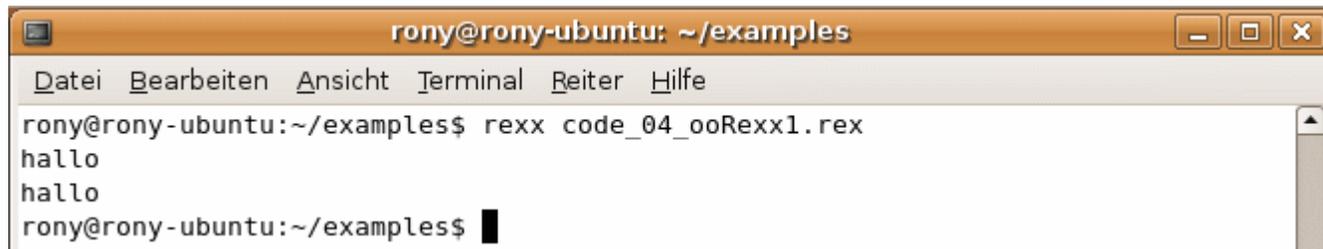
Erstaunlich wenige (= leicht zu lernende) Klassen!

Class Name	Comment
Object	Root class, fundamental
Alarm	Send message asynchronously
Array	Collection class, no predefinition of size or dimension necessary, ordered
Class	Metaclass, fundamental
Directory	Collection class, index is a string, one object per index, no order implied
List	Collection class, ordered
Message	Fundamental class
Method	Fundamental class
Monitor	Monitors messages sent to objects
MutableBuffer	Comparable to Java's StringBuffer
Queue	Collection class, ordered
CircularQueue	Collection class, ordered
RegularExpression	Regular expression class
Relation	Collection class, index is any object, multiple objects per index possible, no order implied
Bag	Index and associated object are the same object
Stem	Represents "classic Rexx" stems
Stream	Stream (e.g. file) input/output
String	Object's string values are not mutable
Supplier	Iterator for collection classes
Table	Collection class, index is any object, one object per index, no order implied
Set	Index and associated object are the same object

Klassisches und Object Rexx gemischt (Code 4)

```
#!/usr/bin/rexx

say reverse("ollah")      /* Eingebaute Funktion liefert: hallo */
say "ollah"~reverse       /* Stringmethode liefert:      hallo */
```



```
rony@rony-ubuntu: ~/examples
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
rony@rony-ubuntu:~/examples$ rexx code_04_ooRexx1.rex
hallo
hallo
rony@rony-ubuntu:~/examples$
```

- Klassisches Rexx
 - Syntax nach wie vor erlaubt
 - Anweisungen werden "hinter den Kulissen" in ihre objektorientierte Entsprechung umgeformt

Object Rexx

Klassen definieren und benutzen (Code 5)

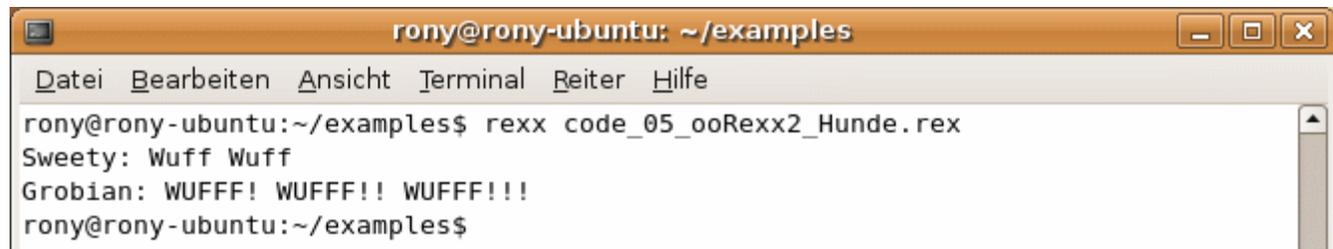
```
#!/usr/bin/rexx

.Hund      ~new("Sweety") ~belle  /* erzeuge Hund, lasse ihn bellen      */
.GrosserHund~new("Grobian")~belle /* erzeuge großen Hund, lasse ihn bellen */

::class Hund /* definiere eine Klasse, die "Hund" heißen soll      */
::method init /* Konstruktormethode (läuft für jedes neu erzeugte Objekt) */
  expose name /* definiere Objektattribute "name"                  */
  use arg name /* Argument wird im Objektattribut gespeichert      */

::method name attribute /* Attributmethode: Werte abfragen und setzen */
::method belle          /* Definiere eine Methode "belle"           */
  say self~name:" " "Wuff Wuff"

/* "GrosserHund" spezialisiert "Hund": alles was "Hund" kann wird verfügbar*/
::class GrosserHund subclass Hund
::method belle          /* Große Hunde bellen anders als normale Hunde      */
  say self~Name:" " "WUFFF! WUFFF!! WUFFF!!!"
```



```
rony@rony-ubuntu: ~/examples
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
rony@rony-ubuntu:~/examples$ rexx code_05_ooRexx2_Hunde.rex
Sweety: Wuff Wuff
Grobian: WUFFF! WUFFF!! WUFFF!!!
rony@rony-ubuntu:~/examples$
```

Object Rexx

Mehrfachvererbung (Code 6)

```
#!/usr/bin/rexx

/* Mehrfach-Vererbung */
.StrassenFahrzeug ~new("LKW")           ~fahre
.WasserFahrzeug   ~new("Boot")          ~schwimme
.AmphibienFahrzeug~new("SchwimmAuto")~zeigWasDuKannst

::CLASS Fahrzeug          /* Definiere eine Klasse, die "Fahrzeug" heißen soll */
::METHOD name ATTRIBUTE /* Definiere Attributmethode für "name" */
::METHOD init             /* Implementiere eine Konstruktormethode */
    self~name=ARG(1)      /* Argument wird dem Objektattribut "name" zugewiesen */

::CLASS StrassenFahrzeug  MIXINCLASS Fahrzeug /* Definiere Straßenfahrzeuge */
::METHOD fahre           /* define das Verhalten "fahre" */
    SAY self~name": 'Ich fahre jetzt...'" /* fragt Objektattribut "name" ab */

::CLASS WasserFahrzeug    MIXINCLASS Fahrzeug /* Definiere Wasserfahrzeuge */
::METHOD schwimme        /* define das Verhalten "schwimme" */
    SAY self~name": 'Ich schwimme jetzt...'"

::CLASS AmphibienFahrzeug SUBCLASS StrassenFahrzeug INHERIT WasserFahrzeug
::METHOD zeigWasDuKannst/* zeige Effekt der Mehrfachvererbung */
    self~fahre           /* Methode aus Klasse "StrassenFahrzeug" */
    self~schwimme        /* Methode aus Klasse "Wasserfahrzeug" */
```

```
rony@rony-ubuntu:~/examples$ rexx code_06_ooRexx2_Amphi.rex
LKW: 'Ich fahre jetzt...'
Boot: 'Ich schwimme jetzt...'
SchwimmAuto: 'Ich fahre jetzt...'
SchwimmAuto: 'Ich schwimme jetzt...'
```

BSF4Rexx – "Bean Scripting for Rexx"



BSF4Rexx, 1

- Apache's "Bean Scripting Framework"

<http://jakarta.apache.org/bsf/>

- Erlaubt es in sehr einfacher Form, von Java aus Skripte in fremden Sprachen ausführen zu lassen
- Ermöglicht es aufgerufenen Skripten, auf Java-Objekte zuzugreifen

- BSF4Rexx

<http://wi.wu-wien.ac.at/rgf/rexx/bsf4rexx/current/>

- BSF mit Rexx-Unterstützung
- Bestandteil der ooRexx-Distribution für MacOSX !

BSF4Rexx, 2

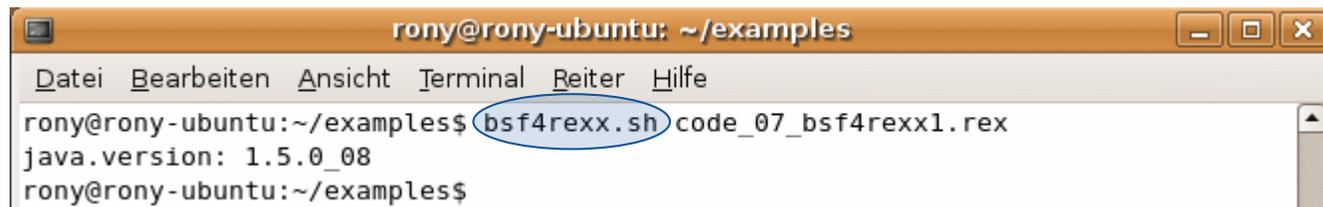
- "BSF.CLS"
 - ooRexx-Programm, in der BSF4Rexx-Distribution enthalten
 - Stellt spezielle Object Rexx Unterstützung bereit
 - Lässt (streng typisiertes) Java als Object Rexx erscheinen
 - Vereinfacht das Programmieren mit Java drastisch
 - Man muss nur die Javadocs (HTML) lesen können!
 - BSF4Rexx-Programme sind *plattformunabhängig* !
 - Java ist praktisch auf jedem Rechner installiert
 - *Die "Java Runtime Environment (JRE)" kann als riesige, externe und bereits portierte ooRexx-Klassenbibliothek angesehen werden!*

Object Rexx

Java-Klasse "java.lang.System" benutzen (Code 7)

```
#!/usr/bin/rexx

call bsf.cls      /* rufe das Programm "BSF.CLS" auf (Java-Unterstützung) */
s=.bsf~bsf.loadClass("java.lang.System") /* lade die Java-System-Klasse */
say "java.version:" s~getProperty('java.version') /* liefert Java-Version */
```



```
rony@rony-ubuntu: ~/examples
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
rony@rony-ubuntu:~/examples$ bsf4rex.sh code_07_bsf4rex1.rex
java.version: 1.5.0_08
rony@rony-ubuntu:~/examples$
```

- Steuert Java, Java-Anwendungen und Programme, die über Java-Schnittstellen verfügen
- Java Objekte sehen aus wie Object Rexx Objekte, an die man einfach Object Rexx Nachrichten schickt
- Man kann daher sehr einfach ooRexx als Skriptsprache auch in Java-Anwendungen einbauen!

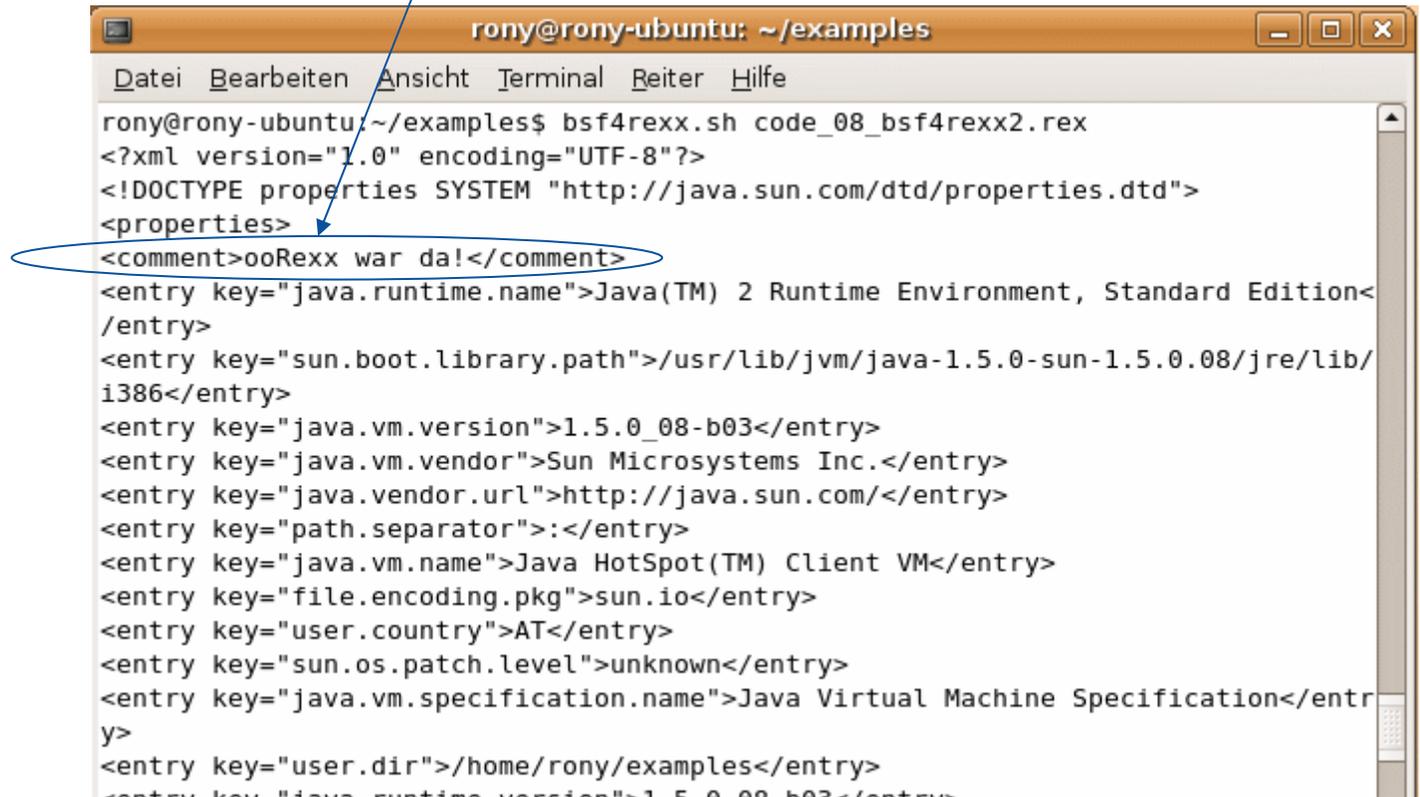
Object Rexx

Java-Klasse "java.lang.System" benutzen (Code 8)

```
#!/usr/bin/rexx
```

```
s=.bsf~bsf.loadClass("java.lang.System") /* lade die Java-System-Klasse */  
s~getProperties~storeToXML(s~out, "ooRexx war da!")
```

```
::requires bsf.cls /* ruft als Allererstes das Programm "BSF.CLS" auf */
```



```
rony@rony-ubuntu: ~/examples  
Datei Bearbeiten Ansicht Terminal Reiter Hilfe  
rony@rony-ubuntu:~/examples$ bsf4rexx.sh code_08_bsf4rexx2.rex  
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">  
<properties>  
<comment>ooRexx war da!</comment>  
<entry key="java.runtime.name">Java(TM) 2 Runtime Environment, Standard Edition</entry>  
<entry key="sun.boot.library.path">/usr/lib/jvm/java-1.5.0-sun-1.5.0.08/jre/lib/i386</entry>  
<entry key="java.vm.version">1.5.0_08-b03</entry>  
<entry key="java.vm.vendor">Sun Microsystems Inc.</entry>  
<entry key="java.vendor.url">http://java.sun.com/</entry>  
<entry key="path.separator">:</entry>  
<entry key="java.vm.name">Java HotSpot(TM) Client VM</entry>  
<entry key="file.encoding.pkg">sun.io</entry>  
<entry key="user.country">AT</entry>  
<entry key="sun.os.patch.level">unknown</entry>  
<entry key="java.vm.specification.name">Java Virtual Machine Specification</entry>  
<entry key="user.dir">/home/rony/examples</entry>  
<entry key="java.runtime.version">1.5.0_08-b03</entry>
```

Object Rexx & "OpenOffice.org (OOo)"



"OpenOffice.org (OOo)", 1

- OpenOffice.org (OOo)
 - <http://www.OpenOffice.org>
- "Universal Network Objects (UNO)"
 - Komponentenarchitektur
 - Ursprünglich in C++ geschrieben
 - Mittlerweile sind alle Schnittstellen in Java verfügbar
 - OOo kann von Java aus so angesteuert werden wie in C++ !
 - Seit OOo Version 2.0 eine offene Skriptingschnittstelle, die in Java geschrieben ist
 - Bis dahin nur OOo Basic und Python als Skriptingsprachen verfügbar
 - Nun auch JavaScript (Rhino), BeanShell

"OpenOffice.org (OOo)", 2

- Java ?
- BSF4Rexx !
- "UNO.CLS"
 - ooRexx-Programm, in der BSF4Rexx-Distribution enthalten!
 - Spezielle Unterstützung für die Ansteuerung der UNO-Komponenten
 - Beinhaltet zahlreiche hilfreiche Klassen und Routinen, die das Leben von Programmierern sehr erleichtern
 - Beinhaltet auch die Möglichkeit, zur Laufzeit die aktuellen Definitionen von UNO Komponenten herauszufinden, was manchmal eine *ungeheuerliche Vereinfachung* beim Recherchieren bedeuten kann!

"OpenOffice.org"

Ein Textverarbeitungsdokument mit Java (1/2)

```
import com.sun.star.beans.PropertyValue;
import com.sun.star.comp.helper.Bootstrap;
import com.sun.star.frame.XComponentLoader;
import com.sun.star.frame.XDesktop;
import com.sun.star.lang.XComponent;
import com.sun.star.lang.XMultiComponentFactory;
import com.sun.star.text.XText;
import com.sun.star.text.XTextDocument;
import com.sun.star.uno.UnoRuntime;
import com.sun.star.uno.XComponentContext;

class CreateTextDocument {
    public static void main (String args[]) {
        try {
            XComponentContext xContext=Bootstrap.bootstrap(); // bootstrap UNO
            XMultiComponentFactory xMCF=xContext.getServiceManager();
            if (xMCF != null) {
                Object oDesktop=xMCF.createInstanceWithContext("com.sun.star.frame.Desktop", xContext);
                XDesktop xDesktop=(XDesktop) UnoRuntime.queryInterface(XDesktop.class, oDesktop);

                XComponentLoader xComponentLoader=(XComponentLoader)
                    UnoRuntime.queryInterface(XComponentLoader.class, xDesktop);

                String url="private:factory/swriter"; // define a text document
                PropertyValue noProps[]=new PropertyValue[0]; // no properties
                XComponent xWriterComponent=xComponentLoader.loadComponentFromURL(
                    url, "_blank", 0, noProps);
            }
        }
    }
}

// Fortsetzung nächste Seite ...
```

"OpenOffice.org"

Ein Textverarbeitungsdocument mit Java (2/2)

```
// ... Fortsetzung von vorheriger Seite
```

```
    XTextDocument xTextDocument=(XTextDocument)  
        UnoRuntime.queryInterface(XTextDocument.class, xWriterComponent);
```

```
    XText xText=xTextDocument.getText();  
    xText.setString("Hallo Chemnitz, hier spricht Java!");
```

```
    }
```

```
  }
```

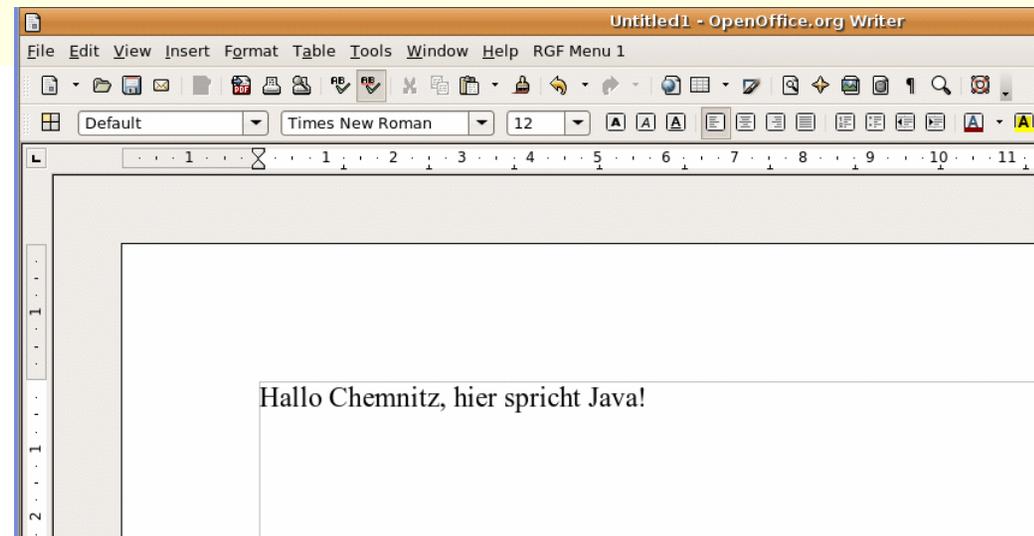
```
  catch( Exception e) {  
    e.printStackTrace(System.err);  
    System.exit(1);
```

```
  }
```

```
  System.exit(0);
```

```
}
```

```
}
```



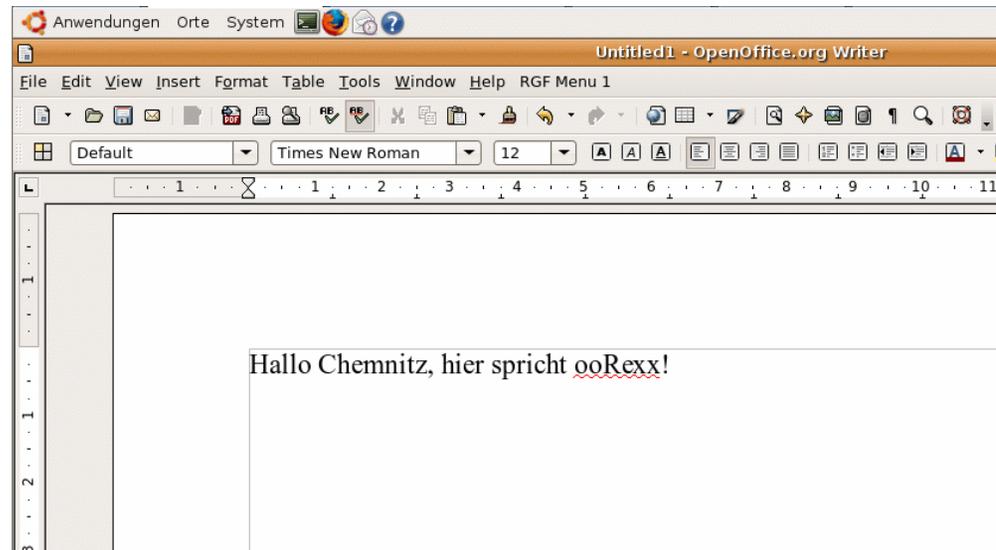
"OpenOffice.org"

Ein Textverarbeitungsdocument mit Object Rexx

```
#!/usr/bin/rexx

call uno.cls /* rufe das Programm "UNO.CLS" auf (OOo-Unterstützung) */
oDesktop=UNO.createDesktop() /* erstelle das OOo-Desktop-Objekt */
xComponentLoader=oDesktop~XDesktop~XComponentLoader

/* URL-Definition bewirkt das Erzeugen eines leeren Textdokuments */
url = "private:factory/swriter" /* URL könnte auch auf Datei verweisen */
xWriterComponent=xComponentLoader~loadComponentFromURL(url,"_blank", 0,.UNO~noProps)
xText=xWriterComponent~XTextDocument~getText /* fordert TextObjekt an */
xText~setString("Hallo Chemnitz, hier spricht ooRexx!") /* setzt Text */
```



"OpenOffice.org"

Informationen über Textdokumente

```
#!/usr/bin/rexx

/* erstelle das OOo-Desktop-Objekt, erfrage die Service-Factory */
ocl=UNO.createDesktop()~XDesktop~XComponentLoader

/* definiert Dokumenten-URL, file-, ftp- oder http-URL auch möglich */
url = "private:factory/swriter" /* "swriter": text component */
otc=ocl~loadComponentFromURL(url, "_blank", 0, .UNO~noProps)

/* Informations-Dialog zeigen */
.bsf.dialog~messageBox("Hallo Chemnitz, hier spricht ooRexx!")

LF="0a"x /* "LINEFEED" (Zeilenende) Zeichen */
str="ooRexx, am" date("S",,, "-") time('L')
str=str "(Aktuelle UNO-Definitionen erfragen.)" LF LF
/* zeige Services und Interfaces eines Textdokuments */
str=str || ppd("Services: " otc~uno.getServiceNames, " ", LF) LF
str=str || ppd(" Interfaces: " otc~uno.getInterfaceNames, " ", LF)
/* fordert TextObbjekt an */
oText=otc~XTextDocument~getText~setString(str) /* setzt Text */

::requires UNO.CLS /* ruft als Allererstes das Programm "UNO.CLS" auf */
```



ooRexx, am 2007-03-04 20:33:33.065910 (Aktuelle UNO-Definitionen erfragen.)

Services:

com.sun.star.document.OfficeDocument
com.sun.star.text.GenericTextDocument
com.sun.star.text.TextDocument

Interfaces:

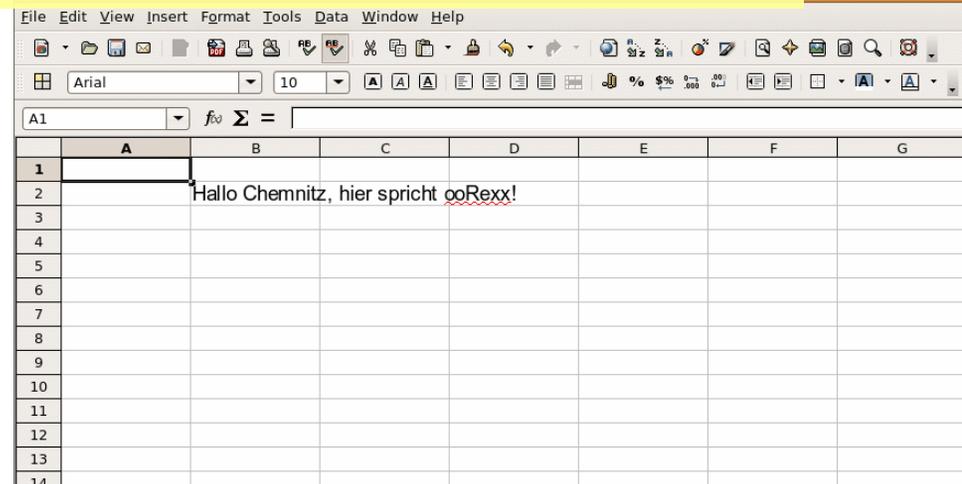
com.sun.star.frame.XModel
com.sun.star.util.XModifiable
com.sun.star.frame.XStorable
com.sun.star.view.XPrintable
com.sun.star.document.XEventBroadcaster
com.sun.star.document.XEventsSupplier
com.sun.star.document.XDocumentInfoSupplier
com.sun.star.document.XViewDataSupplier
com.sun.star.view.XPrintJobBroadcaster
com.sun.star.lang.XMultiServiceFactory
com.sun.star.text.XTextDocument
com.sun.star.util.XSearchable
com.sun.star.util.XRefreshable
com.sun.star.text.XFootnotesSupplier
com.sun.star.text.XEndnotesSupplier
com.sun.star.util.XReplaceable
com.sun.star.text.XPagePrintable
com.sun.star.text.XReferenceMarksSupplier
com.sun.star.text.XLineNumberingSupplier
com.sun.star.text.XChapterNumberingSupplier
com.sun.star.beans.XPropertySet
com.sun.star.text.XTextGraphicObjectsSupplier
com.sun.star.text.XTextEmbeddedObjectsSupplier
com.sun.star.text.XTextTablesSupplier
com.sun.star.style.XStyleFamiliesSupplier
com.sun.star.text.XBookmarksSupplier
com.sun.star.text.XDocumentIndexesSupplier
com.sun.star.text.XTextFieldsSupplier
com.sun.star.text.XTextFramesSupplier
com.sun.star.text.XTextSectionsSupplier
com.sun.star.util.XNumberFormatsSupplier

"OpenOffice.org"

Ein Tabellenkalkulationsdokument mit Object Rexx

```
#!/usr/bin/rexx
call uno.cls /* rufe das Programm "UNO.CLS" auf (OOo-Unterstützung) */
oDesktop=UNO.createDesktop() /* erstelle das OOo-Desktop-Objekt */
xComponentLoader=oDesktop~XDesktop~XComponentLoader

/* URL-Definition bewirkt das Erzeugen eines leeren Calc-Dokuments */
url = "private:factory/scalc" /* URL könnte auch auf Datei verweisen */
calc=xComponentLoader~loadComponentFromURL(url,"_blank", 0, .UNO~noProps)
/* lade die Sammlung der Calc-Blätter */
blaetter =calc~XSpreadSheetDocument~getSheets
/* fordere das "XIndexAccess"-Interface an, lade das 1. Blatt (Index=0) */
tabellenKalkulationsBlatt1=blaetter~XIndexAccess~getByIndex(0) ~XSpreadSheet
/* fordere das "XSpreadSheet"-Interface an */
xSheet=tabellenKalkulationsBlatt1~XSpreadSheet
/* füge einen Eintrag ein */
call UNO.setCell xSheet, "B2", "Hallo Chemnitz, hier spricht ooRexx!"
```



"OpenOffice.org"

Informationen über Tabellenkalkulationsdokumente

```
#!/usr/bin/rexx
call uno.cls /* rufe das Programm "UNO.CLS" auf (OOo-Unterstützung) */
oDesktop=UNO.createDesktop() /* erstelle das OOo-Desktop-Objekt */
xComponentLoader=oDesktop~XDesktop~XComponentLoader
/* URL-Definition bewirkt das Erzeugen eines leeren Calc-Dokuments */
url = "private:factory/scalc" /* URL könnte auch auf Datei verweisen */
calc=xComponentLoader~loadComponentFromURL(url,"_blank", 0, .UNO~noProps)
/* lade das 1. Blatt (0-basiert) aus der Sammlung der Calc-Blätter */
xs=calc~XSpreadSheetDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet
/* Informations-Dialog zeigen */
.bsf.dialog~messageBox("Hallo Chemnitz, hier spricht ooRexx!")
str="ooRexx, am" date("S",,, "-") time('L')
call uno.setCell xs, "A1", str "(aktuelle UNO-Definitionen abfragen.)"
/* zeige Services und Interfaces eines Textdokuments */
i=trageInfosEin(xs, "Services", calc~uno.getServiceNames, 2)
i=trageInfosEin(xs, "Interfaces", calc~uno.getInterfaceNames, i)

::requires UNO.CLS /* ruft als Allererstes das Programm "UNO.CLS" auf */

::routine trageInfosEin /* trägt Infos in Tabellenkalkulationsblatt ein*/
use arg xs, titel, verkodierteInfos, startZeile
/* Koordinaten können auch als (0-basierte) Zahlen angegeben werden: */
call uno.setCell xs, 0, startZeile, titel /* Titel eintragen */
do i=startZeile+1 while verkodierteInfos<>" "
    parse var verkodierteInfos info verkodierteInfos /* zerlege String */
    call uno.setCell xs, 1, i, info /* Zeige extrahierte Info */
end
return i /* liefere nächste Zeilenposition zurück */
```



A1 f_x Σ = ooRexx, am 2007-03-04 20:40:44.203882 (aktuelle UNO-Definitionen abfragen.)

	A	B	C	D	E	F	G	H	I
1	ooRexx, am 2007-03-04 20:40:44.203882 (aktuelle UNO-Definitionen abfragen.)								
2									
3	Services								
4		com.sun.star.sheet.SpreadsheetDocument							
5		com.sun.star.sheet.SpreadsheetDocumentSettings							
6		com.sun.star.document.OfficeDocument							
7	Interfaces								
8		com.sun.star.lang.XMultiServiceFactory							
9		com.sun.star.frame.XModel							
10		com.sun.star.document.XActionLockable							
11		com.sun.star.document.XLinkTargetSupplier							
12		com.sun.star.util.XProtectable							
13		com.sun.star.sheet.XSpreadsheetDocument							
14		com.sun.star.sheet.XCalculatable							
15		com.sun.star.sheet.XDocumentAuditing							
16		com.sun.star.sheet.XConsolidatable							
17		com.sun.star.sheet.XGoalSeek							
18		com.sun.star.drawing.XDrawPagesSupplier							
19		com.sun.star.style.XStyleFamiliesSupplier							
20		com.sun.star.util.XNumberFormatsSupplier							
21		com.sun.star.beans.XPropertySet							
22		com.sun.star.frame.XModel							
23		com.sun.star.util.XModifiable							
24		com.sun.star.frame.XStorable							
25		com.sun.star.view.XPrintable							
26		com.sun.star.document.XEventBroadcaster							
27		com.sun.star.document.XEventsSupplier							
28		com.sun.star.document.XDocumentInfoSupplier							
29		com.sun.star.document.XViewDataSupplier							
30		com.sun.star.view.XPrintJobBroadcaster							
31									
32									

Zusammenfassung und Ausblick, 1

- ooRexx
 - Einfache Syntax
 - Einfach zu erlernen, daher sehr produktive Skriptsprache
 - Vollständig rückwärtskompatibel zu "klassischem" Rexx
 - ooRexx als Ersatz für "klassische" Rexx-Interpreter
 - Aktuelle Version 3.1.1
 - Beinhaltet Bugfixes im Vergleich zu IBM Object REXX 2.1
 - Beinhaltet bereits Erweiterungen und Weiterentwicklungen
 - Weitere Version 3.x geplant
 - Erweiterungen und Weiterentwicklungen (z.B. neue Methoden)

Zusammenfassung und Ausblick, 2

- Zukunft
 - ooRexx 4.x
 - Seit 2005 in Arbeit
 - 64-Bit
 - Neues, einfach zu handhabendes API für C++
 - ooRexx Objekte als Proxies für C++ Objekte
 - C++ Objekte als Proxies für ooRexx Objekte
 - Eine Reihe interessanter, neuer Konzepte angedacht, z.B.
 - Snobol-ähnliches Pattern-Matching
 - Einführung einer optionalen Typisierung von Argumenten
 - ...
 - Kommt, "wenn es fertig ist"

Zusammenfassung und Ausblick, 3

- Was können **Sie** tun?
 - Ausprobieren und Spaß haben! ☺
 - Freie Opensource Alternativen aktiv einsetzen
 - ooRexx auf weitere Plattformen portieren
 - ooRexx für neue Distributionen verfügbar machen
 - Bei der Weiterentwicklung aktiv mithelfen, z.B.
 - Nützliche ooRexx Klassenbibliotheken aufbauen/testen helfen
 - Nützliche ooRexx Frameworks erstellen/testen helfen
 - Weitere TestUnits für das Testen aller Aspekte von ooRexx miterstellen helfen
 - Arbeiten am Kernel, wenn Sie Spaß an C++/Java/Mono und Skriptsprachen haben
 - ...

URLs, 1

- URLs zu Rexx, Open Object Rexx (ooRexx)

<http://www.ooRexx.org/download.html>

- Aktuelle Version von ooRexx, gelistet nach Betriebssystemen

<http://wi.wu-wien.ac.at/rgf/rexx/bsf4rexx/current/>

- Aktuelle Version von BSF4Rexx mit zahlreichen Beispielen

<http://sourceforge.net/projects/oorexx/>

<http://www.ooRexx.org/docs.html>

<http://www.RexxLA.org/>

<news:comp.lang.rexx>

<http://www.rexx.org>

<http://www.rexxinfo.org/>

URLs, 2

- Deutsche Folien zu ooRexx-Einführung an der WU Wien
 - Foliensätze 1-6 sind identisch (Einführung in ooRexx)
<http://wi.wu-wien.ac.at/rgf/wu/lehre/autojava/material/folien/>
 - Einführung in ooRexx **und** Automatisierung von Windows OLE/ActiveX
<http://wi.wu-wien.ac.at/rgf/wu/lehre/autowin/material/folien/>
 - Einführung in ooRexx **und** BSF4Rexx (Java-Anbindung): Linux, MacOSX, Windows
http://wi.wu-wien.ac.at/rgf/wu/lehre/poolv/material/English/foils/excercises/poolv_aufgaben.htm
 - Folienabgestimmte Musteraufgaben und Musterlösungen
- Englische Folien zur ooRexx-Einführung an der WU Wien
<http://wi.wu-wien.ac.at/rgf/wu/lehre/poolv/material/English/foils/>
 - Einführung in ooRexx; mit Metaklassen, Nebenläufigkeiten, Security Manager !
http://wi.wu-wien.ac.at/rgf/wu/lehre/poolv/material/English/foils/excercises/poo_exercises.htm
 - Folienabgestimmte Musteraufgaben und Musterlösungen

URLs, 3

- Studentenarbeiten, die Nutzen stiften können

<http://wi.wu-wien.ac.at/rgf/diplomarbeiten>

- Seminararbeiten, Bakkarbeiten, Diplomarbeiten
- Unter anderem über BSF4Rexx

http://wi.wu-wien.ac.at/rgf/diplomarbeiten/BakkStuff/2005/200507_Subversion_Hoisl/200507_AutomatingSubversion.pdf

- Unter anderem Automatisierung von OpenOffice mit ooRexx

http://wi.wu-wien.ac.at/rgf/diplomarbeiten/BakkStuff/2005/200511_OOo-Ahammer/200511_OOoAutomation.pdf

http://wi.wu-wien.ac.at/rgf/diplomarbeiten/BakkStuff/2006/200605_Burger/Bakk_Arbeit_Burger20060519.pdf

http://wi.wu-wien.ac.at/rgf/diplomarbeiten/BakkStuff/2006/200607_Hinz/20060712_OOo_calc_automation.pdf

http://wi.wu-wien.ac.at/rgf/diplomarbeiten/BakkStuff/2006/200607_Prem/20060724_ooRexxSnippetsOOoWriter_2.1_odt.pdf

http://wi.wu-wien.ac.at/rgf/diplomarbeiten/BakkStuff/2007/200702_Schmid/OODatabase_v4_1_final.pdf

http://wi.wu-wien.ac.at/rgf/diplomarbeiten/Seminararbeiten/2006s_wu/20060628_BSF4RexxSnippets_version_4.pdf

- Offizielle OpenOffice Snippets mit ooRexx-Beispielen !

<http://codesnippets.services.openoffice.org/>