

2010-Edition von BSF4ooRexx für Windows und Linux

<http://wi.wu.ac.at/rgf/rexx/bsf4oorexx/current/>

GSE WG Rexx plus ISPF



Prof. Dr. Rony G. Flatscher

Agenda

- Ausgangslage
- Das größte externe Rexx-Funktionspaket der Welt
- Demonstration unter Windows und Linux
- Zusammenfassung und Ausblick

Ausgangslage, 1

- Rexx
 - eine einfach zu erlernende, mächtige, typenlose Sprache
 - auf vielen Plattformen frei verfügbar
 - fehlende Funktionalität kann Rexx beigesteuert werden
 - "Rexx-Funktions-Pakete", z.B. Mark Hesslings "Rexx/SQL"
 - zumeist in C/C++ geschrieben
 - muss zwingend für jede Plattform und für jedes Betriebssystem extra kompiliert werden
 - wer macht diese Portierung?
 - wie lange ist diese Portierung möglich?
 - gibt es genug C/C++ Programmierer, die extra für Rexx neue benötigte Funktionspakete schreiben (können)?

Ausgangslage, 2

- ooRexx
 - ist Rexx !
 - erlaubt es zusätzlich
 - Strukturen mit Funktionen ("abstrakte Datentypen", abgekürzt: ADT) zu implementieren, indem man "Klassen" dafür schreibt
 - "Attribute" erlauben Daten vorzuhalten
 - "Methoden" erlauben Funktionen/Prozeduren zu implementieren, die direkt die verfügbaren Attribute benutzen können
 - von jeder Klasse ("Struktur", "Datentyp") kann man beliebig viele Ausprägungen ("Instanzen", "Objekte") erzeugen
 - die Funktionalität einer jeden Instanz ist in der Klasse als Methode festgelegt
 - durch Schicken von Methodennamen an Instanzen, können diese ausgeführt werden

Ausgangslage, 3

Beispiel

- ooRexx-Programm

```
p1=.Person~new          /* erzeuge eine Ausprägung der Klasse      */
p1~Vorname="Dietmar"    /* weise dem Attribut einen Wert zu        */
p1~Nachname="Stroh"     /* weise dem Attribute einen Wert zu      */
say "p1:" p1~string     /* zeige die Zeichenkette an              */

/* Implementierung der Struktur/des ADT mit Namen "Person" */
::class Person         /* Name der Struktur/des Datentyps        */

::attribute Vorname    /* Attribut zum Speichern des Vornamens    */
::attribute Nachname   /* Attribut zum Speichern des Nachnamens   */

::method string        /* erzeuge und retourniere eine Zeichenkette */
  expose Vorname Nachname /* direkter Zugriff auf die Attribute */
  return Nachname "," Vorname /* erzeuge und liefere Rückgabewert */
```

- Ausgabe

p1: Stroh, Dietmar

Ausgangslage, 4

- ab ooRexx 4.0 (Herbst 2009 veröffentlicht)
 - verfügbar für zahlreiche Plattformen, z.B. AIX, Linux (auch Linux/s390x!), MacOSX, MS Windows
 - erstmals 32-bit und 64-bit-Versionen von ooRexx verfügbar!
 - 32-bit-Version von ooRexx kann herkömmliche externe Rexx-Funktionspakete benutzen
 - 64-bit-Version von ooRexx benötigt eigens kompilierte, 64-bit externe Rexx-Funktionspakete!
 - der Quellcode von 64-bit externen Funktionspaketen muss auf die neuen APIs von ooRexx 4.x umgeschrieben werden!
 - für jede Zielplattform und jedes Zielbetriebssystem muss extra ein Kompilat erzeugt werden

- Rexx
 - "fehlende Funktionalität kann Rexx beigesteuert werden"
 - wie wäre es, wenn man diese fehlende Funktionalität anstatt mit C/C++ Programmen mit Java zur Verfügung stellt?
 - Ad Java
 - mittlerweile schnell genug
 - es gibt praktisch für jede nur vorstellbare Funktionalität entweder Java-Bibliotheken oder Java-Schnittstellen!
 - weit verbreitet: praktisch auf jedem Computer bereits installiert!
 - unabhängig von Hardware !
 - unabhängig von Betriebssystemen !

Das größte externe Rexx-Funktionspaket der Welt

- BSF4ooRexx !
 - schlägt Brücke von ooRexx nach Java !
 - Java-Klassen können einfach wie ooRexx-Klassen benutzt werden!
 - Java-Objekten kann man die Namen von Java-Methoden von ooRexx aus schicken!
 - Datenkonvertierungen erfolgen für uns Rexx-Programmierer vollautomatisch in beide Richtungen!
 - **Alle** Java-Bibliotheken können unverändert auf allen Hardware- und Betriebssystemplattformen benutzt werden!
 - Keine Notwendigkeit, Java-Klassen neu zu kompilieren!
 - Rexx-Programme sind damit (endlich!) plattformunabhängig!

Beispiel: Benutzung der Java-Klasse "java.awt.Frame" von ooRexx aus

- ooRexx-Programm

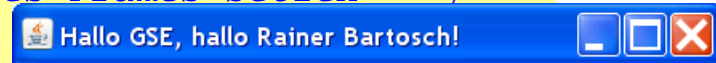
```
/* Erzeuge eine Ausprägung der Java-Klasse "java.awt.Frame",
   ein Fenster mit Rahmen, der einen Titel erhalten soll */
f=.bsf~new("java.awt.Frame", "Hallo GSE, hallo Rainer Bartosch!")
f~setVisible(.true) /* das Frame anzeigen */

zeit=5 /* schlafe fünf Sekunden */
say "Jetzt wird" zeit "Sekunden geschlafen..."
call sysSleep zeit

hoehe=f~getHeight /* beziehe die aktuelle Höhe in Pixel */
say "Das Frame ist" f~getWidth "Pixel breit und" hoehe "Pixel hoch."
breite=f~getWidth*3 /* aktuelle Breite wird verdreifacht */
say "Jetzt wird das Frame" breite "Pixel breit gemacht werden..."
f~setSize(breite, hoehe)/* Breite und Höhe des Frames setzen */

say "Jetzt wird" zeit "Sekunden geschlafen..."
call sysSleep zeit
say "Auf Wiedersehen!"

::requires BSF.CLS /* lade die Java-Unterstützung vorab */
```



```
Jetzt wird 5 Sekunden geschlafen...
Das Frame ist 148 Pixel breit und 40 Pixel hoch.
Jetzt wird das Frame 444 Pixel breit gemacht werden...
Jetzt wird 5 Sekunden geschlafen...
Auf Wiedersehen!
```

- Was muss man von Java wissen?
 - Die vollqualifizierten Namen der Java-Klassen (Datentypen)
 - Die APIs (Methoden) und Felder ("Attribute") der Java-Klassen
 - Java-Dokumentationen sind lediglich verlinkte HTML-Texte
 - über das Internet verfügbar, z.B.
 - <http://download.oracle.com/javase/1.4.2/docs/api/> ("Java 4")
 - <http://download.oracle.com/javase/6/docs/api/> ("Java 6")
 - Java-Dokumentationen können als zip-Archive sogar auf den eigenen Rechner geladen und ausgepackt werden
 - Tipp: lesen Sie immer zuerst die 1.4er-Dokumentation (wesentlich leichter lesbar), erst dann die 1.6er-Dokumentation

Java und Rexx: BSF4ooRexx-Brücke

- Java-Sprachphilosophie
 - Strikt typisiert, statisch
 - Zur Kompilierzeit möglichst vollständig überprüfbar
- Rexx-Sprachphilosophie
 - praktisch nicht typisiert ("alles ist Zeichenkette"), dynamisch
 - wird erst zur Laufzeit überprüft
- BSF4ooRexx überbrückt beide Sprachphilosophien
 - Java kann von Rexx praktisch ohne Typisierung benutzt werden!
 - Benutzung von Java wird dadurch **wesentlich vereinfacht!**

BSF4ooRexx, einige weitere Informationen, 1

- Mit ooRexx 4.0 eine neue, passende externe Funktionsbibliothek
 - in 32-bit **und** 64-bit verfügbar
- Spezielle Unterstützung für ooRexx
 - REXX-Paket "**BSF.CLS**"
 - verkleidet Java-Klassen und Java-Objekte als ooRexx-Klassen und ooRexx-Objekte
 - erlaubt das Senden von ooRexx-Nachrichten an Java-Objekte
 - REXX-Paket "**UNO.CLS**"
 - erlaubt vereinfachtes Programmieren von OpenOffice.org (OOo) !

BSF4ooRexx, einige weitere Informationen, 2

- Erlaubt sogar die Implementation von abstrakten Java-Methoden in Rexx (!!)
 - Methoden von Java-Interface-Klassen
 - per Definition sind Methoden in Java-Interfaces abstrakt
 - Java-Interfaces werden häufig für das "Listener"-Pattern benutzt
 - z.B. für die graphische Programmierung (awt-Listener-Objekte), für die Verarbeitung von XML-Daten (Listener-Objekte), etc.
 - Abstrakte Methoden von abstrakten Java-Klassen
 - z.B. zum Filtern der Dateien, die in Datei-Dialogen für die Auswahl durch die Benutzer dargestellt werden sollen

2010-Edition von BSF4ooRexx, 1

- Sämtliche Beispiele im Ordner "**samples**" überarbeitet
- Neues Installationsprogramm
 - für Linux und Windows
 - erstellt Menü "**BSF4ooRexx**" zum einfachen
 - Auffinden der Ordner ("**Information**", "**Samples**" und "**Utilities**")
 - Reinstallieren und Deinstallieren von BSF4ooRexx
 - wenn OpenOffice.org (OOo) am Zielrechner installiert ist
 - Installation der direkten OOo-Unterstützung ("**UNO.CLS**")
 - wenn OOo später installiert wird, dann im Menü einfach
 - "*BSF4ooRexx* → *Installation* → *Reinstall*" auswählen



2010-Edition von BSF4ooRexx, 2

- Dateiassoziationen
 - ".rex", ".rexx", ".orx": herkömmliches Rexx-Programm
 - ".rxj": BSF4ooRexx-Programm, benutzt das Rexx-Paket "BSF.CLS"
 - ".rxo": OOO-Programm, benutzt das Rexx-Paket "UNO.CLS"
- "RexxPaws"
 - hält nach dem Beenden der Rexx-Programme das Terminalfenster offen, sodass man Ausgaben darin in Ruhe studieren kann
- Template-Dateien (Vorlagen)
 - für ".rex", ".rxj" und ".rxo"
 - erzeugen jeweils entsprechendes, einfaches Rexx-Programm

2010-Edition von BSF4ooRexx, 3

- beinhaltet Rexx-Paket "rgf_util2.rex"
 - beinhaltet zahlreiche und hilfreiche Routinen sowie Klassen, z.B.
 - zum einfacheren Sortieren von Sammlungen ("Collections"), `sort2()`
 - zum einfachen Auflisten der Bestandteile von Sammlungen, `dump2()`
 - Zeichenketten-BIFs ("2" an den Namen angehängt)
 - ignorieren Groß- und Kleinschreibung, z.B.
 - `say pos2("niXI", "da ist nixi drinnen")` liefert "8"
 - erlauben negative (!) Positions- und Längenangaben
 - `say substr2("niXI", -2)` liefert "XI" (die letzten beiden Buchstaben)
 - `say substr2("niXI", -2, -2)` liefert "ix"
 - zum Zerlegen von Zeichenketten in Wörtern, die z.B. nur aus zusammenhängenden Buchstaben bestehen
 - damit möglich, Wörter mit deutschen Sonderzeichen zu verarbeiten
 - Dokumentation (mit Beispielen): *"BSF4ooRexx → Information"*
 - "UNO.CLS" lädt das Paket "rgf_util2.rex" automatisch

Demonstration

- Installation von BSF4ooRexx
- Erstelltes Menüsystem
 - BSF4ooRexx-Verzeichnisse
- Dateiassoziationen
- Beispiele aus dem "**samples**"-Verzeichnis, soweit Zeit da
 - Menü "*BSF4ooRexx* → *Samples*"
 - Zahlreiche kommentierte Lehrbeispiele, die Java-Klassen benutzen, die XML-Dateien verarbeiten, die OOo steuern, ...

Zusammenfassung und Ausblick

- Neue Installation mit eigenem Menü und Dateiassoziationen!
- 100 %-ig rückwärtskompatibel mit Vorgänger "BSF4Rexx"
 - ältere BSF4Rexx-Programme laufen unverändert mit "BSF4ooRexx"
- Stellt Rexx die "größte Funktionsbibliothek der Welt" einfach zur Verfügung, die de facto auf jedem Rechner installiert ist:
 - **Java** !
 - Somit stehen Rexx-Programmierern sämtliche Java-Klassenbibliotheken der Welt *plattformunabhängig* zur Verfügung!
- Alle modernen Programme, die über Java-Schnittstellen verfügen, können von Rexx aus direkt angesteuert werden
 - Z.B. "OpenOffice.org"
- Rexx-Programme werden damit inhärent **plattformunabhängig!**

URLs

- Rexx Language Association (RexxLA)
 - <http://www.RexxLA.org>
- Homepage von ooRexx
 - <http://www.ooRexx.org>
- Build-Seite von ooRexx
 - <http://build.ooRexx.org>
- Homepage von BSF4ooRexx
 - <http://wi.wu.ac.at/rgf/rexx/bsf4oorex/current>
- Studentenarbeiten zu ooRexx, BSF4Rexx, BSF4ooRexx und OOo
 - <http://wi.wu.ac.at/rgf/diplomarbeiten/>