



FAKULTÄT
FÜR INFORMATIK
Faculty of Informatics

Programmiersprachen
LVA 185.208, VU, 3 ECTS, 2013 S



2. Übungsaufgabe

Überblick:

Entwickeln Sie ein Werkzeug zur Bearbeitung von Dateien nach einem festlegbaren Schema in einer dynamisch typisierten Scripting-Sprache wie Perl, PHP, Ruby, etc. Das Werkzeug soll einen in einer Konfigurationsdatei festgelegten Ablauf abarbeiten. Eine solche Konfigurationsdatei erfüllt eine ähnliche Aufgabe wie ein Shell-Script.

Konfigurationsdatei:

Folgende Aktionen sollen in der Konfigurationsdatei festlegbar sein:

- Eine *primitive Aktion* ruft ein Programm auf bzw. setzt ein (Unix-)Kommando ab. Sie kann auch Parameter enthalten (die als einfache Parameter interpretiert werden, siehe unten). Es soll auch möglich sein, das Werkzeug selbst (mit derselben oder einer anderen Konfigurationsdatei) als primitive Aktion rekursiv auszuführen. Die Ausführung einer primitiven Aktion ist erfolgreich, wenn der Programmaufruf mit dem Wert 0 terminiert, sonst ist sie gescheitert. Alle anderen (nicht-primitiven) Aktionen erweitern oder kombinieren einfachere Aktionen.
- Eine *Sequenz* besteht aus mehreren Aktionen, die in der gegebenen Reihenfolge hintereinander ausgeführt werden. Wenn die Ausführung einer Aktion scheitert, werden die weiteren Aktionen nicht mehr ausgeführt, und die Ausführung der Sequenz gilt als gescheitert, sonst als erfolgreich.
- Eine *Alternative* besteht aus mehreren Aktionen, von denen nur die erste immer ausgeführt wird. Weitere Aktionen werden nur ausgeführt, wenn alle davor stehenden Aktionen gescheitert sind – quasi als Gegenstück zur Sequenz. Die Ausführung einer Alternative ist genau dann erfolgreich, wenn die Ausführung einer darin enthaltenen Alternative erfolgreich ist.
- Eine *Mengen-Aktion* ist eine Aktion, die von genau einem Parameter abhängt, welcher als Mengen-Parameter interpretiert wird (siehe unten). (Sind mehrere Parameter zu verwenden, handelt es sich um mehrere, ineinander geschachtelte Mengen-Aktionen.) Für jeden Wert im Parameter wird die Aktion einmal ausgeführt. Die Mengen-Aktion ist genau dann erfolgreich, wenn jede Ausführung der Aktion erfolgreich ist.

Jeder Parameter hat einen Namen und enthält eine Zeichenkette. Ein Parameter ist auf folgende Arten verwendbar:

- Als *einfacher Parameter* in einer primitiven Aktion entspricht der

einzigste Wert des Parameters der ganzen Zeichenkette.

- Als *Mengen-Parameter* in einer Mengen-Aktion wird der Inhalt der Zeichenkette als Menge von durch White-Space voneinander getrennten Werten betrachtet. Jeder Wert enthält keinen White-Space, und der Mengen-Parameter kann keinen, einen oder mehrere Werte enthalten.

Alle Parameter sollen in der Konfigurationsdatei deklariert werden. Die Initialisierung der Parameter soll auf folgende Arten erfolgen können:

- Werte der Parameter werden als Kommandozeilenargumente beim Aufruf des Werkzeugs mitgegeben.
- Parameterwerte ergeben sich einfach durch Zuweisung einer Zeichenkette in der Konfigurationsdatei.
- Parameter werden mit den Namen aller existierenden Dateien initialisiert, die einem vorgegebenen Muster entsprechen. Muster können beispielsweise durch reguläre Ausdrücke festgelegt werden. Solche Parameter sind vor allem in Mengen-Aktionen sinnvoll.
- Parameterwerte ergeben sich durch Kopieren anderer Parameter, wobei alle Zeichenfolgen, die einem vorgegebenen Muster (z.B. einem regulären Ausdruck) entsprechen, durch eine vorgegebene Zeichenfolge ersetzt werden.

Weitere Möglichkeiten zur Initialisierung von Parametern sind erlaubt, aber zur Lösung der Aufgabe nicht unbedingt notwendig.

Die Syntax und Details der Semantik von Konfigurationsdateien sollen Sie selbst festlegen.

Eine Konfigurationsdatei enthält in gewisser Weise ein einfaches Programm. Ihre Aufgabe besteht darin, Details der Sprache festzulegen und einen einfachen Interpreter dafür zu implementieren. Bitte halten Sie die Syntax und Semantik so einfach, dass die Aufgabe leicht ohne spezielle Compilerbau-Werkzeuge lösbar ist.

Anwendungsbeispiele:

Obige Beschreibung ist abstrakt gehalten. Hier werden einige Anwendungsbeispiele angerissen, die den Zweck klarer umschreiben und als Testfälle dienen können.

- Das Werkzeug überprüft, ob neue Dateien in einem Eingangs-Ordner erstellt wurden, verschiebt Dateien mit bestimmten Endungen im Namen in andere Ordner und führt ein Protokoll über verschobene Dateien.
- Angenommen, Sie wollen mehrere Texte, die Sie per Mail bekommen und in einem Verzeichnis nach einem bestimmten Namensschema abgelegt haben, zu einer Einheit zusammenfügen. Das Werkzeug kann Sie dabei unterstützen, indem es die entsprechend dem Namensschema zusammengehörenden Dateien zu großen Dateien zusammenfasst und je eine zusammengefasste Datei in einem Editor für weitere Korrekturen

öffnet.

- Das Werkzeug erstellt für alle Bilder in einem Ordner je zwei Kopien in unterschiedlichen Größen in anderen Ordnern und fügt für jedes Bild eine passende Zeile in eine bestehende HTML-Datei ein, sodass die Bilder in einer Übersichtstabelle und im Detail auf einem Web-Browser betrachtet werden können.

Implementieren Sie das Werkzeug und probieren Sie es in einigen einfachen Anwendungsfällen aus.

Anfang | HTML 4.01 | letzte Änderung: 2013-05-17 (Puntigam)