

Übung 1

Einführung

Hinweise: Einen Java-Compiler und die Java Virtual Machine finden Sie für verschiedene Betriebssysteme unter <http://java.sun.com/javase/downloads/index.jsp>.

In **Cygwin** ist Java, sofern es bei der Installation ausgewählt wurde (**gcj**) automatisch vorhanden. Wenn Sie die Java-Umgebung von SUN bevorzugen, benötigen Sie nur das Java Runtime Environment (JRE) mit dem Java-Interpreter **java** und das Java Development Kit (JDK) mit dem Java-Compiler **javac**. Bei diversen Linux-Distributionen (z.B. der **KNOPPIX Live-DVD**) ist Java automatisch integriert.

Entwicklungsumgebungen und Editoren für Java finden Sie z.B. unter

<http://www.eclipse.org/> oder (ältere Version, aber funktioniert trotzdem)

<http://www.javaeditor.de/>.

1. Machen Sie sich mit der Java-Umgebung vertraut (dies sollte eigentlich für Sie eine Wiederholung aus früheren Vorlesungen sein). Stimmen Programmsuchpfade, d.h. werden Java-Compiler (**javac**) und Runtime-Umgebung (**java**) gefunden (**PATH**-Variable)? Ist der **CLASSPATH** richtig gesetzt und enthält auch das aktuelle Verzeichnis „.“? Ist die gewünschte IDE (eclipse, J++, Java Editor..) funktionsfähig und kennt ebenfalls die genannten Programme und Suchpfade? Übersetzen Sie das allseits bekannte Java „Hello World“-Programm und lassen Sie es in der Java-Runtime bzw. der IDE ablaufen.

Das sollte auch ohne Musterlösung gehen.

2. Entwerfen Sie ein Programm, das folgende Beschreibung erfüllt¹:

„Ein Texterkennungs- und Sprachausgabe-System für das Einscannen und Vorlesen von deutschen Texten ist in Java zu implementieren. Dieses System soll, per Scanner oder alternativ als Bild, graphische Dateien (z.B. eingescannte Zeitungsartikel) auf das Vorhandensein von Textinhalten untersuchen, diese graphischen Texte in reine Textdateien umwandeln und diese anschließend mit ansatzweise korrekter Betonung der einzelnen Wörter sowie ganzer Sätze über eine vorhandene Soundkarte ausgeben.“

[Demonstration Sprachausgabe-Funktion Linux-Desktop]

*Wie Sie vielleicht nach einer Recherche anhand von **ähnlichen Projekten** erahnen können, ist es keine Aufgabe, die sich in ein paar Stunden erledigen lässt, ein komplettes Texterkennungs- und Sprachsynthese-System mit lexikalischer Textanalyse zu bauen, abgesehen von den erforderlichen sprachwissenschaftlichen Kenntnissen.*

¹ Sie sollten nicht mehr als eine halbe Stunde für diesen Aufgabenteil investieren. Warum?

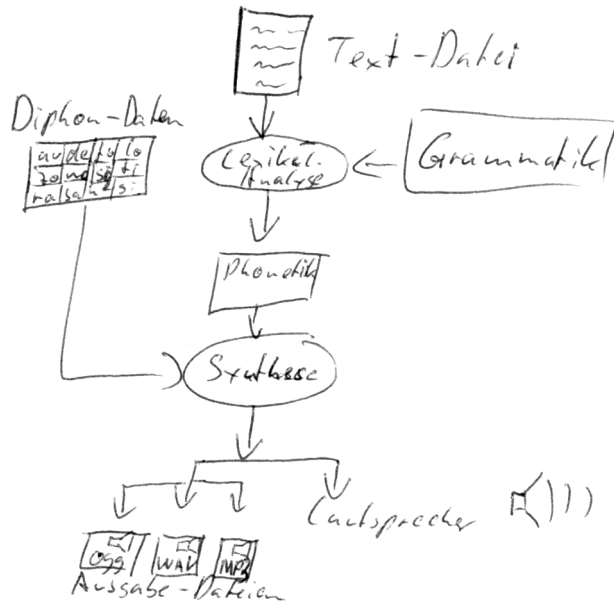
Auch, wenn Sie vorhandene Sprachsynthesysteme einbeziehen, dürfte sich der Aufwand als wesentlich höher herausstellen, als es die scheinbar sehr knappe Aufgabenstellung vermuten lässt.

Ziel dieser ersten Aufgabe, die einen einzelnen Entwickler klar überfordern würde, ist es, eine erste Einschätzung für den Aufwand zu entwickeln, den eine komplexe Software bedingt. Bei dieser Aufgabe dürfte er bei einer Neuentwicklung einer deutschen Diphon- oder Unit-Selection-basierten deutschen Sprachsynthese (diese beiden Algorithmen erfüllen die Forderung nach „Verständlichkeit“ sehr gut) bei etwa einem bis zwei Personenjahren liegen.

Entsprechendes gilt für die korrekte Erkennung von Texten in Grafikdateien, wofür unter Mitverwendung von Open Source Komponenten **OCROpus**, **tesseract** oder **gocr** verwendet werden könnten.

3. Führen Sie eine Anforderungsanalyse und Aufwandsabschätzung für die in der vorigen Aufgabe spezifizierte Software durch, mit Ansätzen in Form eines vorläufigen **Lastenhefts**.
 - (a) *Ausgangssituation und Zielsetzung: Neuerstellung eines Texterkennungs- und Sprachausgabe-Systems mit guter Verständlichkeit und Natürlichkeit der Sprache.*
 - (b) *Produkteinsatz: Vorlesen von eingescannten deutschen Buchtexten oder Zeitungen, z.B. während Autofahrt.*
 - (c) *Produktübersicht: Das Programm soll beliebige deutsche Texte, die als einfache Textdatei vorliegen, vorlesen bzw. über die Soundkarte des Rechners ausgeben. Der Benutzer kann die Eingabedatei per Kommandozeilenaufruf, oder dialoggesteuert dem Programm übergeben, und mittels Optionen bestimmte Einstellungen für die Sprachausgabe tätigen.*
 - (d) *Funktionale Anforderungen: Aus einer Eingabedatei bzw. einem eingescannten Blatt eines deutschen Textes soll durch die Software wahlweise eine WAV, Ogg- oder MP3-Datei erzeugt werden, oder Direktausgabe über Lautsprecher. An Systemvoraussetzungen soll eine Standard PC-Hardware „mittlerer Leistungsklasse“ zum Einsatz kommen, Betriebssystem und Java-Runtime sind vorinstalliert. Die Reaktionszeit soll unter 15 Sekunden für eine A4-Seite gescanntem Text betragen.*
 - (e) *Nichtfunktionale Anforderungen*
 - i. *Benutzbarkeit: Die Software soll auch von Laien bzw. Computer-Anfängern eingesetzt werden können, und darf nur ein Minimum an technischem Hintergrundwissen voraussetzen.*
 - ii. *Zuverlässigkeit: (Siehe auch Abnahmekriterien): Eine gewisse, wenn auch kleine, Anzahl von Aussprachefehlern wird toleriert, jedoch dürfen auch „exotische“ Texte nicht zu einem Programmabbruch führen, d.h. der Text soll bis auf Audio-mäßig nicht darstellbare Elemente komplett vorgelesen werden.*
 - iii. *Effizienz: Eine Lauffähigkeit auch auf alter Hardware ist wünschenswert, aktuelle kostengünstige Hardware ist zu unterstützen.*
 - iv. *Änderbarkeit: Spätere Erweiterungen wie Texterkennung gescannter Seiten sind in der Schnittstellendefinition zu berücksichtigen, so dass das Programm später entsprechend weiterentwickelt werden kann.*

- v. **Übertragbarkeit:** Die Software soll sich leicht in bestehende textverarbeitende Systeme integrieren lassen, und durch eine Open Source Lizenz in ihrer Weiterverwendung uneingeschränkt sein.
 - vi. **Risikoakzeptanz:** Die Software wird nicht in kritischen Bereichen eingesetzt, und kann daher früh produktiv eingesetzt und ggf. noch bis Projektende verbessert werden.
- (f) **Skizze des Entwicklungszyklus und der Systemarchitektur oder auch ein Struktogramm:** Hatten wir noch nicht in der Vorlesung, aber so könnte es aussehen:



- (g) **Lieferumfang:** Sowohl unter GNU/Linux als auch Windows installierbare JAVA-Klassen mit technischer Anleitung und kurzer Benutzereinführung.
- (h) **Abnahmekriterien:** Eine Anzahl vorgefertigter deutscher Texte (z.B. Zeitungsartikel, Ausschnitte aus Büchern oder Webseiten) soll eingescannt und vorgelesen werden, eine kleine Anzahl von Texterkennungs- und Aussprachefehlern (1-2 von 100 Wörtern) ist tolerierbar.

4. Richten Sie sich die ROBOCODE-Entwicklungsumgebung ein. Anleitung und Links hierfür finden Sie unter <http://knopper.net/bw/sep/>.

Sollte so aussehen:

