Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
Prof. Dr. W.-P. de Roever
Martin Steffen, Andreas Niemann, Melanie Zessack

Softwarepraktikum: Enigma

Sommersemester 2003 **Handout 8** Mai 2003

Handout 8: Genauere Spezifikation zu Aufgabe II (Gui)

Ausgabetermin: Mai 2003

Dieser Zettel enthält ein paar genauere Spezifikationen, was in der zweiten Teilaufgabe des Semesters abgeliefert werden soll.

Allgemeines

Die Aufgabe besteht darin eine *graphische Benutzeroberfläche* für den im ersten Teil programmierten Enigma-Automaten zu entwerfen und zu programmieren. Ganz allgemein soll die Oberfläche die Funktionsweise der Enigma einschließlich der inneren Verdrahtung und die Arbeit der Rotoren *visualisieren* und *Benutzerinteraktion* ermöglichen, wie sie auch bei der reellen Enigma vorgesehen waren (Walzen austauschen, Ringstellung verdrehen etc.)

Was die graphische Darstellung insbesondere der Walzenstellungen bestrifft, geben wir keine speziellen Varianten der Darstellung vor, aus denen sich jede Gruppe sich eine aussuchen soll. Dennoch sollten jede Gruppe den Vorschlag ihres Entwurfs, das Layout etc. vorher abklären.

Jetzt noch einige spezielle Punkte, auf die zu achten ist.

Plattform Die Oberfäche soll auf "*Swing*"-Basis entwickelt werden, d.h., insbesondere das in der Vorlesung vorgestellte Ereignismodell der Java2 Plattform verwenden.[1] Da Swing eventuell älteren Browsern Probleme bereitet, d.h., entweder gar nicht oder nur mit Plug-Ins lauffähig ist, ist das Kriterium für die Applets die Lauffähigkeit im *Appletviewer*. Als Standardgröße der Darstellung gehe man von eine Auflösung von *800x600* aus.

Enigma Für den zweiten (und danach auch den dritten) Teil ist es ausreichend, nur die *konkrete Enigma* für die Simulation, Visualisierung ... zu berücksichtigen. Die konkrete Engima muß wie ihr reales Vorbild, natürlich konfigurierbar sein. Wie im ersten Teil bereits nicht-graphisch zu realisiert, sind also folgende Bestandteile zu berücksichtigen:

- drei drehenden Walzen
- ein Vorrat an Walzen I – V, wobei je nur eine Verwendung erlaubt ist
- Reflektoren B und C

[1] Daß wir grundsätzlich unter Java-1.4 entwickeln, haben wir bereits zu Beginn festgelegt, aber man könnte im Prinzip auch damit "abwärtskompatibel" programmieren.

- Eintrittswalzen A und T
- Einstellbar soll daneben sein: Steckerbrett, Walzenstellung, Ringstellung

Visualisierung und I/O Wie erwähnt, soll neben den Verschlüsselungsfähigkeiten ("Klartext rein, Schlüsseltext raus" und umgekehrt) auch die interne *Funktionsweise* der Enigma aus der Darstellung ersichtlich werden, das Fortschalten der Walzen und die damit Verbundene Änderung des (virtuellen) "Stromflussess", die Anomalie .... Als Eingabe soll sowohl *Mauseingabe* durch Klicken auf eine virtuelle Tastatur als auch echte *Tastatureingabe* möglich sein.

Die *"stand-alone"*-variante soll *Laden* und *Speichern* von Dateien können. Es sollen dabei *2 Fileformate* unterstützt werden:

- eines mit *Konfigurationsheader* und
- eines ohne.

Der Konfigurationsheader wird so funktionieren wie die Dateiname bei den Beispiel des ersten Teils.

Anwendung und Applet Abgesehen von der Tatsache, daß Applets gewissen Einschränkungen unterworfen sind, sollen selbstverständlich Applet und Anwendung soviel Kode wie möglich teilen, insbesondere ist eine gemeinsame Hauptklasse vorzusehen. Man sollte beim "graphischen" Entwurf der Oberfläche bedenken, daß Applets "ein-fenstrig" sind.

Sonstiges Es reichte eine *sequentielle* Realisierung ("*single-threaded*"). Man sollte auch im Hinblick auf das *Endprodukt* am Ende des Semesters folgendes im Auge behalten: Das Endprodukt besteht aus einem "Applet" bzw. einer Applikation, und der Anwender interagiert mit der Anwendung über eine Seite bzw. er bekommt die Anwendung als fertig-verschnürtes *jar-Paket*.