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 6** — 16. April 2003

Handout 6: Genauere Spezifikation zu Aufgabe I (Enigma)

Ausgabetermin: 16. April 2003

Dieser Zettel enthält ein paar genauere Spezifikationen, was in der ersten Teilaufgabe des Semesters abgeliefert werden soll. Dies dient dazu einerseits leichter testen zu können, ob die Anforderungen erfüllt sind, und andererseits dazu, den Aufwand und den Umfang der Lösungen einigermaßen zu vereinheitlichen.

Anforderungen an die abstrakte Engima

- ein festes Alphabet mit 26 Buchstaben
- eine beliebige Anzahl von Walzen, sowie deren Mehrfachnutzung
- Verwendung von Walzen mit beliebiger Permutation und Kerbenanzahl
- Verwendung von festen Eintrittswalzen mit beliebiger Permutation
- Verwendung von festen Reflektoren mit beliebiger selbstinverser Permutation
- ein Steckerbrett für 0 bis 13 gültigen Steckerverbindungen.

• Überprüfen/Abfangen von "unsinnigen" Konfigurationen

Anforderungen an die konkrete Enigma

- Verwendung von genau drei Walzen
- Ausschließliche und einmalige Verwendung der bekannten Walzen I-V
- Ausschließliche Verwendung der Reflektoren B und C
- Ausschließliche Verwendung der Eintrittswalzen A und T

```
    | A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
----+-----------------------------------------------------
  A | A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
----+-----------------------------------------------------
  T | Q|W|E|R|T|Z|U|I|O|A|S|D|F|G|H|J|K|P|Y|X|C|V|B|N|M|L
```

Die oben angeführten Anforderungen sind beim Entwurf und der Implementierung zu berücksichtigen. Bei Unklarheiten oder Unsicherheit bitten wir, auch im eigenen Interesse, um vorherige Rücksprache mit dem Veranstalter oder einem der Hilfskräfte.

Testen der Implementierung

Zum Testen der Implementierung werden verschlüsslete *Beispieltexte* auf unserer Webseite bereitgestellt, mit denen man seine Implementierung auf eine möglichst hohe Korrektheit hin überprüfen kann. Die Beispieltexte beinhalten zur Einstellung der Enigma in der ersten Zeile eine entsprechende *Konfiguration.* Die Konfiguration ist dabei wie folgt zu lesen:

```
C145T-BKT020608-AGHIKL
|||||  |||||||||  ||||||
|||||  |||||||||  +++++++--- Steckerverbindungen mit
|||||  |||||||||            A<->G, H<->I und K<->L
|||||  |||||||||
|||||  |||++++++----------- Ringstellungen
|||||  |||                  1. Walze = 08, 2. Walze = 06
|||||  |||                  und 3. Walze = 02
|||||  |||
|||||  +++----------------- Grundstellung
|||||                       1. Walze = T, 2. Walze = K
|||||                       und 3. Walze = B
|||||
||||+--------------------- Eintrittswalze T
||||
|+++---------------------- 1. Walze = Walze V, 2. Walze = Walze IV
|                         und 3. Walze = Walze I
|
+------------------------ Umkehrwalze C
```

Soll keine Steckerverbindung gesetzt werden, so entfällt der letzte Teil einschließlich dem Gedankenstrich. Nach der Entschlüsselung sollte jeder der gewonnen Texte gleich und mit verstaendlichem Inhalt sein.

Für die erste Abgabe sollte ein "`make test`" erstellt werden, welches die acht Beispieltexte einliest, entschlüsselt und mit der neuen Endung "`.klartext`" wieder ausgibt.

Dateiformat

Wie das Format beim Speichern aussehen soll wurde bereits erläutert. An die Quelle werden weniger Anforderungen gestellt. Grundsätzlich soll jede Datei gelesen werden können, ohne dass es dabei zu Abstürzen des Programmes führt. Die sogenannten *Whitespaces* sollen dabei einfach ignoriert werden und Kleinbuchstaben in Grossbuchstaben umgewandelt werden. Tauchen daneben weiterhin Zeichen auf, die nicht verarbeitet werden können, so soll der Benutzer geeignet informiert werden.