

Aufgabe 7 – Verschlüsselung (Properties, IO/Streams)

Realisiert werden soll eine sehr einfache Verschlüsselung, bei der jeder Buchstabe des Alphabetes durch einen anderen ersetzt wird. Dazu werden 4 Textdateien Dateien erstellt – teils händisch, teils durch das Programm.

verschluesselung.conf:

Ist eine „Properties“-Datei (siehe Teil 4 aus IOBeispiele), kann händisch erstellt werden, wird am Programmbeginn eingelesen und liefert die Informationen für die Dateinamen. Folgende Einträge stehen in der Datei:

```
original = text.txt
schluessel = schluessel.txt
codiert = verschluesselterText.txt
decodiert = entschluesselterText.txt
```

D.h. das Programm erhält hieraus alle Infos – es sollen also bis auf den Namen dieser Konfigurationsdatei keine Dateinamen fix im Quelltext (Konstruktor,...) stehen.

text.txt: (oder ein beliebiger anderer Name, entsprechend dem Property „original“)

Enthält den zu verschlüsselnden Text: Ein normaler Text, bestehend aus Groß- und Kleinbuchstaben, der einfach mittels Texteditor erstellt werden kann.

schluessel.txt: (oder ein beliebiger anderer Name, entsprechend dem Property „schluessel“)

Definiert die Abbildung der Buchstaben: z.B.: $a \rightarrow x$, $b \rightarrow g$, aus $c \rightarrow w$, usw.

Das kann entweder in Form einer Properties-Datei umgesetzt werden oder in einem eigenen Format.

Wichtig: die Abbildung soll nur mittels Kleinbuchstaben beschrieben werden

codiert.txt: (oder ein beliebiger anderer Name, entsprechend dem Property „codiert“)

In dieser Datei wird der verschlüsselte Text gespeichert. Dabei sollen alle Buchstaben entsprechend der in obiger Schlüsseldatei definierten Abbildung geändert werden.

Wichtig: Die Groß- und Kleinschreibung soll erhalten bleiben. Zeichen für die keine Abbildung definiert wurde, werden unverändert übernommen. D.h. Satzzeichen, Sonderzeichen,... werden unverändert übernommen; eine eventuell leere Schlüsseldatei bewirkt, daß die verschlüsselte Datei eine Kopie des Originales ist.

decodiert.txt: (oder ein beliebiger anderen Name – entsprechend dem Property „decodiert“)

In dieser Datei wird der wieder entschlüsselte Text gespeichert.

Regeln: wie bei der Verschlüsselung, nur die Abbildung ist dabei umgekehrt.

Vorgabe für den Entwurf der Klassen / Methoden:

- Der **Konstruktor** liest die Konfiguration ein (die „Properties“ aus verschluesselung.conf).

Folgende Methoden sollen erstellt werden:

- Zum **Einlesen** einer Datei (Originaltext oder verschlüsselter Text)
- Zum **Schreiben** einer Datei (verschlüsselter Text, entschlüsselter Text)
- Zum **Kodieren** (verwendet Einlesen / Schreiben)
- Zum **Dekodieren** (verwendet Einlesen / Schreiben)
- Zum **Vergleichen** zweier Dateien (Original mit entschlüsselter Datei)
Für diesen Vergleich sollen in der Methode gleichzeitig zwei Streams geöffnet werden und solange Zeichen für Zeichen verglichen werden, bis ein Unterschied gefunden wurde, oder das Dateiende erreicht wurde.
Vergleichbar mit dem Kommando **fc** (filecompare).

Exceptions:

Alle auftretenden Exceptions müssen abgefangen werden und in Form einer Textmeldung auf System.err ausgegeben werden. Kein direkter Stack-Trace!

Kommentare:

Alle Klassen & Methoden sind zu kommentieren!