Übungen zu Grundlagen der Informatik I
Sommersemester 2009
Michael Stürmer M. Sc., Dipl.-Inf. Marcus Prümmer, Bruno Escherl

Aufgabenblatt 4 vom 18.05.2009, Abgabe am 05.06.2009

Aufgabe 1: Verschlüsseln von Texten

Strings, Kommandozeilenparameter
16 Punkte

Die Vigenère-Verschlüsselung ist ein einfacher Verschlüsselungsalgorithmus, der nach dem Prinzip der polyalphabetischen Ersetzungschiffren arbeitet. Dabei wird ein Zeichen des zu verschlüsselnden Klartextes abhängig vom aktuellen Zeichen eines Schlüssels im Alphabet verschoben. Als Alphabet wird hierbei einer Folge aller Großbuchstaben gefolgt von allen Kleinbuchstaben angenommen: "ABC...XYZabc...xyz". Unbekannte Zeichen werden ohne Umwandlung in den verschlüsselten Text übernommen.

Im folgenden Beispiel wird der Klartext "Hallo Welt!" durch den Schlüssel "AKEY" chiffriert. Da die Schlüssellänge kürzer als der Klartext ist, wird der Schlüssel nach einmaligem Durchlauf wieder vom ersten Zeichen an angewendet.

```
Hallo Welt!
AKEYAKEYAKE
HkpJo aClD!
```

Die Verschiebung des einzelnen Zeichens ergibt sich dabei durch die Position des aktuellen Schlüsselzeichens im Alphabet. Das `A` des Schlüssels ist an Position 0 im Alphabet, d.h. das erste Zeichen `H` wird nicht verändert. Das `K` ist an Position 10, also wird das `a` um 10 Stellen im Alphabet nach rechts verschoben und man bekommt ein `k`. Wird durch das Verschieben das Alphabet verlassen, so wird durch die Modulo-Operation dafür gesorgt, dass am Anfang des Alphabets fortgefahren wird. Aus dem `t` wird so durch die Verschiebung um `K` ein `D`.

Die Entschlüsselung funktioniert analog dazu, nur dass die Verschiebung in umgekehrter Richtung erfolgt. Aus dem `k` des verschlüsselten Textes wird somit wieder das `a`.

Führen Sie nun folgende Schritte durch, um ihr eigenes Chiffre-Programm zu implementieren:

1. Legen Sie eine Datei `VigenereChiffre.java` mit einer main-Methode an.
2. Schreiben Sie eine Methode **encrypt()**, die den Klartext und den Schlüssel als Parameter übergeben bekommt. Als Ergebnis soll die Methode den nach oben geschildertem Prinzip verschlüsselten String zurückgeben.

   Hinweis: Sehen Sie sich die Methoden der String-Klasse genauer an, besonders diejenigen, die einen Zugriff auf einzelne Zeichen eines Strings sowie die Bestimmung der Position eines Zeichens innerhalb eines Strings erlauben. Benutzen Sie dazu die Methodenvorschau von Eclipse oder schlagen Sie in der Java-Dokumentation nach, die unter folgender URL erreichbar ist:
   http://java.sun.com/javase/6/docs/api/java/lang/String.html
3. Schreiben Sie analog dazu eine Methode **decrypt()**, die als Parameter einen verschlüsselten String sowie den Schlüssel übergeben bekommt. Geben Sie als Rückgabewert den entschlüsselten String zurück.

4. Ihr Programm soll als Kommandozeilenparameter einen Parameter `encrypt` vom Typ `boolean` unterstützen, der angibt, ob der eingelesene Text ver- oder entschlüsselt werden soll. Den Kommandozeilenparameter können Sie aus `String[] args` auslesen, das an die main-Funktion als Parameter übergeben wird. Um aus dem String-Wert einen boolean zu erzeugen, benutzen Sie das Unterprogramm `valueOf` der Klasse Boolean.

5. Definieren Sie außerdem einen String, in dem Sie den Schlüssel abspeichern.

6. Lesen Sie nun zeilenweise den Text ein, der umgewandelt werden soll. Benutzen Sie dazu die bereits bekannte Scanner-Klasse. Rufen Sie abhängig vom Parameter `encrypt` ihre `encrypt` bzw. `decrypt` Methode auf. Geben Sie jede umgewandelte Zeile auf `stdout` aus.

Zum Testen Ihres Programms laden Sie sich die Dateien `testEnc.txt` und `testDec.txt` von der GDI-Website herunter, die mit dem Passwort `geheim` chiffriert worden sind und speichern Sie diese in Ihrem Projektordner. Öffnen Sie eine Shell und wechseln sie mit `cd` in den Projektordner, in dem sich auch die Datei `VigenereChiffre.class` befindet. Führen Sie anschließend folgendes Kommando aus:

```
java VigenereChiffre false < testEnc.txt > myDecode.txt
```

Dieses Kommando führt Ihr Programm VigenereChiffre aus und gibt an, dass die Eingabe entschlüsselt werden soll. Die Eingabe der Datei erfolgt über `< testEnc.txt`, die Ausgabe ihres Programms wird über `> myDecode.txt` in die Datei `myDecode.txt` umgeleitet. Vergleichen Sie nun die Ausgabe Ihres Programms mit der Datei `testDec.txt` mit Hilfe des Befehles `diff`. War die Entschlüsselung erfolgreich, gibt diff nur eine leere Zeile aus. Die Verschlüsselung können Sie auf dieselbe Art und Weise testen.

Geben Sie die Datei `VigenereChiffre.java` über Ihre persönliche GdI-Übungsseite ab.

Wenn Sie die Aufgabe zusammen mit einem Übungspartner bearbeitet haben (s.o.), darf **nur einer von Ihnen** die Datei abgeben! Kontrollieren Sie vor dem Abgeben **UNBEDINGT** ob Ihre Matrikelnummer und ggf. die Ihres Übungspartners im Kommentar am Anfang der Datei angegeben ist. Sollten Sie es dennoch vergessen haben, können Sie die Kommentare einfügen und die Datei noch einmal abgeben.

Aufgabe 2: Bonusaufgabe

Sichere Passworteingabe, Dateioperationen
keine Punkte

Wer das Programm weiter ausbauen möchte, kann sich an folgender fortgeschrittener Bonusaufgabe versuchen. Geben Sie aber unbedingt die Version aus Aufgabe 1 ab!

Das feste Abspeichern des Passwortes im Quelltext ist natürlich nicht besonders sicher. Um dies zu beheben können Sie die Funktion `readPassword` von `System.console` benutzen. Die Benutzung der Pipe-Symbole $<$ und $>$ funktioniert damit aber nicht mehr. Geben Sie daher die Ein- und Ausgabedatei als weitere Kommandozeilenparameter an. Das Einlesen erfolgt über den Scanner, die Ausgabe über die Klasse `FileWriter`.

Verzweifeln Sie nicht, wenn Sie nicht alles verstehen, die meisten Fähigkeiten, die Sie für die Bonusaufgabe benötigen lernen Sie noch während des Semesters. Die Aufgabe ist aber mit ein wenig Eigeninitiative und Recherche in der Java Dokumentation machbar.