

**Informatik 1 - Translation Studies in Information Technology****2. PFLICHTÜBUNG**

Praktikum zur Vorlesung Informatik 1 (INF1), Bachelorstudiengang

Stephan Mechler, mechler@hs-mannheim.de, <http://www.informatik.hs-mannheim.de/~mechler>

**Abgabetermin:** 09.12.2024 (elektronisch) (05.12.2024 in der Vorlesung)

**Abnahmetermi**n: 12.12.2024

**Wichtig:**

**Für diese Übung gibt es 3 Bonus-Punkte für die Klausur.**

**AUFGABE 1**

a) (String\_3\_A\_NN.java):

Lesen Sie zwei Textzeilen von der Konsole ein. Verknüpfen Sie die beiden Textzeilen zu einer und geben Sie die Länge des zusammengesetzten Strings aus.

b) (String\_3\_B\_NN.java):

Lesen Sie einen Text von der Konsole ein. Geben Sie das 2te Zeichen des Strings aus.

c) (String\_3\_C\_NN.java):

Lesen Sie einen Text von der Konsole ein. Stellen Sie sicher, dass der Text mindestens 6 Zeichen hat. Konkatenieren Sie die letzten 4 und die ersten 2 Zeichen des eingegebenen Strings  
Geben Sie den neuen String auf der Konsole aus.

d) (String\_3\_D\_NN.java):

Lesen Sie einen Text von der Konsole ein. Stellen Sie sicher, dass der Text mindestens 16 Zeichen hat. Geben Sie den Teilstring von Zeichen 3 bis 6, den Teilstring 10 bis 14.  
Geben Sie die beiden Strings einzeln auf der Konsole aus.

e) (String\_3\_E\_NN.java):

Lesen Sie 2 Strings von der Konsole ein und vergleichen Sie deren Länge.  
Geben Sie den kürzeren String aus. Sind die beiden Strings gleich lang, geben Sie beide Strings auf der Konsole aus.

Suchen Sie ggf. nach geeigneten Methoden des String-Objekts, mit der Sie die Problemstellung geschickt lösen können.

## AUFGABE 2

(VERSCHLUESSLUNG\_NN.java):

Gesucht ist ein einfacher Verschlüsselungsalgorithmus: In einer Zeichenkette wird jeder Buchstabe durch ihren 2ten Vorgänger ersetzt:  $A \rightarrow D$ ,  $B \rightarrow E$ ,  $C \rightarrow F$ , ...  $W \rightarrow X$ ,  $X \rightarrow A$ ,  $Y \rightarrow B$ ,  $Z \rightarrow C$ . Alle anderen Zeichen bleiben unverändert, deutsche Umlaute brauchen ebenfalls nicht berücksichtigt werden. Genauso soll die Groß-, Kleinschreibung der Worte sich nicht ändern. So wird aus " Informatik " die Zeichenkette " Lqirupdwln". Die Methoden, die einen String auf einmal in ein Array of Character und umgekehrt umwandeln, dürfen nicht verwendet werden. Nutzen Sie nur Konzepte die Sie bereits kennen. Die Realisierung der Verschlüsselung ausschließlich mittels Bedienungen soll ebenfalls vermieden werden.

Entwickeln Sie zuerst ein Nassi-Shneiderman-Struktogramm und dann unter Eclipse ein Java-Programm für diesen Algorithmus.

Hinweis:

Der Datentyp char enthält ein Zeichen ( char ch = 'a'; )

Beispiel:

Ausgabe:

Informatik

Gldmpkyrgi

## AUFGABE 3

(LISTE\_NN.java):

Gesucht ist ein Programm zur Ausgabe von Texten.

Folgende zwei Listen sollen verwaltet werden:

Eagle, ape, bee, ant, moose, elephant, duck, donkey, fish, fly, dog, beetle, cat, cow, lion, mouse, nightingale, ox, horse, rat, sheep, butterfly, swan, pig, spider, bird, whale, wolf, worm, goat

Adler, Affe, Biene, Ameise, Elch, Elefant, Ente, Esel, Fisch, Fliege, Hund, Käfer, Katze, Kuh, Löwe, Maus, Nachtigal, Ochse, Pferd, Ratte, Schaf, Schmetterling, Schwan, Schwein, Spinne, Vogel, Wal, Wolf, Wurm, Ziege

Die Listen entsprechen einem Tier in englischer Sprache und der deutschen Übersetzung.

Der Benutzer soll folgende Auswahlmöglichkeiten per Tastatur erhalten:

Wie möchten Sie die Liste durchsuchen:

1. Index auswählen

2. Eingabe des Begriffs

3. Abbruch

Bei 2 erfolgt die Eingabe einer Zahl, wird die Übersetzung an der Indexstelle gesucht und die Anzahl der Buchstaben ausgegeben.

Bei 1 erfolgt die Eingabe einer Zeichenkette, wird die Übersetzung gesucht und die Anzahl der Buchstaben ausgegeben. **(freiwillig)**

Dies soll in beide Richtungen (DE <-> EN) funktionieren.

Beispiel 1:

**8**

(EN): donkey, 1\*d, 1\*o, 1\*n, 1\*k, 1\*e, 1\*y

(DE): Esel, 2\*e, 1\*s, 1\*l

Beispiel 2:

**donkey**

(EN): donkey, 1\*d, 1\*o, 1\*n, 1\*k, 1\*e, 1\*y

(DE): Esel, 2\*e, 1\*s, 1\*l

Bei 3 erfolgt der Abbruch des Programms

Die Eingabe von nicht auswertbaren Eingaben im Menü (bspw. Buchstaben) sollen zu einer Fehlermeldung und nicht zum Abbruch des Programms führen.

Entwickeln Sie zuerst ein Nassi-Shneiderman-Struktogramm und dann unter Eclipse ein Java-Programm, das die obige Aufgabenstellung realisiert.

Achtung! Das Nassi-Shneiderman-Struktogramm wird idR. Recht umfangreich.

## HINWEISE ZUR ABGABE

Alle Aufgaben werden während der angesetzten Übungsstunden im Rechenzentrum testiert. Bei der Abgabe (und vor der Abnahme) sind folgende Unterlagen zu schicken:

Bitte Nennen Sie in den Dokumenten immer Ihren Namen und Ihre Matrikelnummer!

- Pro Aufgabe ein Blatt (LSG) [PDF] auf dem Sie beschreiben, welche Herangehensweise Sie wählten, welche Probleme bzw. Schwierigkeiten sie lösen mussten.
- Pro Aufgabe ein Blatt (NSD) [PDF + NSD/STR] (oder mehrere Blätter) mit dem geforderten Nassi-Shneiderman-Struktogramm.

- Pro Aufgabe der Quellcodes [als JAVA-Datei ].

Benennen Sie Ihre Aufgabe nach dem vorgegebenen Namen in den runden Klammern und ersetzen NN durch Ihre Gruppennummer.

Beispiel:

Sortierung\_01.java

Sortierung\_01\_LSG.pdf

Sortierung\_01\_NSD.pdf

Zur Abnahme ist der Quellcode im Rechenzentrum vorzuführen.

Alle vorgenannten Dokumente müssen an diesem Tag am Rechner bereitstehen.

Die Aufgaben können auch einzeln testiert werden!

## LERNZIELE

- Entwurf von Algorithmen
- Elementare Datentypen
- Schleifen, String, Arrays