Einführung in die Programmierung

Prof. Dr. Detlef Krömker
Alexander Wolodkin

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Übungsblatt Nr. 6[1]

Ausgabe: 21.12.2017
Abgabe: 20.01.2018
16:00 Uhr

Ein-/Ausgabe in Dateien

Hinweis: *Dieses Aufgabenblatt ist mit einer Bearbeitungszeit von zwei Wochen dafür ausgelegt in einem Zweierteam gelöst zu werden. Neben der Implementierung wird bei der Bearbeitung Wert auf die folgenden Punkte gelegt: Dokumentation[2], Strukturierung und Einhalten des Style-Guides[3].*

Achtung: Achten Sie darauf die Variable `__author__` in **allen** Quellcode Dateien **korrekt** zu setzen. Abgaben die nicht dieser Vorgabe entsprechen werden **nicht** bewertet! Quellcode muss als *.py* Datei und alles Weitere als *.pdf* Datei abgegeben werden. Bei Abgaben mehrer Dateien müssen diese als *.zip* zusammengefasst werden. Abgaben, die nicht diesen Regeln entsprechen, werden ebenfalls **nicht** bewertet! Außerdem muss ihr Name in jeder abgegebenen *.pdf* Datei zu finden sein. Jeder von Ihnen MUSS bei EPR05 oder EPR06 für den Rest des Semesters den Partner wechseln! Sollten Sie dies nicht tun, werden die Aufgaben nicht bewertet.

Σ __ / **25**

Aufgabe 6.1: Einleitung

Wir wollen in diesem Übungsblatt vor allem das I/O in Dateien kennenlernen und einüben. Beiliegend finden Sie einen Unicode-Textdokument „Morgen_Kinder“ mit einem Gedicht von Erich Kästner. Speichern Sie dieses Dokument in irgendeinem Verzeichnis auf Ihrem Rechner (nicht unmittelbar im selben Verzeichnis wie der zu entwickelnde Programmcode).

Bedingung: *Die folgenden Programme sind objektorientiert (also insbesondere mit Klassen und Methoden) zu schreiben!*

Aufgabe 6.2: Dateien einlesen

Punkte: ____ / 2

Sie sollen in Aufgabe 6.2 als Beispiel das Textdokument „Morgen_Kinder“ einlesen. Schauen Sie sich hierzu das Kapitel 7 des Python Tutorials an (`https://docs.python.org/3/tutorial/inputoutput.html`). Später sollen Sie strukturierte Python-Daten in eine Datei ausgeben. Beschreiben Sie in einem kurzen Statement den Unterschied zwischen einer JSON (JavaScript Object Notation)-Datei und einer pickle-Datei (Python object serialization) aus Benutzersicht. Was würden Sie wann nutzen?

[1] Es dürfen keine Lösungen aus dem Skript, dem Internet oder anderen Quellen abgeschrieben werden. Diese Quellen dürfen nur mit Quellenangaben verwendet werden und es muss ein hinreichend großer Eigenanteil in den Lösungen deutlich zu erkennen sein.

[2] `http://www.python.org/dev/peps/pep-0257`

[3] `https://www.python.org/dev/peps/pep-0008/`

Aufgabe 6.3: Punkte: ____ / 7

Schreiben Sie ein Python 3.X Programm, das ein (beliebiges) Unicode Textdokument einliest und einfache Statistiken darauf ausführt:

- Anzahl der Wörter
- Anzahl der Anschläge
- Anzahl der Zeichen (ohne Leerzeichen)
- Häufigkeitsverteilung über die Zeichen (mit Leerzeichen)
- Häufigkeitsverteilung über die Wörter
- Mittelwert der Wortlänge

Wählen Sie geeignete Datentypen für die Statistik-Daten und strukturieren Sie ihr Programm objektorientiert. Als Beispiel nehmen Sie das Dokument „Morgen_Kinder“ und geben Sie hierfür die ermittelten Statistik-Werte an.

Aufgabe 6.4: Punkte: ____ / 6

Schreiben Sie die errechneten Statistik-Daten für „Morgen_Kinder“ in eine Datei zum Austausch mit Kolleg*innen. Was müssen und sollten (2 Angaben) Sie Ihrer Kolleg*in mitteilen, damit sie/er Ihre Ergebnisse möglichst problemlos wieder einlesen kann. Geben Sie diese Angaben für Ihre Datei an.

Aufgabe 6.5: Punkte: ____ / 5

Schreiben Sie ein Python 3.X Programm mit denselben Funktion wie in 6.2 und 6.3 zusammen unter Nutzung des file-open-Dialogs von Tkinter. Sie dürfen den Code der Aufgaben 6.2 und 6.3 natürlich benutzen und weiterentwickeln.

Aufgabe 6.6: Punkte: ____ / 5

Schreiben Sie ein Konsolen-User-Interface für das Problem aus 6.4, das einen ähnlichen Komfort für die Benutzer bietet wie der file-open-Dialog von Tkinter.