



Institut für Physikalische Chemie

**Übungen zum Methodenkurs „Programmierkonzepte in der Physikalischen Chemie“
im WS 2013/2014**

Dr. Till Biskup

— Aufgabenblatt 6 vom 05.02.2014 —

Aufgabe 6—1

Schreiben Sie eine Funktion (ggf. mit Unterfunktionen), die eine Infodatei einliest, parst und als hierarchische Struktur (`struct`) zurückgibt. Die Schnittstelle ist in Listing 1 gezeigt. Eine entsprechende Vorlage für eine generelle Infodatei können Sie von der Internetseite¹ des Kurses herunterladen.

Listing 1: Schnittstelle einer Funktion zum Einlesen und Parsen von Infodateien

```
metaData = readInfoFile(filename);
```

Die Funktion soll möglichst robust programmiert sein und den übergebenen Dateinamen auf Gültigkeit und die korrespondierende Datei auf Existenz überprüfen. Ferner soll die Funktion Dateinamen ohne Endung erkennen und automatisch die Endung „.info“ ergänzen.

Vergessen Sie nicht den Kommentarkopf der Funktion mit Schnittstellendokumentation und Urheberrechtshinweis/letztem Änderungsdatum.

Aufgabe 6—2

Überlegen Sie sich, welche Informationen mindestens in jeder Form von Infodatei enthalten sein sollten. Entwerfen Sie entsprechende Blöcke für eine minimale Infodatei.

Achten Sie darauf, einerseits die Zahl der Felder nicht unnötig zu vergrößern, andererseits nicht zu viel Information in einem einzelnen Feld zu verpacken. Feldspezifisches Parsen führt nur zu unnötigen Komplikationen des Parsers.

Aufgabe 6—3

Entwerfen Sie eine Infodatei für UV/vis-Messungen. Gehen Sie von einem kommerziellen Gerät aus, das austauschbare Bestandteile (Lampen, Küvettenhalter, Kühlsystem, Rührer) hat und entsprechend unterschiedlich konfiguriert werden kann.

Aufgabe 6—4

Welche grundlegenden Möglichkeiten haben textbasierte Nutzerschnittstellen zur Interaktion mit dem Nutzer?

Welche Funktionen stellt Matlab für diese Zwecke bereit? Warum kann es sinnvoll sein, eigene Funktionen zu schreiben, die auf die Matlab-Funktionen zurückgreifen, diese aber erweitern?

¹<http://till-biskup.de/de/lehre/programmierkonzepte/>

Aufgabe 6—5

Entwerfen Sie eine Funktion für Auswahlmenüs für eine textbasierte Nutzerschnittstelle. Übergabeparameter sollen die Liste von Optionen sowohl eine Reihe optionaler Parameter als Schlüssel-Wert-Paare sein (Listing 2). Verwenden Sie den `inputParser` zur Implementierung der optionalen Parameter.

Listing 2: Schnittstelle einer Funktion zur Erzeugung von Auswahlmenüs für Matlab-CLIs

```
answer = cliMenu(options, <parameter>, <option>)
```

Der Eingabeparameter `options` ist ein `cell array` mit zwei Spalten, der Rückgabeparameter `answer` ein `string` mit der vom Nutzer gewählten Option. Optionale Parameter (Typen in Klammern):

- `title` (string)
Titel, der oberhalb des Menüs angezeigt wird.
Wenn nicht angegeben: „Please choose an option:“.
- `prompt` (string)
Text, der vor dem Cursor für die Nutzereingabe ausgegeben wird.
Wenn nicht angegeben: „Your choice:“
Bei vorgewählter Option soll diese angezeigt werden, z.B. „Your choice (default: [a]):“.
- `default` (string)
Option, die ausgewählt wird, wenn der Nutzer die Return-Taste drückt.
Wenn nicht angegeben: leer

Die Funktion soll für ein einheitliches und sauber ausgerichtetes Layout (wie in Listing 3 gezeigt) sorgen.

Listing 3: Beispiel einer Ausgabe der zu schreibenden Funktion

```
Please chose simulation parameters
[1] Overall inhomogeneous linewidth Gaussian (careful when using strains!)
[2] Overall homogeneous linewidth Lorentzian (careful when using strains!)
[3] local inhomogenous linewidth, D strain (don't use with g strain!)
[4] local inhomogenous linewidth, E strain (don't use with g strain!)
[5] g strain in x direction (don't use with D strain!)
[6] g strain in y direction (don't use with D strain!)
[7] g strain in z direction (don't use with D strain!)
[8] Number of orientations in powder between 0 and 90 deg
[9] Refinement factor for interpolation of orientational grid
[10] No additional parameters
Your choice (default: [10]):
```

Darüber hinaus soll ggf. die vorausgewählte Option angezeigt werden und durch einfaches Drücken der Return-Taste auswählbar sein. Die Funktion soll die (interaktive) Nutzereingabe auf Gültigkeit überprüfen und so lange zur Auswahl zurückkehren, bis eine gültige Option ausgewählt wurde, die dann von der Funktion zurückgegeben wird.

Zusatzaufgabe Implementieren Sie die Möglichkeit, mehrere Parameter auswählen zu können (für das in Listing 3 gezeigte Beispiel sehr sinnvoll) und fügen Sie dafür einen zusätzlichen optionalen Parameter `multiple` (mit dem Typ `boolean`) hinzu. Berücksichtigen Sie diese Möglichkeit entsprechend bei der Angabe vorausgewählter Parameter.