

Übungsblatt - Vorbereitung

FlexNow

Für die Klausurzulassung ist das erfolgreiche Absolvieren der Übungen erforderlich. Wenn Sie an den Übungen teilnehmen möchten, müssen Sie sich bis **Di., 25.04.2023, 23:55 Uhr** in FlexNow zum Übungsmodul **B.Inf.1102.Ue: Grundlagen der Praktischen Informatik - Übung** anmelden.

Übung

Markdown und AsciiMath

Einfacher Text (*plain text*) besteht nur aus Zeichen eines feststehenden Zeichensatzes (z.B. ASCII, ISO-Latin-15, UTF-8). Die Gliederung eines einfachen Textes ist nur durch die Steuerzeichen möglich (z.B. Tabulator, Zeilenumbruch, Seitenwechsel), die im Zeichensatz enthalten sind. Schriftzeichen verschiedener Sprachen oder Symbole (z.B. Klammern, Operatoren aus Arithmetik oder Mengenlehre) stehen nur zur Verfügung, wenn der Zeichensatz diese enthält. Formatierung (z.B. Fett, Kursiv, Hochstellen, Tiefstellen) ist nicht möglich.

Eine Auszeichnungssprache (*markup language*) kann für die Gliederung und Formatierung von einfachem Text verwendet werden. Weiterhin ist es möglich nicht im Zeichensatz enthaltenen Schriftzeichen und Symbole zu kodieren. Dazu werden ausschließlich Zeichenfolgen des Zeichensatzes verwendet.

Für die Gliederung und Formatierung von einfachen Texten mit dem Zeichensatz ASCII (*American Standard Code for Information Interchange*) (https://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange) benutzen wir die Auszeichnungssprache **Markdown** (<https://markdown.de/>) und für zusätzliche Schriftzeichen und Symbole **AsciiMath** (<http://asciimath.org/>)

Beispiel

Die Datei `alternativen.md`, die dem Anfang des Abschnitts 1.4 aus dem Skript entspricht, umgesetzt mit Markdown und AsciiMath, finden Sie in der Stud.IP-Veranstaltung *Grundlagen der Praktischen Informatik (Informatik II)* unter *Dateien→Übung→uebung00-data*.

```
# Grundlagen der Praktischen Informatik

## 1. FUNKTIONALE PROGRAMMIERUNG

### 1.4 Alternativen

Betrachten wir folgende Definition des Betrags einer ganzen Zahl.
```

```
$"abs"(x) = {(-x,"wenn",x < 0),(x,"sonst",):}$
```

Um diese Definition umzusetzen, bietet sich ein bedingter Ausdruck an.

```
**if** *[test]* **then** *[expr_if]* **else** *[expr_else]*
```

Anhängig vom *[test]* (ein Ausdruck der nach ‘Bool‘ ausgewertet wird) nimmt der Ausdruck für

- *[test]* ‘== True‘ den Wert von *[expr_if]* an.
- *[test]* ‘== False‘ den Wert von *[expr_else]* an.

```
'''  
absolute :: Int -> Int  
absolute x = if x < 0 then -x else x  
'''
```

Bemerkungen

- Ein Absatz besteht aus einer oder aus mehreren Textzeilen, Markdown ignoriert Zeilenumbrüche in Absätzen, d.h. damit Gliederungen, z.B. Listen, erkannt werden können, müssen diese mit einer Leerzeile eingeleitet werden.
- Um AsciiMath in Markdown einzubetten werden `$`-Zeichen verwendet und keine ‘-Zeichen (*backticks*). In Markdown zeichnet man *inline code* mit *backticks* aus.

Sie können `alternativen.md` mit dem in Stud.IP *Grundlagen der Praktischen Informatik (Informatik II)* unter Lernmodule hinterlegten ILIAS-Test *GdPI 00 - Markdown+AsciiMath-Umwandler Vorschau* testen.

In Stud.IP *Grundlagen der Praktischen Informatik (Informatik II)* unter *Dateien→Sonstiges* sind die Bash-Skripte `mdjs2html` und `htmljs2pdf` hinterlegt. Damit kann aus `alternativen.md` eine *html* und daraus eine *pdf* Datei erzeugt werden. Um die Skripte starten zu können, müssen sie nach dem Download als ausführbar gekennzeichnet werden. Dazu können Sie folgende Befehle verwenden.

```
chmod u+x mdjs2html  
chmod u+x htmljs2pdf
```

Dann kann `alternativen.md` wie folgt verarbeitet werden.

```
./mdjs2html alternativen.md  
./htmljs2pdf alternativen.html
```

Beide Skripte sind im Rechnerpool der Informatik lauffähig. Das Skript `mdjs2html` lädt JavaScript und CSS (*Cascading Style Sheets*) von <https://cdnjs.cloudflare.com/> nach. Für `htmljs2pdf` muss `chromium` installiert sein.

In der Regel ist eine Umwandlung in *html* zur Kontrolle der eigenen Lösung, die dann im Browser betrachtet werden kann, bereits ausreichend. Soll trotzdem ein *pdf* erstellt werden, können Sie alternativ zu `htmljs2pdf` auch die *print*-Funktion Ihres Browsers verwenden.

Achtung

Sie sollten die **Lösungen** für die Markdown+AsciiMath-Aufgaben immer **offline anfertigen** und mit dem ILIAS-Test oder ebenfalls offline kontrollieren.

Die Lösungen sollte **nicht** direkt im ILIAS-Test angefertigt werden, weil kein automatisches Speicher stattfindet. Beim Ablauf des Timers, einem technische Problem, etc. ist dann alles verloren, was Sie bis dahin eingegeben haben.

Es ist **wichtig**, dass Sie nach dem Kopieren Ihrer Lösung in den ILIAS-Test diesen mit der Schaltfläche **Test beenden** beenden, damit die Daten ins System übertragen werden.

Aufgabe

Falls Sie noch nicht mit Markdown und AsciiMath gearbeitet habe, schauen Sie sich das Video *Informatik I - Markdown und AsciiMath* an, das in Stud.IP unter *Grundlagen der Praktischen Informatik (Informatik II) → Medien* hinterlegt ist.

Testen Sie `alternativen.md` mit dem ILIAS-Test *GdPI 00 - Markdown+AsciiMath-Umwandler*, sowie den Bash-Skripten `mdjs2html` und `htmljs2pdf` bzw. Ihrem Browser.

Machen Sie sich mit den Möglichkeiten von **Markdown** (<https://markdown.de/>) und **AsciiMath** (<http://asciimath.org/>) vertraut.

Praktische Übung

Linux

Falls Sie noch nicht mit Linux gearbeitet haben, schauen Sie sich das Video *Informatik I - Linux-Einführung* an, das in Stud.IP unter *Grundlagen der Praktischen Informatik (Informatik II) → Medien* hinterlegt ist.

Remote Login

Es besteht, per SSH (*secure shell*), die Möglichkeit der entfernten Anmeldung (*remote login*) in den Rechnerpool des Instituts für Informatik (IfI) über folgenden Anmeldeserver.

`shell.stud.informatik.uni-goettingen.de`

Für die Anmeldung ist eine Zwei-Faktor-Authentifizierung nötig, Informationen darüber können Sie in folgenden Quellen finden.

- Stud.IP *Grundlagen der Praktischen Informatik (Informatik II) → Medien → Informatik I - IfI-Account anlegen und remote login ins IfI-Netzwerk (von M. Kosche)*
- Stud.IP *Grundlagen der Praktischen Informatik (Informatik II) → Sonstiges → Informatik I - Hinweise fuer Fernzugriff.txt*
- <https://user.informatik.uni-goettingen.de/~damm/info1/aktuell/2FA.html>
- <https://doc.informatik.uni-goettingen.de/wiki/index.php/Shell>

Testat der praktischen Übung

Die Lösung einer praktischen Übung wird in einem Testat, das *in Präsenz* oder *online* während einer Rechnerübung stattfindet, vorgestellt und direkt bewertet.

Testate in Präsenz finden im Rechnerpool des IfI statt.

Online Testate können *remote* im Rechnerpool stattfinden oder alternative auf einem lokal installierten Linux vorgeführt werden. Für Windows-Nutzer bietet sich dafür Ubuntu in einer virtuellen Maschine, z.B. VirtualBox, an.

Um eine praktische Übung testen zu lassen **müssen** Sie einen Termin in der zugehörigen Rechnerübung reservieren. Ein Testat ohne Termin ist nicht möglich. Details dazu werden noch bekannt gegeben.

Abgabe der praktischen Übung

Für eine praktische Übung werden **keine Dateien**, sondern nur **eine Prüfsumme** abgegeben.

- Erstellen Sie ein Archiv, das **alle Dateien** enthält, die Sie beim Testat vorstellen möchten.

- Beim Testat werden nur Dateien aus einem Archiv testiert, dessen Prüfsumme **exakt** der von Ihnen übermittelten Prüfsumme entspricht.
- Berechnen Sie die Prüfsumme des Archivs mit dem **sha1sum** Befehl.

Aufgabe

1. Probieren Sie mit dem Test *GdPI 00 - Testat - Vorbereitung*, der in Stud.IP *Grundlagen der Praktischen Informatik (Informatik II) → Lernmodule* hinterlegt ist, das Übermitteln der Prüfsumme.
2. Machen Sie sich mit dem *Glasgow Haskell Compiler (GHC)* vertraut. Programmieren Sie die Haskell-Beispiele der Vorlesung nach und testen Sie diese mit **ghci**.