

Abgaberichtlinien für die Übung Programmierparadigmen

Martin Wittiger Felix Krause Timm Felden

15. Juni 2017

Dieses Dokument definiert die verbindlichen Richtlinien für alle Abgaben der Übung Programmierparadigmen. Missachtung dieser Richtlinien kann zu Punktabzug führen.

1 Allgemein

Die Lösungen für jedes Übungsblatt haben bis spätestens zum auf dem Blatt angegebenen Termin über ILIAS abgegeben zu werden. Pro Aufgabenblatt hat genau eine zip-Datei abgegeben zu werden, welche für jede bearbeitete Aufgabe genau eine Datei enthält. Diese enthaltenen Dateien haben die auf dem Blatt angegebenen Namen zu tragen. Jede Aufgabe definiert auf dem Aufgabenblatt den Dateinamen der korrespondierenden Lösung. Die zip-Datei hat sich mittels unzip auf Marvin entpacken zu lassen.

Fremde Arbeit als die eigene zu verkaufen, ist (zumindest) wissenschaftliches Fehlverhalten. Sollten wir identische oder offensichtlich aus derselben Quelle stammende Lösungen entdecken, erhalten Sie für das gesamte Blatt keine Punkte. Sollte eine solche Verfehlung nachträglich auffallen, werden wir Ihnen bereits vergebene Punkte abziehen. Dies gilt unabhängig davon, wer die Lösung tatsächlich erstellt hat. Verbreiten Sie Ihre Lösungen also nicht.

Lösungen sollten so kurz wie möglich formuliert und übersichtlich dargestellt sein. Unnötige Bestandteile wie ausführliche Kommentare werden nicht berücksichtigt. Geben Sie Ihren Lösungsweg nur an, wenn dies explizit gefordert ist.

Ihre Lösungen müssen am PC erstellt werden. Wer Abgaben einstellt, die ab fotografiert oder eingescannt sind, muss mit Punktabzug rechnen. Wird eine PDF-

Datei als Abgabe erwartet, so muss diese ein gültiges PDF-Dokument sein, welches mit dem Adobe Reader geöffnet werden kann. Die Lösungen von Programmieraufgaben müssen unabhängig von der Dateiendung immer als reine Textdatei vorliegen. Schreiben Sie also beispielsweise den Programmcode in eine Word-Datei und benennen die Dateiendung danach von `.doc` nach `.c` um, ist dies keine gültige Abgabe.

2 Programmieraufgaben

Quelltextdateien haben in UTF-8 kodiert zu sein, Zeilenumbrüche haben die Linux-Konvention zu verwenden (ein Zeilenumbruch entspricht einem Linefeed-Character, ASCII `0x0A`). Achten Sie darauf, dass die Dateien kein Byte-Order-Mark enthalten. Vermeiden Sie Umlaute und Sonderzeichen, wo dies möglich ist. Verwenden Sie keine Tabulator-Zeichen (ASCII `0x09`). Entfernen Sie vor Abgabe unnötige Programmteile wie Debugausgaben. Verändern Sie in vorgegebenen Dateien nur die vorgesehenen Bereiche. Diese sind stets klar markiert.

Es werden nur Abgaben berücksichtigt, die sich übersetzen lassen. Beachten Sie die unten aufgeführten sprachspezifischen Hinweise. Sollte Ihre Abgabe unangemessen langsam sein, werden Ihnen dafür voraussichtlich Punkte abgezogen.

Keine der gestellten Programmieraufgaben setzt tiefere Kenntnisse der jeweiligen Sprache voraus.

Ada

Der Referenzcompiler ist `gnat` auf Marvin. Ihre Programme müssen sich mit dem folgenden Befehl ohne Fehler und (Style-) Warnungen übersetzen lassen.

```
gnatmake -gnat05 -gnatwa -gnatwl -gnaty3abcefhiklmnp rt dateiname
```

C

Der Referenzcompiler ist `gcc` auf Marvin. Ihre Programme müssen sich mit dem folgenden Befehl ohne Fehler und Warnungen übersetzen lassen.

```
gcc -std=c99 -pedantic -Wall -Wextra -Werror dateiname
```

Hinweis: Es wird insbesondere Windows-Nutzern nahegelegt, mit dem Compiler auf marvin zu testen, da die Windows C-Standardbibliothek den Standard C99 nicht unterstützt. Auch die GCC-Portierung MinGW kann hier keine Abhilfe schaffen, da sie ebenfalls diese Standardbibliothek verwendet.

C++

Der Referenzcompiler ist gcc auf Marvin. Ihre Programme müssen sich mit dem folgenden Befehl ohne Fehler und Warnungen übersetzen lassen.

```
c++ -std=c++11 -Wall -Werror -pedantic -O3 dateiname
```

Haskell

Der Referenzcompiler ist ghc 7.8.4. Importieren Sie keine Module. Achten Sie ganz besonders darauf, keine Tabulator-Zeichen zu verwenden. Ihre Programme müssen sich mit dem folgenden Befehl ohne Fehler und Warnungen übersetzen lassen.

```
ghc -fwarn-tabs dateiname
```

Java

Der Referenzcompiler ist javac 1.7. Ihre Programme müssen sich mit dem folgenden Befehl ohne Fehler und Warnungen übersetzen lassen. Achten Sie insbesondere bei Aufgaben, bei denen weitere Java-Dateien als Zusatzquellen zur Verfügung gestellt werden, darauf, dass Sie Ihrer Abgabe *keine* package-Deklaration hinzufügen. Dies führt dazu, dass der Compiler die anderen Klassen nicht mehr findet.

```
javac -Xlint:all -Xlint:-serial -Werror dateiname
```

Prolog

Der Referenzcompiler ist der Online-Interpreter auf <http://swish.swi-prolog.org/>.

Scala

Der Referenzcompiler ist scalac 2.11.