

LiesMich

Diese Datei ist für TeilnehmerInnen der Lehrveranstaltung TB5-CPB an der Beuth Hochschule Berlin im Sommersemester 2014 gedacht.

Erste Schritte

Besorgen Sie sich (am besten in der ersten Übung der Lehrveranstaltung Compilerbau) einen USB-Stick, der den Ordner `cpb` mit dem folgenden Inhalt enthält:

```
cpb
  cyan-14076
    compiler
    example
    runtime
    about.txt
  projects
    01_jasmin
    02_bspJasmin
    03_auxFiles
    04_forTextpad
    alg10
    alg20
    ...
    alg33
    ancestor01
    ancestor02
    ...
    zahlen0Bis255
```

In diesem Semester wird die Gentle-Version **cyan-14076** verwendet. Die jeweils neuste Version können Sie von der Adresse <http://cyan.compilertools.net/> herunterladen. Dort finden Sie auch eine umfangreiche **Dokumentation** zum Gentle Compiler, die viele kleine, ausführbare Beispiele enthält.

Der Ordner `cyan-14076` enthält eine beta-Version des Gentle-Compilers. Der Gentle-Compiler ist ein Java-Programm, welches Gentle-Dateien (`.g`-Dateien) in Java-Dateien (`.java`-Dateien) und anschließend in Bytecode-Dateien (`.class`-Dateien) übersetzt. Dazu sollten der Java-Compiler `javac.exe` und der Java-Interpreter `java.exe` der Firma Sun/Oracle installiert und über die Umgebungsvariable `PATH` erreichbar sein.

Am Anfang der Gentle-Dokumentation finden Sie im Abschnitt `Getting Started` einen Unterabschnitt `A First Example`. **Führen Sie die dort beschriebenen Schritte aus** um um sich zu vergewissern, dass Ihre Compilerbau-Werkzeuge richtig funktionieren.

Projektordner

Ein Projektordner ist ein *direkter Unterordner* des Ordners `projects`. Die Ordner `ancestor01` und `alg10` sind Beispiele für *vorgegebene* Projektordner. Nach Bedarf können Sie weitere Projektordner anlegen. Die nummerierten Ordner (deren Namen mit `01_`, `02_`, `03_`, `04_` beginnen) sind *keine* Projektordner.

Kommandos

Der Ordner `projects` enthält (ausser zahlreichen Projektordnern und dieser `LiesMich`-Datei) zwei Skripte namens `gen.cmd` und `exe.cmd`. Wenn das aktuelle Arbeitsverzeichnis ein Projektordner ist, der die Gentle-Quelldatei `spec.g` und die Eingabedaten-Datei `t01` enthält, dann haben die folgenden Aufrufe die folgenden Wirkungen:

```
> call ..\gen.cmd spec.g
```

Aus der Datei `spec.g` wird mit dem Gentle-Compiler ein Java-Programm generiert.

```
> call ..\exe.cmd t01
```

Das zuletzt generierte Java-Programm wird ausgeführt und auf die Datei `t01` angewendet.

Wenn man die vorhandene Gentle-Version `cyan-14076` gegen eine neuere Version (etwa `cyan-14099`) austauschen will, muss man nur den Ordner `cyan-14076` durch den Ordner `cyan-14099` ersetzen und die beiden Skripte `gen.cmd` und `exe.cmd` entsprechend anpassen.

Relative Pfadnamen

Da in den hier beschriebenen Kommandos nur *relative* Pfadnamen verwendet wurden (z.B. `..\gen.cmd spec.g` oder `..\exe.cmd t01`), funktionieren sie auch dann, wenn man den Ordner `cpb` an eine andere Stelle kopiert oder wenn er auf einem Stick steht, dem auf verschiedenen Rechnern verschiedene Laufwerksbuchstaben (z.B. `I:` oder `K:` etc.) zugeordnet werden.

Test-Skripte in den Projektordnern `ancestor01` bis `ancestor07`

In jedem dieser Projektordner sollte es ein Skript `TST1.CMD` und eine Datei `EXPECTED1.OUT` geben. Das Skript wendet das aktuelle Programm (Parser bzw. Compiler) auf verschiedene korrekte und unkorrekte Eingaben an, sammelt die Ausgaben des aktuellen Programms in einer Datei namens `FOUND1.OUT`, vergleicht die Datei `EXPECTED1.OUT` mit `FOUND1.OUT` und meldet alle Differenzen. Falls (mehr als null) Differenzen gemeldet werden, enthält entweder die vorgegebene Datei `EXPECTED1.OUT` oder das (von Ihnen entwickelte) aktuell Programm noch Fehler.

Test-Skripte in den Projektordnern `alg30` bis `alg33`

In jedem dieser Projektordner sollte es zwei Skripte `TST1.CMD` und `TST2.CMD`, zwei Dateien `EXPECTED1.OUT` und `EXPECTED2.OUT` und eine korrekte `alg`-Quelldatei `TST1.ALG` geben.

Das Skript `TST1.CMD` wendet den aktuellen Compiler auf das *korrekte* `alg`-Quellprogramm `TST1.ALG` an, lenkt die Ausgabe des Compilers in eine Datei namens `FOUND1.OUT` um, vergleicht die Datei `EXPECTED1.OUT` mit `FOUND1.OUT` und meldet alle Differenzen.

Das Skript `TST2.CMD` wendet den aktuellen Compiler auf mehrere *unkorrekte* `alg`-Quellprogramme an, sammelt die Ausgaben des Compilers in einer Datei namens `FOUND2.OUT`, vergleicht die Datei `EXPECTED2.OUT` mit `FOUND2.OUT` und meldet alle Differenzen.

Falls von einem der beiden Skripte (mehr als null) Differenzen gemeldet werden, enthält entweder die betreffende (vorgegebene) `EXPECTED`-Datei oder der (von Ihnen entwickelte) aktuelle Compiler noch Fehler.

Test-Skripte in weiteren Projektordnern

Auch in einigen weiteren Projektordnern gibt es Testskripte namens `TST1.CMD` bzw. `TST2.CMD` und die dazugehörigen Dateien `EXPECTED1.OUT` bzw. `EXPECTED2.OUT`. Auch diese Dateien funktionieren wie oben beschrieben.