Prof. Dr. Thomas Breuel
Dipl.-Inf. Patrick Michel (AG ST)
Dipl.-Ing. Ludwig Schmidt-Hackenberg

TU Kaiserslautern
Fachbereich Informatik
AG Bildverstehen und Mustererkennung

Übungsblatt 1: Software-Entwicklung 1 (WS 2010/11)

Ausgabe: 26. Oktober 2010
Abgabe: per E-Mail bis 12.30 Uhr am Tag vor der ersten eigenen Übung
Abnahme: spätestens zwei Tage nach der Übungsstunde

Das erste Übungsblatt soll Sie mit grundlegenden Funktionalitäten vertraut machen, die für die Bearbeitung der nächsten Übungsblätter benötigt werden. **Jeder** Student muss bei der Abnahme durch den Tutor in der Lage sein, **alle** Lösungen zu demonstrieren.
Beachten Sie bitte folgende Punkte, bevor Sie mit der Bearbeitung des Übungsblatts beginnen:

1. Sofern noch nicht geschehen, beantragen Sie bitte einen Account beim SCI (Gebäude 48, Erdgeschoß). Die Abnahme der Übungen im Rahmen der Vorlesung *SE 1* erfolgt auf dem *SCI-Account*.
2. Sofern noch nicht geschehen, melden Sie sich bitte bei einer der Übungsgruppen zur Vorlesung an. Alle Details dazu sind auf der Internetpräsenz der AG unter `http://www.iupr.com` zu finden.

Zur Beantwortung von Fragen und Hilfe bei Problemen stehen Ihnen die Tutoren in der ersten Woche *zu den Zeiten der zehn Übungsgruppen* im Terminalraum von Gebäude 48 (48-211) zur Verfügung. Nehmen Sie diese Gelegenheit war, um sich mit Ihrer Arbeitsumgebung und den Werkzeugen vertraut zu machen!

Aufgabe 1 Die Shell

Neben den heute üblichen grafischen Benutzeroberflächen kann ein Anwender auch mittels der *Shell*[1] Programme ausführen und so mit dem Rechner interagieren. Dies geschieht durch Eingabe des Programmnamens, dem gegebenenfalls noch weitere Parameter folgen, gefolgt vom Drücken der `RETURN`-Taste.

Beispiel: Der Aufruf des Programms `ls` listet die Dateien im aktuellen Verzeichnis auf.

a) Machen Sie sich mit der Funktionalität der Kommandos `pwd`, `cd`, `mkdir`, `cp`, `mv` und `rm` durch Lesen der sogenannten *Man-Pages*[2] vertraut (die Man-Page zu einem bestimmten Kommando wird mittels

`man` <*Name des Kommandos*>

angezeigt). Beschreiben Sie die Funktionalität jedes Kommandos *kurz* mit eigenen Worten. Was bedeuten die Kürzel der Kommandos?

Tipp: Es lohnt sich neben dem eigentlichen Kommando auch gängige Parameter zu kennen. Besonders nützlich für das Kommando `ls` *sind beispielsweise* `-l`, `-h` *und* `-a`. *Sie lassen sich auch einfach kombinieren:* `ls -alh`.

Tipp: Schauen Sie sich in der Man-Page zu `rm` *auch den Parameter* `-r` *an, für* `mkdir` *den Parameter* `-p` *und finden Sie heraus was* `cd` *ohne jeglichen Parameter macht.*

b) Erläutern Sie die Prinzipien und den Aufbau, der hinter Dateiverzeichnissen in UNIX (-artigen) Betriebssystemen steht. Erläutern Sie insbesondere die Bedeutung von '.' und '..' im Kontext von Verzeichnissen.

c) Erläutern Sie den Unterschied zwischen *absoluten* und *relativen Verzeichnispfaden*.

[1] Manchmal wird, insbesondere im Kontext grafischer Benutzeroberflächen, auch vom sogenannten *Terminal* gesprochen.
[2] Die abkürzende Bezeichnung *Man-Page* steht für *Manual-Page*, also quasi die Anleitung zum jeweiligen Kommando.

Aufgabe 2 Der Editor GEdit

Für die Eingabe von Texten im Allgemeinen und Programmcode im Speziellen wird ein sogenannter *Editor* verwendet. Ein UNIX-System stellt in der Regel eine Vielzahl verschiedener Editoren zur Verfügung, so dass jeder Benutzer den zu den persönlichen Präferenzen am besten passenden auswählen kann. Im Rahmen der Veranstaltung *SE 1* soll für alle Programmieraufgaben der Editor GEdit verwendet werden.

a) Öffnen Sie eine neue Shell und führen Sie folgendes Kommando aus:

```
gedit hello.hs
```

Es öffnet sich der Editor GEdit und sie können die Datei `hello.hs` editieren. Hat diese Datei noch nicht existiert, können Sie sie nun durch Speichern in GEdit (*shortcut:* `STRG-S`) anlegen.

Geben Sie folgenden Text in GEdit ein und speichern Sie die Datei:

```
module Main where

helloText = "Hello" ++ " " ++ "World!"
main = putStrLn helloText
```

Übersetzen Sie die Datei in der Shell mit dem Haskell-Compiler durch Eingabe folgenden Kommandos:

```
ghc --make hello.hs
```

Ist kein Fehler in der Datei und dem Kommando, sehen Sie die folgende Ausgabe:

```
[1 of 1] Compiling Main             ( hello.hs, hello.o )
Linking hello ...
```

Führen Sie das neu erstellte Programm `hello` nun in der Shell aus durch Eingabe von:

```
./hello
```

Tipps zur Arbeit mit GEdit und dem Terminal:

- Kommandos wie `gedit`, die ein Programm starten, das sich nicht direkt beendet, blockieren das Terminal. Durch das Anfügen von `&` an ein Kommando können Sie das Terminal direkt weiter benutzen!
- Haben Sie bereits ein Programm gestartet, welches nun das Terminal blockiert, können Sie es mit `STRG-Z` kurz unterbrechen und mit dem Kommando `bg` im Hintergrund weiter laufen lassen. Damit erhalten Sie wieder die Kontrolle über das Terminal. (Das duale Kommando `fg` holt es wieder in den Vordergrund.)
- Reagiert ein Programm im Terminal nicht mehr auf Eingaben, oder haben sie zum Beispiel ein selbst geschriebenes Programm ausgeführt, das nicht mehr terminiert, können Sie es mit `STRG-C` beenden!
- Beim Entwickeln eines Programms lohnt es sich sowohl den Editor als auch das Terminal ständig offen zu halten. Änderungen am Programm können Sie schnell mit `STRG-S` speichern. Mit `ALT-TAB` können Sie zwischen Fenstern wechseln und so zum Terminal gelangen. Läuft dort zum Beispiel gerade der GHCi, können Sie Ihr Programm mit `:l` laden, bzw. mit `:r` erneut laden. (`:?` zeigt die Befehlsübersicht an.)
- Die meisten Terminals besitzen eine Historie, die Sie mit den Pfeil-Hoch bzw. Pfeil-Runter Tasten durchlaufen können. Damit können Sie zum Beispiel einen Aufruf des GHC, nach einer Änderung an Ihrem Programm, schnell erneut ausführen.
- Wir haben einige Zusatzbefehle für GEdit vorbereitet, die Ihnen beim Arbeiten damit helfen sollen. Sie können diese installieren, indem Sie einmalig das folgende Kommando im Terminal ausführen:

  ```
  /home/p_michel/install.sh
  ```

  Sie erhalten damit die folgenden drei zusätzlichen Befehle in GEdit (im *Tools*-Menü):
  - `STRG-R` (*Load in GHCi*) Öffnet ein neues Terminal, startet GHCi darin und lädt die aktuelle Datei.
  - `STRG-E` (*Compile and Run*) Die aktuelle Datei wird mit dem GHC übersetzt und falls erfolgreich direkt ausgeführt. Die Ausgabe Ihres Programms sehen sie in GEdit im unteren Fenster.
  - `STRG-T` (*Lookup in Hoogle*) Öffnet einen Firefox (bzw. ein neues Tab in einem bestehenden Firefox), indem das aktuelle Wort unter dem Textcursor in *Hoogle* gesucht wird. Hoogle ist eine Suchmaschine für die Standardbibliothek von Haskell, mit der Sie sich über Funktionen informieren können.

  Sind diese nicht direkt nach Installation verfügbar, öffnen Sie GEdit neu oder loggen Sie sich ggf. kurz aus.

b) Laden Sie die Datei `hello.hs` mit dem Haskell-Interpreter `GHCi` durch Eingabe von:

```
ghci hello.hs
```

Geben Sie im Interpreter das Wort `helloText` ein und bestätigen Sie mit der `RETURN`-Taste. Sie können den Interpreter mit dem Kommando `:quit` oder kurz `:q` wieder verlassen. Mit `:help` erhalten Sie einen Überblick über andere hilfreiche Kommandos des Interpreters.

c) Informieren Sie sich über das Kommando `wc` und wenden Sie dieses auf die von Ihnen erstellte Datei `hello.hs` an. Was wird ausgegeben? Was bedeutet das Kürzel `wc`?

Aufgabe 3 E-Mail (Einreichaufgabe)

Die einfachste Art, außerhalb von Übungsstunden und Abnahmen mit Ihrem Tutor in Kontakt zu treten, bieten E-Mails. Insbesondere sollen Sie sich über den Verlauf des Semesters bei jeglichen Problemen beim Lösen der Übungsaufgaben an Ihren Tutor wenden.

a) Wofür steht das Kürzel „RFC“? Was wird im RFC 2822 definiert?

Wichtig: Die sogenannte *Netiquette* stellt den (inoffiziellen) Verhaltenskodex unter anderem für die Kommunikation mittels E-Mails und Newsgroups dar und ist als RFC 1855 verfügbar. Machen Sie sich mit dem Inhalt des RFCs vertraut – insbesondere mit dem Abschnitt 2.1.1.

b) Falls Sie die E-Mail Adresse, die Sie zusammen mit Ihrem SCI-Account erhalten haben, noch nicht eingerichtet haben, können Sie dies mit dem Programm Thunderbird tun. Machen Sie sich in diesem Fall mit der Bedienung von Thunderbird vertraut. Wenn Sie Hilfe beim Einrichten brauchen, wenden Sie sich wie bei allen Fragen zu diesem Blatt zu den angegebenen Zeiten an einen Tutor im Terminalraum (48-211).

Senden Sie eine E-Mail mit dem Betreff „[SE-1] Übungsblatt 1“ an die E-Mail-Adresse Ihres Tutors, in der Sie in wenigen Sätzen eine Antwort auf die folgenden Fragen formulieren:

- In welcher Beziehung steht der Begriff *Softwaresystem* zu *technisches System*?
- Was sind typische Eigenschaften von Softwaresystemen?
- Was ist ein *Programm*?
- Wie alt ist das Internet?
- Was ist der Unterschied zwischen einer Webseite und einer Website?

c) Senden Sie eine weitere E-Mail an Ihren Tutor, welche die in Aufgabe 2a erstellte Textdatei *als Anhang* enthält. Im Laufe des Semesters wird es notwendig sein, manche Aufgaben auf diesem Weg bei Ihrem Tutor abzugeben.

Aufgabe 4 Basiswissen der Vorlesung

Kreuzen Sie an, ob folgende Aussagen wahr oder falsch sind. Falls eine Aussage nicht eindeutig ist, diskutieren Sie die verschiedenen Interpretationsmöglichkeiten.

Bereiten Sie diese Aufgabe bis zu Ihrer nächsten Übungsstunde vor, so dass Sie bei Unklarheiten nachfragen und die möglichen Antworten diskutieren können.

wahr	falsch	
☐	☐	Ein modernes Auto enthält keine Software.
☐	☐	Software verschleißt nicht durch Benutzung.
☐	☐	Ein Softwaresystem steht für sich alleine.
☐	☐	Es gibt Software, die beliebig lange läuft.
☐	☐	Ein Qualitätskriterium für Software ist ihre Wartbarkeit.
☐	☐	Software spielt keine Rolle im täglichen Leben.
☐	☐	Ein Softwaresystem kann Teil eines technischen Systems sein.
☐	☐	Das Verhalten von Softwaresystemen wird durch Programme beschrieben.
☐	☐	Programme können nicht in jeder Sprache verfasst werden.
☐	☐	Der erste Schritt beim Entwickeln von Software ist das Programmieren.
☐	☐	Folgende Anforderung ist nicht-funktional: Die Software soll quadratische Gleichungen lösen.
☐	☐	Folgende Anforderung ist nicht-funktional: Das Ergebnis einer Berechnung soll nach spätestens zwei Sekunden angezeigt werden.
☐	☐	Das Ergebnis der Entwurfsphase ist ein Modell der zu entwickelnden Software.
☐	☐	Nach der Installation eines Systems ist der Entwicklungsprozess abgeschlossen.
☐	☐	Es wird mehr Geld für Wartung bestehender Software verwendet als für die Entwicklung neuer Systeme.
☐	☐	Auch ein gutes Benutzerhandbuch reicht als Dokumentation eines Softwaresystems nicht aus.
☐	☐	Egal wie groß ein Softwaresystem ist, die Entwicklungsmethoden unterscheiden sich nicht.
☐	☐	Große Softwaresysteme werden von vielen Entwicklern gemeinsam erstellt.
☐	☐	Informatik beschäftigt sich mit der automatischen Verarbeitung von Informationen.