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

TU Kaiserslautern
Fachbereich Informatik
AG Bildverstehen und Mustererkennung

Lösungshinweise/-vorschläge zum Übungsblatt 2: Software-Entwicklung 1 (WS 2010/11)

Die Hinweise und Vorschläge in diesem Dokument sollen der Lösungsfindung dienen und erheben demnach weder Anspruch auf Vollständigkeit noch Korrektheit. Sollten Sie Fehler finden würden wir uns freuen wenn Sie uns diese mitteilen. (Kontaktdaten finden Sie auf unserer Webpräsenz.)

Aufgabe 1 Algorithmisches Beschreiben (Präsenzaufgabe)

[Um Platz zu sparen schreiben wir Befehle nicht untereinander, sondern trennen sie durch ein Komma und markieren Blöcke nicht durch Einrückung sondern runde Klammern.]

a) Teilen Sie den Anfangs-Kartenstapel (von *A*) in zwei gleich große Stapel (nach *B* und *C*) auf.
Wiederhole *A* (**Hole** *L A*, **Lege** *L B*, **Hole** *L A*, **Lege** *L C*)

b) Überführen Sie den Anfangs-Kartenstapel (von *A*) nach *B*, wobei wieder die ursprüngliche Reihenfolge der Karten gelten soll.
Wiederhole *A* (**Hole** *L A*, **Lege** *L C*), **Wiederhole** *C* (**Hole** *L C*, **Lege** *L B*)

c) Überführen Sie den Anfangs-Kartenstapel (von *A*) nach *B*, wobei wieder die ursprüngliche Reihenfolge der Karten gelten soll. Die unterste Karte des Anfangsstapels soll jedoch auf Stapel *C* liegen.
Wiederhole *A* (**Hole** *L A*, **Lege** *L C*), **Hole** *L C*, **Lege** *L A*, **Wiederhole** *C* (**Hole** *L C*, **Lege** *L B*), **Hole** *L A*, **Lege** *L C*

Aufgabe 2 Algorithmisches Beschreiben (Einreichaufgabe)

a) Suchen Sie aus dem Anfangs-Kartenstapel die größte Karte heraus, und legen Sie sie auf Stapel *B* ab.
Hole *L A*, **Wiederhole** *A* (**Hole** *R A*, **Vergleiche** (**Vertausche**) (), **Lege** *R C*), **Lege** *L B*

b) Suchen Sie aus dem Anfangs-Kartenstapel die größte Karte heraus, und legen Sie sie auf Stapel *B* ab, wobei die restlichen Karten wieder an Position *A* liegen sollen.

GROESSTE_*A*_*B* = **Hole** *L A*, **Wiederhole** *A* (**Hole** *R A*, **Vergleiche** (**Vertausche**) (), **Lege** *R C*), **Lege** *L B*, **Wiederhole** *C* (**Hole** *L C*, **Lege** *L A*)

c) Sortieren Sie den Anfangs-Kartenstapel nach absteigendem Wert und legen Sie ihn an Position *A* ab.

Wiederhole *A* GROESSTE_*A*_*B*, **Wiederhole** *B* *B*_NACH_*C*, **Wiederhole** *C* (**Hole** *R C*, **Lege** *R A*)

d) *(freiwillige Zusatzaufgabe*) Suchen Sie aus dem Anfangs-Kartenstapel die größte Karte heraus, und legen Sie sie auf Stapel *B* ab, wobei die restlichen Karten wieder *in der gleichen Ordnung* an Position *A* liegen sollen.
B_NACH_A = **Wiederhole** *B* (**Hole** *R B*, **Lege** *R A*)

C_NACH_A = **Wiederhole** *C* (**Hole** *R C*, **Lege** *R A*)

SCHLEIFE = **Wiederhole** *A* (**Hole** *R A* , **Vergleiche** (**Lege** *L B*, **Lege** *R C*, C_NACH_A, **Hole** *L A*) (**Lege** *R C*))

HAUPTPROGRAMM = **Hole** *L A*, SCHLEIFE, C_NACH_A, B_NACH_A, **Lege** *L B*

Aufgabe 3 Funktionen und Ausdrücke (Präsenzaufgabe)

a) Beschreiben Sie in einem Satz, welche Berechnung jede der obigen Funktionen durchführt.

- inc(x) inkrementiert x um 1.
- triple(x) verdreifacht x.
- add(x, y) addiert x und y.
- quadsum(x, y) berechnet $(x + y)^2$.
- half(x) liefert den ganzzahligen Anteil der Division von x durch 2.

b) Welche Ergebnisse liefert die Auswertung folgender Ausdrücke?

inc(99) = 100	quadsum(inc(2), add(3, triple(1))) = 81
add(13, 29) = 42	half(inc(12)) = 6
add(half(2), triple(2)) = 7	triple(half(add(3, inc(3)))) = 9

c) Geben Sie drei verschiedene Ausdrücke an, deren Wert 37 ist. Jeder Ausdruck soll mindestens zwei Funktionen verwenden, und jede Funktion muss mindestens einmal vorkommen.
Drei Beispiele von unendlich vielen: inc(triple(12)), inc(half(72)), add(quadsum(2, 4), 1)

d) Schreiben Sie die nachfolgenden Funktionen (sie können $+$, $-$, $*$, div und die Funktionen von oben verwenden):

1. times5(x): Multiplikation von x mit 5.
   times5(x) $= 5 * x$ oder times5(x) $= add(triple(x), add(x, x))$ oder times5(x) $= x * x * x * x * x$
2. modulo(x, y): Berechnet den Rest der ganzzahligen Division von x durch y.
   modulo(x, y) $= x - (x \text{ div } y) * y$

Aufgabe 4 Funktionen und Ausdrücke (Einreichaufgabe)

a) Erstellen Sie eine Textdatei mit den Definitionen aller fünf Funktionen von oben und führen Sie diese Datei mit dem Interpreter aus.

b) Überprüfen Sie, ob ihre Ergebnisse aus den Aufgaben 3b) und 3c) korrekt sind.

c) Definieren Sie die Funktionen `times5` und `modulo`. Verwenden Sie die Datei aus Teil a) und starten Sie entweder den Interpreter mit der geänderten Datei neu oder erzwingen Sie ein erneutes Laden der Datei im Interpreter mit dem Kommando `:reload` oder kurz `:r`.

```
times5(x) = 5 * x
modulo(x,y) = x - (x `div` y) * y
```

d) Schreiben Sie eine Funktion, welche die Fläche eines Rechtecks bestimmt. Benutzen Sie diese, um Funktionen zu schreiben, welche die Oberfläche von Quadern und Würfeln bestimmen.
Ein Würfel ist ein Quader bei dem alle Seiten gleich lang sind. Die Berechnung von Volumen und Oberfläche kann daher mit den Funktionen für den Quader erfolgen, indem man ihnen drei mal die gleiche Seitenlänge übergibt.

```
rect(x, y) = x * y
cuboid(x, y, z) = 2 * rect(x, y) + 2 * rect(y, z) + 2 * rect(x, z)
cube(x) = cuboid(x, x, x)
```

e) Gegeben seien folgende zwei Funktionen:

```
f :: Int -> Int
f(0) = 1
f(n) = n * f(n-1)
```

```
g :: Integer -> Integer
g(0) = 1
g(n) = n * g(n-1)
```

1. Welchen Wert liefern die Ausdrücke `f(0)` bis `f(6)`? 1, 1, 2, 6, 24, 120, 720 Versuchen Sie

herauszufinden, was diese Funktionen berechnen.
Beide berechnen die Fakultätsfunktion.

2. Was passiert, wenn Sie `f` und `g` mit größeren Zahlen aufrufen? Können Sie sich vorstellen warum?
Die Funktion `f` liefert nur bis `f(12)` korrekte Ergebnisse. Danach ist der Zahlenbereich überschritten und läuft über. Nach einer Folge von teils positiven, teils negativen Zahlen wird bei 32 und 33 `minBound` von `Int` erreicht, was ab 34 dazu führt, dass alle weiteren Funktionswerte 0 sind. Für `g` gilt dies nicht, da wir hier den unbeschränkten Typ `Integer` haben. Trotzdem gilt für beide Funktionen, dass deren Auswertung immer länger dauert für große Zahlen und für zu große mit einem Stack Overflow abbricht.

   Zu beachten ist, dass abhängig von der Maschine der Datentyp `Int` einen unterschiedlichen Wertebereich haben kann. Garantiert ist er von -2^{29} bis $2^{29} - 1$, den genauen Bereich kann man mit `minBound` und `maxBound` ermitteln. Auch die Größe vom Stack kann eingestellt werden, weshalb auch keine klare Schranke ab wann man einen Überlauf bekommt angegeben werden kann.

Aufgabe 5 Grundkonzepte von Softwaresystemen

Kreuzen Sie an, ob folgende Aussagen wahr oder falsch sind. Bereiten Sie diese Aufgabe bis zu Ihrer nächsten Übungsstunde vor, so dass Sie bei Unklarheiten nachfragen und die Antworten diskutieren können.

wahr	falsch	
☒	☐	Es gibt Softwaresysteme, die auf mehreren Rechnern verteilt laufen.
☒	☐	Entwurf und Programmierung stehen aufgrund ähnlicher Modelle und Konzepte zueinander in Beziehung.
☐	☒	Jede Programmiersprache besitzt eine formale Semantik.
☐	☒	Anforderungen werden immer in einer formalen Sprache beschrieben.
☒	☐	Da jede Programmiersprache ähnliche Konzepte mitbringt, fängt man beim Lernen einer neuen Programmiersprache nicht wieder bei Null an. *Die grundlegenden Konzepte sind programmiersprachenunabhängig. Natürlich bietet nicht jede Sprache alle in der Vorlesung als elementar aufgezählten Konzepte und in vielen Sprachen werden weitere Konzepte angeboten, aber die Kenntnis der elementaren Konzepte bildet einen Grundstock für quasi alle Programmiersprachen.*
☒	☐	Für eine kontextfreie Grammatik $\Gamma = (N, T, \Pi, S)$ gilt $N \cap T = \{\}$ und $S \in N$.
☐	☒	Mit der Menge $\mathbb{T}_+ = \mathbb{N} \cup \{+\}$ als Alphabet kann die formale Sprache aller Terme der Form $a + b$, mit $a, b \in \mathbb{N}$ beschrieben werden. $\mathbb{T}_+$ *ist eine unendliche Menge und damit kein Alphabet.*
☐	☒	Es gibt keine formale Sprache, die den Text dieses Übungsblatts beschreibt. *Als Alphabet A wählen wir alle auf diesem Blatt vorkommenden Zeichen und als Teilmenge aus A^* wählen wir genau diejenige Zeichenreihe die unserem Übungsblatttext entspricht.*
☐	☒	Das Startsymbol in einer Sprachdefinition mit Syntaxdiagrammen ist ein Terminalsymbol.
☐	☒	Auf der linken Seite einer Produktion in einer kontextfreien Grammatik steht genau ein Terminalsymbol. *In* kontextfreien *Grammatiken darf dort nur ein* Nichtterminal *stehen. Andere Grammatiken erlauben auch mehrere Nichtterminale oder auch Terminalsymbole.*
☒	☐	Die Produktion $A \rightarrow abA$ kann nur an genau einer Stelle auf die Satzform abA angewendet werden.
☒	☐	Die dadurch mögliche Ableitung ist eine Linksableitung. *Da es nur eine Anwendungsstelle gibt, ist sie gleichzeitig natürlich auch eine Rechtsableitung.*
☐	☒	Eine Grammatik, die keine Produktion mit mehr als einem Nichtterminalsymbol auf der rechten Seite besitzt, ist eindeutig. *Gegenbeispiel:* $\Gamma = (N, T, \Pi, S)$, $T = \{a\}$, $N = \{S, A\}$, $\Pi = \{S \rightarrow A, A \rightarrow Aa, A \rightarrow aA, A \rightarrow a\}$ *In* Γ *hat der Satz* aa *die beiden Ableitungen* $S \Rightarrow A \Rightarrow Aa \Rightarrow aa$ *und* $S \Rightarrow A \Rightarrow aA \Rightarrow aa$*, damit ist* Γ *nicht eindeutig.*
☐	☒	Mit Syntaxdiagrammen lassen sich weniger Sprachen beschreiben als mit kontextfreien Grammatiken.