DEPENDABLE SYSTEMS AND SOFTWARE
Fachrichtung 6.2 — Informatik
Prof. Dr.-Ing. Holger Hermanns
Dipl.-Inform. Lijun Zhang

Übungsblatt 1 (Programmierung I)

Lesen Sie im Skript: Schnellkurs

Aufgabe 1.1: (Lexikalische Syntax)
Betrachten Sie das folgende Programm:

$$\begin{aligned} &val\ x = 7 + 4 \\ &val\ y = x * (x - 1) \\ &val\ z = \ x * (y - 2) \end{aligned}$$

Welche Bezeichner, Konstanten, Operatoren und Schlüsselwörter kommen in dem Programm vor? An welche Werte bindet das Programm die vorkommenden Bezeichner?

Aufgabe 1.2: (Signum)
Schreiben Sie eine Prozedur $signum : int \to int$, die für negative Argumente -1, für positive Argumente 1, und für 0 das Ergebnis 0 liefert.

Aufgabe 1.3: (17. Potenz)
Schreiben Sie eine Prozedur $hoch17 : int \to int$, die zu einer Zahl x die Potenz x^{17} berechnet. Dabei sollen möglichst wenig Multiplikationen verwendet werden. Schreiben Sie die Prozedur auf zwei Arten: Mit einer Hilfsprozedur und mit lokalen Deklarationen.

Aufgabe 1.4: (min)
Schreiben Sie eine Prozedur $min : int * int \to int$, die zu zwei Zahlen die kleinere liefert. Deklarieren Sie min analog zu swap auf 3 verschiedene Arten: mit einem kartesischen Argumentmuster, mit einer lokalen Deklaration, und mit Projektionen.

Aufgabe 1.5: (Min, Mid und Max)
Schreiben Sie Prozeduren min, mid und max des Typs $int * int * int \to int$, die zu drei Zahlen die kleinste, mittlere und größte liefern. Beispielsweise soll $mid(3, 56, 13)$ die Zahl 13 liefern.

Aufgabe 1.6: (Rekursionsfolgen)
Gegeben sei die rekursive Prozedurdeklaration

```
fun f(n: int, a: int): int = if n = 0 then a else f(n - 1, a * n)
```

(a) Geben Sie die Rekursionsfolge für den Aufruf $f(3, 1)$ an.

(b) Geben Sie ein detailliertes Ausführungsprotokoll für den Ausdruck $f(3, 1)$ an.
Halten Sie sich dabei an das Beispiel aus Abbildung 1.1 im Skript. Wenn es mehrere direkt ausführbare Teilausdrücke gibt, soll immer der am weitesten links stehende zuerst ausgeführt werden. Sie sollten insgesamt 18 Ausführungsschritte bekommen.

Aufgabe 1.7: (Prozeduren)
Erklären Sie den Unterschied zwischen einer Prozeduranwendung und einem Prozeduraufruf.

Aufgabe 1.8: (Multiplikation)
Schreiben Sie eine rekursive Prozedur $mul : int * int \to int$, die das Produkt zweier Zahlen mit wiederholter Addition berechnet. Nehmen Sie dabei an, dass der zweite Faktor nicht negativ ist.
Formulieren Sie den zugrunde liegenden Algorithmus zuerst mit Gleichungen.

Aufgabe 1.9: (Größter gemeinsamer Teiler)
Schreiben Sie eine rekursive Prozedur $ggt : int * int \to int$, die zu zwei positiven Zahlen den größten gemeinsamen Teiler berechnet. Beispielsweise soll $ggt(24, 40)$ das Ergebnis 8 liefern. Verwenden Sie den durch die folgenden Gleichungen gegebenen Algorithmus:

$$\begin{array}{ll} ggt(x,x) = x & \text{falls } x > 0 \\ ggt(x,y) = ggt(x-y,y) & \text{falls } x > y > 0 \\ ggt(x,y) = ggt(x,y-x) & \text{falls } y > x > 0 \end{array}$$

Aufgabe 1.10: (Teilbarkeit)
Schreiben Sie eine Prozedur $teilbar : int * int \to bool$, die für (x, y) testet, ob x durch y ohne Rest teilbar ist.

Aufgabe 1.11: (Zeitangaben)
Oft gibt man eine Zeitdauer *HMS-Format* mit Stunden, Minuten und Sekunden an. Beispielsweise ist 2h5m26s eine hervorragende Zeit für einen Marathonlauf.

(a) Schreiben Sie eine Prozedur $sec : int * int * int \to int$, die vom HMS-Format in Sekunden umrechnet. Beispielsweise soll $sec(1, 1, 6)$ die Zahl 3666 liefern.

(b) Schreiben Sie eine Prozedur $hms : int \to int * int * int$, die eine in Sekunden angegebene Zeit in das HMS-Format umrechnet. Beispielsweise soll $hms(3666)$ das Tupel $(1, 1, 6)$ liefern. Berechnen Sie die Komponenten des Tupels mithilfe lokaler Deklarationen.

Aufgabe 1.12: (Quersumme)
Schreiben Sie eine rekursive Prozedur $quer : int \to int$, die die Quersumme einer ganzen Zahl berechnet. Die Quersumme einer Zahl ist die Summe ihrer Dezimalziffern. Beispielsweise hat die Zahl -3754 die Quersumme 19. Verwenden Sie Restbestimmung modulo 10, um die letzte Ziffer einer positiven Zahl zu bestimmen, und ganzzahlige Division durch 10, um die Zahl zu erhalten, die durch Streichen der letzten Ziffer entsteht.

Aufgabe 1.13: (Reversion)
Unter der Reversion einer natürlichen Zahl x wollen wir die natürliche Zahl verstehen, die man durch Spiegeln der Dezimaldarstellung von x erhält. Beispielsweise soll $1234 \to 4321$, $76 \to 67$ und $1200 \to 21$ gelten.

(a) Schreiben Sie eine Prozedur $arity : int \to int$, die zu einer positiven Zahl x die größte Zehnerpotenz kleiner gleich x liefert. Beispielsweise soll $1234 \to 1000$, $76 \to 10$ und $5 \to 1$ gelten.

(b) Schreiben Sie eine Prozedur $reverse : int \to int$, die natürliche Zahlen reversiert. Verwenden Sie dabei die Prozedur *arity*.

Aufgabe 1.14: (Binomialkoeffizienten)
Schreiben Sie eine rekursive Prozedur $binom : int * int \to int$, die für $n,\ k \geq 0$ den Binomialkoeffizienten $\binom{n}{k}$ berechnet. Verwenden Sie den durch die folgenden Gleichungen gegebenen Algorithmus:

$$\begin{array}{ll} \binom{n}{0} = 1 & \\ \binom{0}{k} = 0 & \text{für } k > 0 \\ \binom{n}{k} = \frac{n * \binom{n-1}{k-1}}{k} & \text{für } n,\ k > 0 \end{array}$$

Aufgabe 1.15: (Fakultäten)
Für $n \geq 0$ können wir die so genannte n-te Fakultät $n!$ wie folgt definieren:

$$\begin{array}{ll} 0! = 1 & \\ n! = 1 \cdot \ldots \cdot n & \text{falls } n \geq 1 \end{array}$$

Beispielsweise gilt: $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$.

(a) Geben Sie zwei Gleichungen an, mit denen $n!$ berechnet werden kann.

(b) Realisieren Sie die Gleichungen mit einer Prozedur $fak : int \to int$. Schreiben Sie die Prozedur so, dass ihre Ausführung für negative Argumente wegen Speicherüberschreitung abgebrochen wird.

(c) Die Fakultäten werden schnell groß. Beispielsweise gilt $10! = 3628800$. Ermitteln Sie mit einem Interpreter das erste n, für das die Ausführung Ihre Prozedur zu einem Überlauf führt.