



Programmierung 1 (WS 2020/21)

Aufgaben für die Übungsgruppe A (Lösungsvorschläge)

Hinweis: Diese Aufgaben wurden von den Tutoren für die Übungsgruppe erstellt. Sie sind für die Klausur weder relevant noch irrelevant. 🤔 markiert potentiell schwerere Aufgaben.

Einführung

Aufgabe TA.1 (*Grundbegriffe*)

Identifizieren Sie Bezeichner, Schlüsselwörter, Operatoren und Konstanten in dem folgenden Programm:

```
1 fun p (x : int) : bool = x = 0
2 val x = if (p 3) then ~1 else 3 - 2
```

Lösungsvorschlag TA.1

Bezeichner: p, x
Schlüsselwörter: fun, =, val, if, (,), then, else, :, int, bool
Operatoren: =, -
Konstanten: 0, 3, 1, 2, ~1

Aufgabe TA.2 (*Der Schlüssel zu Schlüsselwörtern*)

- (a) Diskutieren Sie die Unterschiede zwischen Schlüsselwörtern, Bezeichnern, Operatoren und Konstanten.
- (b) Sammeln Sie so viele Schlüsselwörtern, Bezeichnern und Operatoren wie möglich.

Lösungsvorschlag TA.2

- (a)
 - **Bezeichner** dienen als Namen, die bei der Ausführung eines Programmes an Werte gebunden werden.
 - **Konstanten** sind Wörter, die bestimmte Werte bezeichnen. Beispielsweise bezeichnet die Konstante 7 die Zahl 7.
 - **Operatoren** sind Wörter, die Operationen darstellen. Beispielsweise stellt der Operator + die Additionsoption für Zahlen dar.
 - **Schlüsselwörter** dienen dazu, den Aufbau eines Programms darzustellen. Sie sind weder Konstanten, noch dürfen sie als Bezeichner verwendet werden.
- (b) Ein paar Beispiele sind unter anderem:
Bezeichner: x, y, z, max, a
Operatoren: = (als Vergleichsoperator), *, -, <, >=, ~
Schlüsselwörter: val, = (als Teil einer Deklaration), fun, int, if, (,)

Aufgabe TA.3 (*was ist was?*)

Betrachten Sie das folgende Programm:

```
1 val x = (1337 + 42)
2 val y = 3 = 5
3 val z = if y then x else x - 3
4 val foo = if x < ~7 then true else false
```

- (a) Welche Bezeichner, Konstanten, Operatoren und Schlüsselwörter kommen in dem Programm vor?
- (b) An welche Werte werden die vorkommenden Bezeichner gebunden?

Lösungsvorschlag TA.3

- (a) **Bezeichner:** x, y, z, foo
Konstanten: 1337, 42, 3, 5, ~7, true, false
Operatoren: = (Z. 2 bei '3 = 5'), +, -, <
Schlüsselwörter: val, =, if, then, else, (,)
- (b) $x \rightarrow 1379$, $y \rightarrow \text{false}$, $z \rightarrow 1376$, $\text{foo} \rightarrow \text{false}$

Aufgabe TA.4 (*noch mehr Bezeichner*)

Betrachten Sie das folgende Programm:

```
1 fun a (x : int) : int = if x < 0 then ~x else x
2 val b = 17
3 val c = ~21
4 fun foo (y : int, z : int) : int = y - 3 + a z
5 val c = foo (b, c)
```

- (a) Welche Bezeichner, Konstanten, Operatoren und Schlüsselwörter kommen in dem Programm vor?
- (b) An welchen Wert ist c am Ende gebunden?

Lösungsvorschlag TA.4

- (a) **Bezeichner:** a, b, c, foo, x, y, z
Konstanten: 0, 17, ~21, 3
Operatoren: <, ~, -, +
Schlüsselwörter: fun, =, if, then, else, val, (,), :, int
- (b) c hat am Ende den Wert 35

Funktionen und Prozeduren

Aufgabe TA.5 (Prozektion? Funktidur?)

- (a) Was verstehen wir unter Prozeduren und Funktionen? Was sind ihre Unterschiede, wie stehen sie in Beziehung zueinander?
- (b) Bei welchen der folgenden Deklarationen handelt es sich um Prozeduren, bei welchen um Funktionen? Diskutieren Sie mit ihrem Banknachbarn!
- (i) $\text{max} : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}, \text{max}(x, y) = \text{das gr\o{o}\ss te Element der Menge } \{x, y\}$
 - (ii) $\text{fun abs}(a : \text{int}) : \text{int} = \text{if } a \geq 0 \text{ then } a \text{ else } \sim a$
 - (iii) $\text{potenz} : \mathbb{Z} \times \mathbb{N} \mapsto \mathbb{N}, \text{potenz}(x, y) = x^y$
 - (iv) $\text{fun somefunc}(a : \text{int}, b : \text{int}) : \text{int} = \text{if } a > b \text{ then } a \text{ else } b$
 - (v) $\text{fun } g(a, b) = (a + b + \text{abs}(a - b)) \text{div } 2$
 - (vi) Es sei $f : \mathbb{R} \mapsto \mathbb{R}^+$, wobei $\frac{d}{dx}f(x) = f(x)$ und $f(0) = 1$.

Lösungsvorschlag TA.5

- (a) **Funktionen** sind mathematische Objekte. Sie gibt an, wie ihre Argumente dem Ausgabewert zugeordnet werden. Sie geben keine konkrete Berechnungsvorschrift vor.
- Prozeduren** wiederum sind direkte, eindeutige Berechnungsvorschriften. Sie **berechnen** eine bestimmte *Funktion* und sind in der Regel in einer Programmiersprache geschrieben.
- (b)
- Funktionen: i, iii, vi
 - Prozeduren: ii, iv, v

Aufgabe TA.6 (Funktion \rightarrow Prozedur)

Schreiben Sie Prozeduren für folgende Funktionen:

$$f : \mathbb{Z} \rightarrow \mathbb{N} \\ f(x) = x^2$$

$$g : \mathbb{Z} \rightarrow \mathbb{Z} \\ g(x) = x^5 + 3 * x^3 + x + 7$$

Lösungsvorschlag TA.6

```
1 fun prodF (x : int) : int = x * x
2
3 fun prodG (x : int) : int = x * x * x * x * x + 3 * x * x * x + x + 7
```

Aufgabe TA.7 (hoch hinaus)

Schreiben Sie eine Prozedur `hoch17 : int \rightarrow 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.

Lösungsvorschlag TA.7

mit Hilfsprozedur:

```
1 fun quadrat (x : int) : int = x * x
2 fun hoch17 (x : int) : int = quadrat(quadrat(quadrat(quadrat(x)))) * x
```

mit lokalen Deklarationen:

```
1 fun hoch17 (x : int) : int =
2     let
3         val a = x * x
4         val b = a * a
5         val c = b * b
6         val d = c * c
7     in
8         d * x
9     end
```

Aufgabe TA.8 (*mod*)

- (a) Deklarieren Sie eine Prozedur `modulo : int * int → int`, welche für natürliche Zahlen `mod` berechnet, allerdings ohne `mod` selbst zu benutzen.
- (b) Deklarieren Sie eine Prozedur `modComp : (int * int) * (int * int) → bool`, welche ausgibt, ob der Rest der Division von der ersten durch die zweite Zahl gleich dem Rest der Division der dritten durch die vierte Zahl ist.

Lösungsvorschlag TA.8

- (a) `fun modulo(a: int, b: int) : int = x - x div y * y`
- (b) `fun modComp((a:int, b:int), (c:int, d:int)) : bool = modulo(x,y) = modulo(a,b)`