

Einführung in die mathematische Software Übung extra



TECHNISCHE
UNIVERSITÄT
DARMSTADT

PD Dr. Ulf Lorenz
Dipl.-Math. Thomas Opfer

Wintersemester 2012/2013

Dieser Übungszettel soll grundlegende Dinge wiederholen und vertiefen und ist insbesondere an die Teilnehmer gerichtet, die wenig Programmiererfahrung haben. Viele der hier geforderten Funktionalitäten bietet Maple bereits, Sie sollten diese hier aber nicht verwenden, sondern stattdessen selbst implementieren.

Aufgabe 1 Prozeduren

```
p := proc(n)
  ...
end proc;
```

Abbildung 1: Das Code-Gerüst einer Prozedur p in Abhängigkeit von n .

- Schreiben Sie eine Prozedur, die überprüft, ob eine übergebene natürliche Zahl n echt größer als 5 ist, und falls dies der Fall ist, auf dem Bildschirm "Die Zahl ist größer 5." ausgibt. Ansonsten soll nichts passieren. Testen Sie ihre Prozedur mit den Zahlen -6, -5, -4, 0, 5, 6, 7.
- Ergänzen Sie die Prozedur so, dass "Die Zahl ist nicht größer 5." ausgegeben wird, falls die Zahl nicht größer als 5 ist.
- Verändern Sie die Prozedur so, dass sie im Falls $n > 5$ zusätzlich prüft, ob die Zahl gerade ist. Geben Sie eine entsprechende Meldung auf dem Bildschirm aus.
Hinweis: In der Hilfe finden Sie unter dem Stichwort `type/even` Hinweise, wie Sie prüfen können, ob eine Zahl gerade ist. Sie dürfen aber auch gerne die Modulo-Operation verwenden, falls Ihnen diese geläufig ist.
Hinweis: Es ist möglich, `if`-Abfragen zu verschachteln.
- Es ist auch möglich, mehrere Bedingungen mittels logischer Verknüpfungen wie `and` oder `or` zu verknüpfen. Schreiben Sie nun eine Kurzform der Prozedur, die lediglich überprüft, ob eine gegebene Zahl n größer als 5 und gerade ist, und dies ggf. ausgibt. Verwenden Sie keine Verschachtelung von `if`-Abfragen.

Einführung in die mathematische Software – Übung extra

Aufgabe 2 Schleifen

```
p := proc(n)
  local i;
  ...
  for i from 1 to n do
    ...
  end do;
  ...
end proc;
```

Abbildung 2: Beispiel einer in eine Prozedur eingebettete `for`-Schleife, die n mal durchlaufen wird. Dabei nimmt die Variable i sukzessive die Werte $1, 2, \dots, n$ an.

```
p := proc(n)
  local i;
  ...
  while i < 10 do
    ...
  end do;
  ...
end proc;
```

Abbildung 3: Beispiel einer in eine Prozedur eingebettete `while`-Schleife, die solange durchlaufen wird, bis die Bedingung $i < 10$ fehlschlägt.

- Schreiben Sie eine Prozedur, die mit Hilfe einer `for`-Schleife alle Zahlen von 1 bis n jeweils multipliziert mit 8314 ausgibt.
- Wie können Sie das selbe Problem mit Hilfe einer `while`-Schleife lösen.

Aufgabe 3

- a) Schreiben Sie eine Prozedur $p(n)$, die alle Zahlen von 1 bis n ausgibt. Außerdem soll mit Hilfe der Prozedur aus [Aufgabe 1 d\)](#) überprüft werden, ob die Zahlen jeweils größer als 5 und gerade sind.

Beispielausgabe:

```
1
2
3
4
5
6
"Die Zahl ist größer 5 und gerade."
7
8
"Die Zahl ist größer 5 und gerade."
:
```

- b) Prüfen Sie zusätzlich, ob die Zahlen prim sind und geben Sie dies ggf. aus. Denken Sie an den `isprime`-Befehl.

Aufgabe 4 Listen

```
seq1 := NULL;
seq1 := seq1, 3, 9;

seq2 := 1, 2, 7, 3;

list1 := [ seq2 ];

list2 := [ 8, 4, 2 ];

a := list2[2];
```

Abbildung 4: Anlegen der leeren Sequenz `seq1` und Anfügen mehrerer Elemente. Anlegen der Sequenz `seq2`. Umwandeln der Sequenz `seq2` in die Liste `list1`. Anlegen der Liste `list2`. Zugriff auf das 2. Element der Liste `list2`.

Hinweis: Die Anzahl der Elemente einer Liste bestimmen Sie mit dem Befehl `nops`.

- a) Schreiben Sie eine Prozedur `drin(e1, li)`, die überprüft, ob das Element `e1` in der Liste `li` enthalten ist. Legen Sie eine Liste mit den Elementen 0, 8, 15, π , 42, 23 an und prüfen Sie, welche der Elemente 0, 1, π , e und 23 sie enthält oder nicht.
- b) Modifizieren Sie die Prozedur so, dass sie ggf. die Stelle zurückgibt, an der sich das Element befindet.
- c) Schreiben Sie eine Prozedur, die alle Vorkommnisse der Zahl 5 in einer Liste durch die Zahl 8 ersetzt. Legen Sie dazu innerhalb der Prozedur eine lokale Kopie der Liste an und modifizieren Sie diese. Geben Sie sie am Ende zurück. Testen Sie ihre Prozedur mit der Liste `l := [5, 9, 2, 0, 0, 1, 5, 8, 0, 5]`.
- d) Schreiben Sie eine Prozedur, die alle Elemente einer übergebenen Liste aufsummiert.
- e) Die Liste `F := [21, 23, 19, 28, 32, 21, 22, 21, 25, 26, 58]` enthält die Altersangaben einer Fußball-Startelf. Bestimmen Sie das Durchschnittsalter.
- f) Erstellen Sie eine Liste mit den ersten 25 Zahlen, die durch 12 und 15 teilbar und gerade sind.
- g) *Für Fortgeschrittene:* Schreiben Sie eine Prozedur, die die Reihenfolge einer Liste umkehrt und testen Sie sie an der Liste `F`.
- h) *Für Fortgeschrittene:* Schreiben Sie eine Prozedur, die eine Liste aufsteigend sortiert und testen Sie sie an der Liste `F`.