

o++o und EXCEL im Vergleich

Klaus Benecke, Eicke Redweik

(Stand: 11.06.2023)

In der vorliegenden Arbeit werden o++o und EXCEL anhand eines typischen Problems mit einer strukturierten Tabelle bzgl. des Programmieraufwands verglichen. Des Weiteren wird betrachtet, welcher Aufwand für eine leichte Änderung des Schemas der strukturierten Tabelle anfällt. Dabei stellt sich heraus, dass im o++o-Programm nur die Ausgangstabelle leicht erweitert werden muss, die EXCEL-Anwendung dagegen erfordert starke Änderungen. Man kann sogar sagen, dass es sich hierbei um verschiedene EXCEL-Anwendungen handelt. Vom Standpunkt des Anwenders vereinfacht sich die Lösung dieses Problems bzw. dieser Problemklasse durch o++o wesentlich, da viele EXCEL-Anwendungen mit o++o in ähnlicher aber einfacherer Weise realisiert werden können. Wenn man den typischen EXCEL-Anwender mit einem o++o-Anwender vergleicht, wird man feststellen, dass sowohl die Formulierung als auch die nachträgliche Anpassung einer o++o-Lösung deutlich weniger Zeit als die EXCEL-Lösung in Anspruch nimmt. Die Ergebnisse dieser Arbeit gelten für nicht-strukturierte Tabellen ebenso wie strukturierte Tabellen, so diese keine leere Menge enthalten und somit leicht ohne Informationsverlust flach gemacht werden können. Abschließend werden beide Ansätze nach allgemeinen Gesichtspunkten miteinander verglichen.

Inhalt

| | |
|--|----|
| 1. Ausgangsproblem..... | 1 |
| 1.1. Lösung mit o++o..... | 2 |
| 1.2. Lösung mit EXCEL (manuell)..... | 5 |
| 1.3. Lösung mit EXCEL (automatisiert)..... | 6 |
| 2. Erweitertes Problem..... | 10 |
| 2.1. Lösung mit o++o..... | 11 |
| 2.2. Lösung mit Excel (manuell)..... | 12 |
| 2.3. Lösung mit Excel (automatisiert)..... | 14 |
| 3. Auswertung..... | 15 |
| 4. Allgemeine Gegenüberstellung..... | 17 |

1. Ausgangsproblem

Für die folgende Tabelle sollen die Werte des BMI (Body-Mass-Index) nach der Formel $BMI = \text{GEWICHT} : (\text{LAENGE} \cdot \text{LAENGE}) = \text{GEWICHT} : \text{LAENGE} : \text{LAENGE}$ bestimmt werden. Es soll sowohl der Wert für jede einbezogene Person und für jedes für diese Person aufgeführte Alter berechnet werden, als auch gruppiert nach Alter die BMI-Werte gemittelt über die Personen der Altersgruppe, und der Mittelwert aller BMI-Werte. Es sind nur Personen einzubeziehen, für die

ALTER-Werte über 20 vorliegen. Wir werden gleich sehen, dass die Formulierung der Anfrage in o++o leichter zu erfassen ist als die vorangehende Beschreibung in deutscher Sprache.

| NAME | LAENGE | ALTER | GEWICHT |
|----------|--------|-------|---------|
| Klaus | 1.68 | 18 | 61 |
| | | 30 | 65 |
| | | 56 | 80 |
| Rolf | 1.78 | 40 | 72 |
| Kathi | 1.70 | 18 | 55 |
| | | 40 | 70 |
| Walleri | 1.00 | 3 | 16 |
| Viktoria | 1.61 | 13 | 51 |
| Bert | 1.72 | 18 | 66 |
| | | 30 | 70 |

1.1. Lösung mit o++o

Die gegebene Tabelle kann als Datei gespeichert werden oder wie unten in TAB-Darstellung in den Programmteil (oberes hell grünes Feld) eingetragen werden. Das o++o-Programm besteht aus der gewünschten Selektion (avec-Zeile) und einer Umstrukturierung mit Aggregation (gib-Zeile). Durch

```
rnd 2
```

werden anschließend alle Zahlen auf 2 Stellen nach dem Komma gerundet. Das Ergebnis in TAB-Format findet man im unteren dunkelgrünen Feld, wobei sich der Tabellenkopf im Mittelfeld befindet.

The screenshot shows a web browser window with the URL localhost:8888/web/index.html. The main content area is divided into three horizontal sections:

- Top section (light green background):** Contains a table with columns NAME, LAENGE, (ALTER, GEWICHT m)m. The data rows are:

| NAME | LAENGE | (ALTER | GEWICHT m)m |
|-----------|--------|--------|-------------|
| Klaus | 1.68 | 18 | 61 |
| | | 30 | 65 |
| | | 61 | 80 |
| Rolf | 1.78 | 40 | 72 |
| Kathi | 1.70 | 18 | 55 |
| | | 40 | 70 |
| Walleri | 1.00 | 3 | 16 |
| Viktorija | 1.61 | 13 | 51 |
| Bert | 1.72 | 18 | 66 |
| | | 30 | 70 |
- Middle section (grey background):** Contains a control panel with various options:
 - WIDTH: 500
 - META: normal no both only
 - RUN:
 - autoclear
 - [help](#)
- Bottom section (dark green background):** Shows the result of a query:


```
BMI, (ALTER, BMI2, (NAME, BMI3 m) m)
23.12 18      20.98 Bert 22.31
           Kathi 19.03
           Klaus 21.61
           30      23.35 Bert 23.66
           Klaus 23.03
           40      23.47 Kathi 24.22
           Rolf 22.72
           61      28.34 Klaus 28.34
```

o++o: Eingabe-Daten und Verarbeitungscode (oben, grün unterlegt), Schema des Ausgabetafments (mitte, sehr dunkel unterlegt) und Inhalt des Ausgabetafments (unten, dunkelgrün unterlegt)

Speichert man, wie bereits erwähnt die Ausgangsdaten einfach als Textdatei im Format „tab“ unter dem Namen `personen.tab`,

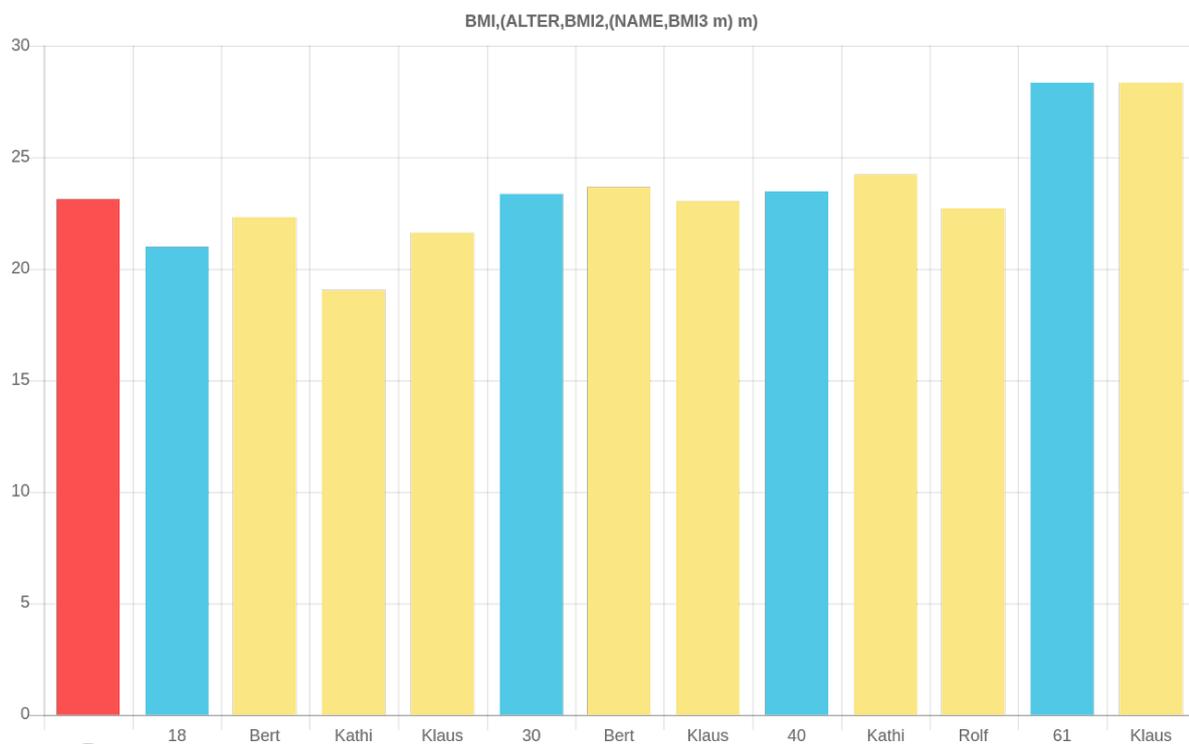
```
NAME,      LAENGE,    (ALTER,  GEWICHT m)m
Klaus      1.68      18      61
           30      65
           61      80
Rolf       1.78      40      72
Kathi      1.70      18      55
           40      70
Walleri    1.00      3       16
Viktorija 1.61      13      51
Bert       1.72      18      66
```

so nimmt obiges Programm die folgende Gestalt an.

```
personen.tab
avec NAME! ALTER>20
gib BMI, (ALTER, BMI, (NAME, BMI m) m) BMI :=GEWICHT:LAENGE:LAENGE!++:
rnd 2
```

Ausführliche Beschreibungen zu o++o findet man unter www.ottops.de.

Durch einen Klick auf den diagram-Button wird eine Auswahlliste für ein gewünschtes **strukturiertes** Diagramm vorgeschlagen. Nach einem weiteren Klick wird das gewählte Diagramm angezeigt. Hier sind die Werte der übergeordnete Säulen bläulich hervorgehoben. Der Balken der nullten Ebene ist rot.



o++o- Diagramm: Säulendiagramm für die strukturierte Ausgabetable

In o++o-Diagrammen werden numerische Werte als Säulenlängen und nichtnumerische Werte als Unterschriften interpretiert. Da ALTER im obigen Programm numerisch ist, musste für die Erzeugung des obigen Diagramms noch die Programmzeile

```
ALTER :=ALTER wort
```

angehängt werden.

1.2. Lösung mit EXCEL (manuell)

Bei Verwendung von EXCEL muss zunächst eine Filterung nach dem Kriterium „Personen mit Alter > 20“ **manuell** durchgeführt werden, da die naive Anwendung Filterbedingung ALTER>20 nach dem „Flachmachen“ der Tabelle relevante Datenzeilen eliminieren würde.

In der Spalte E2 dieser Tabelle wird die Formel für den BMI eingetragen und bis E9 kopiert.

| | A | B | C | D | E | F |
|----|-------|--------|-------|---------|------------|---|
| 1 | NAME | LAENGE | ALTER | GEWICHT | BMI3 | |
| 2 | Klaus | 1,68 | 18 | 61 | 21,6128118 | |
| 3 | | 1,68 | 30 | 65 | 23,0300454 | |
| 4 | | 1,68 | 56 | 80 | 28,3446712 | |
| 5 | Rolf | 1,78 | 40 | 72 | 22,7244035 | |
| 6 | Kathi | 1,7 | 18 | 55 | 19,0311419 | |
| 7 | | 1,7 | 40 | 70 | 24,2214533 | |
| 8 | Bert | 1,72 | 18 | 66 | 22,3093564 | |
| 9 | | 1,72 | 30 | 70 | 23,6614386 | |
| 10 | | | | | | |

BMI Berechnung in EXCEL: Relevante Datenzeilen wurden manuell gefiltert, die LAENGE dupliziert und der BMI je Datenzeile mittels Formel bestimmt und kopiert.

Anschließend erfolgen die Sortierung und Gruppierung nach Alter. Dabei werden auch die Namen in die leeren Zellen kopiert, um eine flache Tabellenstruktur zu erzeugen. Die Berechnung der BMI-Mittelwerte pro Altersgruppe erfolgt in Spalte F und die Berechnung des BMI-Mittelwertes für alle Daten in Spalte G.

| | A | B | C | D | E | F | G | H |
|----|-------|--------|-------|---------|------------|------------|------------|---|
| 1 | NAME | LAENGE | ALTER | GEWICHT | BMI3 | BMI2 | BMI | |
| 2 | Klaus | 1,68 | 18 | 61 | 21,6128118 | 20,9844367 | 23,1169153 | |
| 3 | Kathi | 1,7 | 18 | 55 | 19,0311419 | | | |
| 4 | Bert | 1,72 | 18 | 66 | 22,3093564 | | | |
| 5 | | | | | | | | |
| 6 | | | | | | | | |
| 7 | Klaus | 1,68 | 30 | 65 | 23,0300454 | 23,345742 | | |
| 8 | Bert | 1,72 | 30 | 70 | 23,6614386 | | | |
| 9 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |
| 13 | Rolf | 1,78 | 40 | 72 | 22,7244035 | 23,4729284 | | |
| 14 | Kathi | 1,7 | 40 | 70 | 24,2214533 | | | |
| 15 | | | | | | | | |
| 16 | | | | | | | | |
| 17 | | | | | | | | |
| 18 | | | | | | | | |
| 19 | Klaus | 1,68 | 56 | 80 | 28,3446712 | 28,3446712 | | |
| 20 | | | | | | | | |

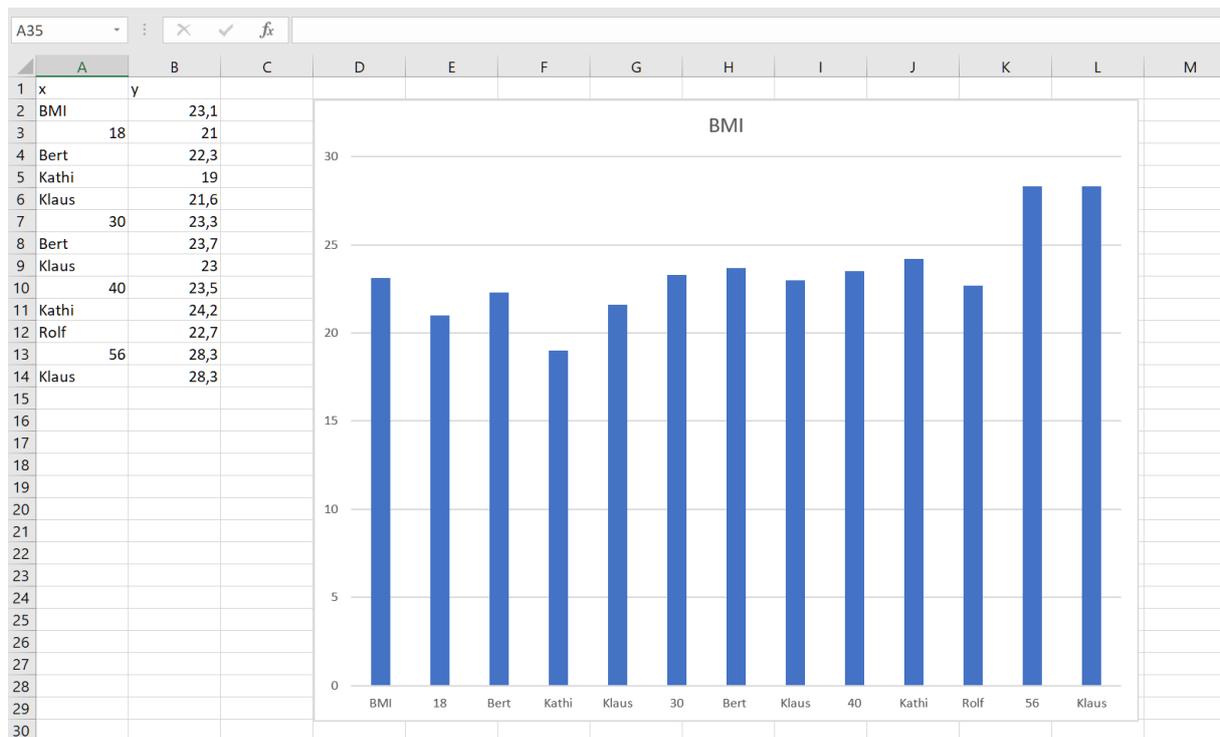
BMI Berechnung in EXCEL auf Gruppiereten Daten: Die Datenzeilen der flachen Tabellen wurden manuell nach ALTER gruppiert und Mittelwerte für den BMI bestimmt.

Werden Zeilen oder Spalten hinzugefügt oder gelöscht, müssen bei der manuellen Bearbeitung die Formelbereiche der berechnenden Zellen korrigiert werden. Gemäß der gewünschten Anordnung müssen Spalten verschoben und in der Ergebnistabelle nicht benötigte Spalten ausgeblendet werden.

| | A | B | C | D | E | H |
|----|------------|-------|------------|-------|------------|---|
| 1 | BMI | ALTER | BMI2 | NAME | BMI3 | |
| 2 | 23,1169153 | 18 | 20,9844367 | Klaus | 21,6128118 | |
| 3 | | 18 | | Kathi | 19,0311419 | |
| 4 | | 18 | | Bert | 22,3093564 | |
| 5 | | 30 | 23,345742 | Klaus | 23,0300454 | |
| 6 | | 30 | | Bert | 23,6614386 | |
| 7 | | 40 | 23,4729284 | Rolf | 22,7244035 | |
| 8 | | 40 | | Kathi | 24,2214533 | |
| 9 | | 56 | 28,3446712 | Klaus | 28,3446712 | |
| 10 | | | | | | |

Manuelle Spaltenselektion in EXCEL: Spalten manuell angeordnet und ausgeblendet.

Diese Ergebnistabelle entspricht bis auf die Rundung und der Redundanz in der ALTER-Spalte dem Berechnungsergebnis von $o++o$. Die grafische Ausgabe erfordert eine weitere manuelle Aufbereitung der darzustellenden Daten, die mit wachsendem Datenvolumen ebenfalls wächst.



Grafische Darstellung in EXCEL: Daten werden in einer flachen Tabelle angeordnet, um ein Säulendiagramm zu erzeugen

1.3. Lösung mit EXCEL (automatisiert)

Vorbemerkung: EXCEL tendiert dazu, leere Zellen als null-Werte oder 0-Werte zu interpretieren. Um das Erscheinen unerwünschter 0-Werte zu unterbinden wird häufig eine Abfrage der Form $IF(<Zelle>="" ; "" ; <Formel>)$ benutzt, welche nicht jedes Mal explizit wiederholt wird.

Des Weiteren können Schritte, welche auf einer variablen Tabellengröße arbeiten, zum Beispiel Zeilenselektion oder Spaltenselektion, in Zellen außerhalb der Ausgabetabelle Fehler werfen, wenn diesen null-Werte übergeben werden. In diesem Fall werden diese Zellen durch `IFERROR(<Formel>;"")` in leere Zellen überführt. Auch dies wird nicht an jeder Stelle explizit wiederholt.

Die Verarbeitungspipeline wird realisiert, indem jeder Schritt auf einem neuen Arbeitsblatt ausgegeben wird. Das Schema wird in vielen Verarbeitungsschritten nicht angetastet und stattdessen in das folgende Arbeitsblatt unverändert übernommen. Die Zellen des Schemas werden als Schemazellen bezeichnet, die weiteren relevanten Zellen als Datenzellen.

EXCEL-Formeln, die ein ganzes Arbeitsblatt verarbeiten können lange Zeit und viel Speicherplatz in Anspruch nehmen. Für dieses Beispiel werden Formeln benutzt, die auf jedem Arbeitsblatt maximal den Bereich `A1:J50` einnehmen (inklusive Schema).

Bei Verwendung von EXCEL können die Daten entweder manuell eingetragen oder als eine Datei im Format „csv“ hinterlegt werden. Durch freie Zellen kann so auch eine strukturierte Tabelle wie im Beispiel vorgegeben für EXCEL angewandt werden. In diesem Fall ist der erste Vorverarbeitungsschritt das „Flachmachen“ der Eingabetabelle, denn die weitere Verarbeitung in EXCEL erfordert eine nicht-strukturierte Tabelle.

Dieser Schritt wird durch die Formel `IF(input.A2="";A1;input.A2)` realisiert. Da die erste Zeile voll belegt ist, wird durch diese Anwendung jede leere Zelle mit dem Inhalt der darüber liegenden Zelle gefüllt.

| | A | B | C | D |
|----|----------|--------|-------|---------|
| 1 | NAME | LAENGE | ALTER | GEWICHT |
| 2 | Klaus | 1,68 | 18 | 61 |
| 3 | | | 30 | 65 |
| 4 | | | 56 | 80 |
| 5 | Rolf | 1,78 | 40 | 72 |
| 6 | Kathi | 1,7 | 18 | 55 |
| 7 | | | 40 | 70 |
| 8 | Walleri | 1 | 3 | 16 |
| 9 | Viktoria | 1,61 | 13 | 51 |
| 10 | Bert | 1,72 | 18 | 66 |
| 11 | | | 30 | 70 |

Arbeitsblatt Input: Strukturierte Tabelle in Tabellenkalkulation

| | A | B | C | D |
|----|----------|--------|-------|---------|
| 1 | NAME | LAENGE | ALTER | GEWICHT |
| 2 | Klaus | 1,68 | 18 | 61 |
| 3 | Klaus | 1,68 | 30 | 65 |
| 4 | Klaus | 1,68 | 56 | 80 |
| 5 | Rolf | 1,78 | 40 | 72 |
| 6 | Kathi | 1,7 | 18 | 55 |
| 7 | Kathi | 1,7 | 40 | 70 |
| 8 | Walleri | 1 | 3 | 16 |
| 9 | Viktoria | 1,61 | 13 | 51 |
| 10 | Bert | 1,72 | 18 | 66 |
| 11 | Bert | 1,72 | 30 | 70 |

Arbeitsblatt Input-Flat: Flache Tabelle in Tabellenkalkulation

Nun kann eine automatische Filterung nach dem Kriterium „Personen mit Alter > 20“ durchgeführt werden. Die Selektion kann am besten in drei Schritten erfolgen:

- Man bestimmt die zu selektierenden Zeilen mit der Formel `$'input-flat'.C2>=20`, wobei die Spalte C die ALTER-Werte enthält (Arbeitsblatt ‚avec-bool‘).
- Man übernimmt die zu selektierenden Zeilen in ein komplett neues Arbeitsblatt mit der Formel `IF(COUNTIF('avec-bool'.B2:B50;$'input-flat'.$A2);$'input-flat'.A2;"")` (Arbeitsblatt ‚avec-holes‘), wobei Leerzeilen verbleiben.
- Leerzeilen werden in zwei Schritten eliminiert:
 - Leerzeilen werden zunächst mit `NOT($'avec-holes'.A2="")` erkannt und in diesem Arbeitsblatt wird mit `COUNTIF(A1:A2;1)` ein Index für (nichtleere) Datenzeilen erzeugt (Arbeitsblatt ‚hole-help‘).

- Mit der Formel `LOOKUP('hole-help'.$C2;$'hole-help'.$B1:$'hole-help'.$B50;'avec-holes'.A2:A50)` werden nur die nichtleeren Datenzeilen eingelesen (Arbeitsblatt ,avec‘).

Man erhält eine Tabelle, welche äquivalent zu der mit `o++o` erzeugten Tabelle nach diesem Verarbeitungsschritt ist.

| | A | B | C | D |
|----|------------|---|---|---|
| 1 | | | | |
| 2 | FALSCH | | | |
| 3 | WAHR Klaus | | | |
| 4 | WAHR Klaus | | | |
| 5 | WAHR Rolf | | | |
| 6 | FALSCH | | | |
| 7 | WAHR Kathi | | | |
| 8 | FALSCH | | | |
| 9 | FALSCH | | | |
| 10 | FALSCH | | | |
| 11 | WAHR Bert | | | |

Arbeitsblatt Avec-Bool: Wahrheitswerte zur Selektion der relevanten Schlüssel

| | A | B | C | D |
|----|--------|---|---|---|
| 1 | | 0 | | |
| 2 | WAHR | 1 | 0 | |
| 3 | WAHR | 2 | 1 | |
| 4 | WAHR | 3 | 2 | |
| 5 | WAHR | 4 | 3 | |
| 6 | WAHR | 5 | 4 | |
| 7 | WAHR | 6 | 5 | |
| 8 | FALSCH | 6 | 6 | |
| 9 | FALSCH | 6 | 7 | |
| 10 | WAHR | 7 | 8 | |
| 11 | WAHR | 8 | 9 | |

Arbeitsblatt Hole-Help: Wahrheitswerte zur Selektion der Leerzeilen

| | A | B | C | D |
|----|-------|--------|-------|---------|
| 1 | NAME | LAENGE | ALTER | GEWICHT |
| 2 | Klaus | 1,68 | 18 | 61 |
| 3 | Klaus | 1,68 | 30 | 65 |
| 4 | Klaus | 1,68 | 56 | 80 |
| 5 | Rolf | 1,78 | 40 | 72 |
| 6 | Kathi | 1,7 | 18 | 55 |
| 7 | Kathi | 1,7 | 40 | 70 |
| 8 | | | | |
| 9 | | | | |
| 10 | Bert | 1,72 | 18 | 66 |
| 11 | Bert | 1,72 | 30 | 70 |

Arbeitsblatt Avec-Holes: Flache Tabelle in Tabellenkalkulation nach Selektion mit Leerzeilen

| | A | B | C | D |
|----|-------|--------|-------|---------|
| 1 | NAME | LAENGE | ALTER | GEWICHT |
| 2 | Klaus | 1,68 | 18 | 61 |
| 3 | Klaus | 1,68 | 30 | 65 |
| 4 | Klaus | 1,68 | 56 | 80 |
| 5 | Rolf | 1,78 | 40 | 72 |
| 6 | Kathi | 1,7 | 18 | 55 |
| 7 | Kathi | 1,7 | 40 | 70 |
| 8 | Bert | 1,72 | 18 | 66 |
| 9 | Bert | 1,72 | 30 | 70 |
| 10 | | | | |
| 11 | | | | |

Arbeitsblatt Avec: Flache Tabelle nach Selektion ohne Leerzeilen

Im Weiteren ist es nun notwendig, die Daten zunächst nach Alter zu sortieren. Der Einfachheit halber werden dafür zunächst die Spalten mit `LOOKUP` dem Schema `ALTER, NAME, GEWICHT, LAENGE` entsprechend sortiert (Arbeitsblatt ,gib-help‘).

Die Sortierung erfolgt nur für einzelne Spalten, also wird zunächst nach `ALTER` sortiert: `SMALL(OFFSET('gib-help'.A2:A50;0;0;49-COUNTIF('gib-help'.A2:A50;""));ROW(F2)-1)`. Zu beachten ist hier insbesondere die Bestimmung der Zeilenzahl in Abhängigkeit von den maximal unterstützten Datenzeilen ($49 = 50 - 1$, da eine Schemazeile reserviert ist).

Für diese sortierte Spalte werden nun zeilenweise die noch fehlenden Datenzellen mittels `INDEX('gib-help'.B:B;'gib-help-sort-help'.$F2+1;1)` ergänzt (Arbeitsblatt ,gib-help-sort‘).

| | A | B | C | D |
|----|-------|----------|---------|--------|
| 1 | ALTER | NAME | GEWICHT | LAENGE |
| 2 | | 18 Klaus | 61 | 1,68 |
| 3 | | 30 Klaus | 65 | 1,68 |
| 4 | | 56 Klaus | 80 | 1,68 |
| 5 | | 40 Rolf | 72 | 1,78 |
| 6 | | 18 Kathi | 55 | 1,7 |
| 7 | | 40 Kathi | 70 | 1,7 |
| 8 | | 18 Bert | 66 | 1,72 |
| 9 | | 30 Bert | 70 | 1,72 |
| 10 | | | | |
| 11 | | | | |

Arbeitsblatt Gib-Help: Tabellenspalten automatisch neu sortiert

| | A | B | C | D |
|----|-------|----------|---------|--------|
| 1 | ALTER | NAME | GEWICHT | LAENGE |
| 2 | | 18 Klaus | 61 | 1,68 |
| 3 | | 18 Kathi | 55 | 1,7 |
| 4 | | 18 Bert | 66 | 1,72 |
| 5 | | 30 Klaus | 65 | 1,68 |
| 6 | | 30 Bert | 70 | 1,72 |
| 7 | | 40 Rolf | 72 | 1,78 |
| 8 | | 40 Kathi | 70 | 1,7 |
| 9 | | 56 Klaus | 80 | 1,68 |
| 10 | | | | |
| 11 | | | | |

Arbeitsblatt Gib-Help-Sort: Sortiert nach erster Tabellenspalte (Alter)

Dies ist das letzte Arbeitsblatt, welches Informationen über GEWICHT und LAENGE enthält, also muss spätestens jetzt der BMI mittels $\$'gib-help-sort'.C2/\$'gib-help-sort'.D2/\$'gib-help-sort'.D2$ berechnet werden (Arbeitsblatt ‚BMI‘).

| | A | B | C | D | E |
|----|-------|----------|---------|--------|-------------|
| 1 | ALTER | NAME | GEWICHT | LAENGE | BMIh |
| 2 | | 18 Klaus | 61 | 1,68 | 21,61281179 |
| 3 | | 18 Kathi | 55 | 1,7 | 19,03114187 |
| 4 | | 18 Bert | 66 | 1,72 | 22,30935641 |
| 5 | | 30 Klaus | 65 | 1,68 | 23,03004535 |
| 6 | | 30 Bert | 70 | 1,72 | 23,66143862 |
| 7 | | 40 Rolf | 72 | 1,78 | 22,72440348 |
| 8 | | 40 Kathi | 70 | 1,7 | 24,22145329 |
| 9 | | 56 Klaus | 80 | 1,68 | 28,3446712 |
| 10 | | | | | |
| 11 | | | | | |

Arbeitsblatt BMI: Eine neue Spalte bildet die Grundlage der BMI-Spalten in der strukturierten Ausgabetablelle.

Die Verarbeitung von gib wird abgeschlossen indem zunächst die Spalten ALTER und NAME übernommen werden (Formel wie bei Spaltenumsortierung: $HLOOKUP(B\$1; 'gib-help-sort'.\$A\$1:\$J\$50; ROW(\$K2))$). Drei neue Spalten werden aus den gesicherten BMI-Werten berechnet:

- $BMI = AVERAGE(\$BMI.E2:E50)$
- $BMI2 = IF(B2=B1; ""; AVERAGEIF(BMI. \$A\$2:\$A\$50; B2; BMI. \$E\$2:\$E\$50))$
- $BMI3 = \$BMI.E2$

Anschließend erfolgen die Sortierung und Gruppierung nach Alter. Dafür müssen auch noch die Namen in die leeren Zellen kopiert werden. Die Berechnung der BMI-Mittelwerte pro Altersgruppe erfolgt in Spalte F und die Berechnung des BMI-Mittelwertes für alle Daten in Spalte G.

Werden Zeilen oder Spalten hinzugefügt oder gelöscht, müssen die Formelbereiche der zu berechnenden Zellen korrigiert werden. Gemäß der gewünschten Anordnung müssen Spalten verschoben und in der Ergebnistabelle nicht benötigte Spalten ausgeblendet werden (Arbeitsblatt ‚gib‘).

| | A | B | C | D | E |
|----|-------------|-------|-------------|-------|-------------|
| 1 | BMI | ALTER | BMI | NAME | BMI |
| 2 | 23,11691525 | 18 | 20,98443669 | Klaus | 21,61281179 |
| 3 | | | | Kathi | 19,03114187 |
| 4 | | | | Bert | 22,30935641 |
| 5 | | 30 | 23,34574198 | Klaus | 23,03004535 |
| 6 | | | | Bert | 23,66143862 |
| 7 | | 40 | 23,47292839 | Rolf | 22,72440348 |
| 8 | | | | Kathi | 24,22145329 |
| 9 | | 56 | 28,3446712 | Klaus | 28,3446712 |
| 10 | | | | | |
| 11 | | | | | |

Arbeitsblatt Gib: Die strukturierte Ausgabetabelle mit allen gruppierten Werten für den BMI

Diese Ergebnistabelle entspricht bis auf die Rundung dem Berechnungsergebnis von $o++o$. Die Rundung wird mittels `IFERROR(ROUND($gib.A1;2);$gib.A1)` durchgeführt (zu beachten: IFERROR wird hier benutzt, um Textzellen zu ignorieren).

| | A | B | C | D | E |
|----|-------|-------|-------|-------|-------|
| 1 | BMI | ALTER | BMI | NAME | BMI |
| 2 | 23,12 | 18 | 20,98 | Klaus | 21,61 |
| 3 | | | | Kathi | 19,03 |
| 4 | | | | Bert | 22,31 |
| 5 | | 30 | 23,35 | Klaus | 23,03 |
| 6 | | | | Bert | 23,66 |
| 7 | | 40 | 23,47 | Rolf | 22,72 |
| 8 | | | | Kathi | 24,22 |
| 9 | | 56 | 28,34 | Klaus | 28,34 |
| 10 | | | | | |
| 11 | | | | | |

Arbeitsblatt Output: Die finale Ausgabetabelle mit gerundeten Werten

2. Erweitertes Problem

Erweitert man das Beispiel um mehrere Werte für das Gewicht pro Person und Alter ergibt sich der folgende Berechnungsablauf. Dabei sollen alle Werte für GEWICHT in die Berechnung der BMI-Mittelwerte pro Alter und des Gesamt-Mittelwertes für den BMI einbezogen werden. Bildet man zuerst die Mittelwerte für GEWICHT pro Person und Alter, folgt daraus ein abweichendes Ergebnis.

| NAME | LAENGE | ALTER | GEWICHT |
|----------|--------|-------|----------|
| Klaus | 1.68 | 18 | 61 60 62 |
| | | 30 | 65 63 67 |
| | | 56 | 80 82 79 |
| Rolf | 1.78 | 40 | 72 70 74 |
| Kathi | 1.70 | 18 | 55 52 |
| | | 40 | 70 71 |
| Walleri | 1.00 | 3 | 16 15 |
| Viktoria | 1.61 | 13 | 51 50 |
| Bert | 1.72 | 18 | 66 65 |
| | | 30 | 70 71 |

2.1. Lösung mit o++o

Eingabe der Tabelle als Tabment und des o++o-Programms. Wir erkennen, dass bis auf das Eintippen der zusätzlichen Daten nur ein l für Liste nach GEWICHT und ein H bei TAB eingefügt werden muss. Das H steht für horizontal, da die Gewichtsdaten jetzt kompakt horizontal dargestellt wurden.

```
<TABH!
NAME,      LAENGE,    (ALTER,  GEWICHTl m)m
Klaus      1.68       18       61 60 62
           30       65 63 67
           56       80 82 79
Rolf       1.78       40       72 70 74
Kathi      1.70       18       55 52
           40       70 71
Walleri    1.00        3       16 15
Viktoria   1.61       13       51 50
Bert       1.72       18       66 65
           30       70 71

!TABH>
avec NAME! ALTER>20
gib BMI, (ALTER,BMI, (NAME,BMI m) m) BMI:=GEWICHT:LAENGE:LAENGE!++:
rnd 2
```

The screenshot shows the OttoPSClient web interface. The top part displays the input code in a light green background. The bottom part shows the output, which is a table of BMI values for different ages and heights.

| BMI | ALTER | BMI2 | (NAME, BMI3 m) m) |
|-------|-------|-------|--|
| 23.26 | 18 | 20.88 | Bert 22.14 Kathi 18.51 Klaus 21.61 |
| 30 | 23.35 | 23.83 | Bert 23.83 Klaus 23.03 |
| 40 | 23.39 | 24.39 | Kathi 24.39 Rolf 22.72 |
| 56 | 28.46 | 28.46 | Klaus 28.46 |

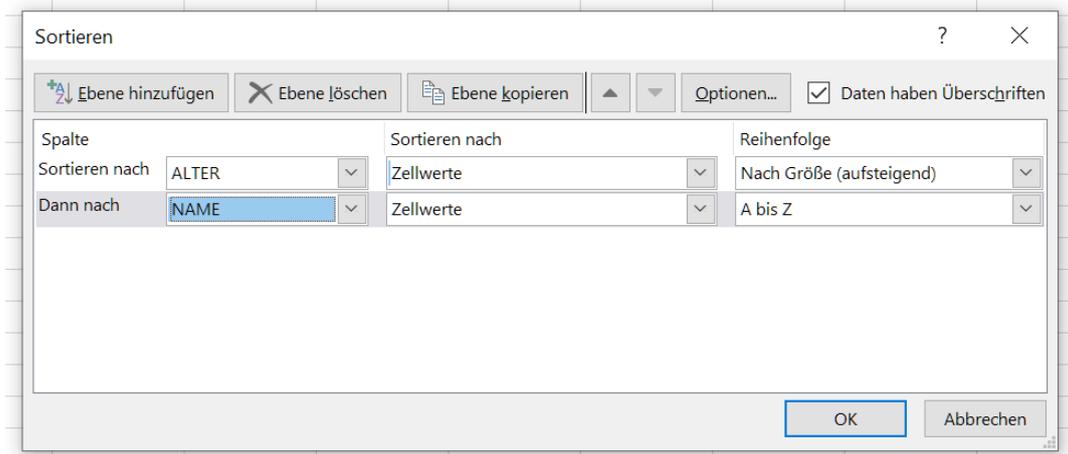
Erweitertes Problem im OttoPSClient: Ausschließlich die Eingabetabelle wird modifiziert und eine Ausgabetabelle mit dem entsprechenden Schema wird durch den gleichen Code erzeugt

Das gewünschte Diagramm kann man wieder durch zwei Klicks erzeugen, wenn man wieder zuvor den Typ von Alter in WORT transformiert. Da es dem obigen Diagramm ähnlich ist, verzichten wir hier auf eine zweite Angabe.

2.2. Lösung mit Excel (manuell)

Nach Anlegen der benötigten Tabellenspalten erfolgt die Eingabe der Werte für NAME, LAENGE, ALTER und GEWICHT. Dabei entsteht Redundanz, weil jedes einzubeziehende Gewicht in einer Zeile mit den zugehörigen Werten für NAME und ALTER stehen muss. Die Filterung nach dem Kriterium „Personen mit Alter > 20“ muss wiederum manuell erfolgen.

Für die Sortierung der Tabellenwerte nach ALTER und NAME wird die benutzerdefinierte Sortierung gewählt. Diese erfolgt halbautomatisch, muss aber nach jeder Modifikation der Eingabedaten erneut manuell ausgelöst werden.



Halbautomatische Sortierung in EXCEL: *Ein Dialog erlaubt die manuelle Auswahl der Spalten, nach welchen sortiert wird*

| | A | B | C | D | E |
|----|-------|--------|-------|---------|---|
| 1 | NAME | LAENGE | ALTER | GEWICHT | |
| 2 | Bert | 1,72 | 18 | 66 | |
| 3 | Bert | 1,72 | 18 | 65 | |
| 4 | Kathi | 1,7 | 18 | 55 | |
| 5 | Kathi | 1,7 | 18 | 52 | |
| 6 | Klaus | 1,68 | 18 | 61 | |
| 7 | Klaus | 1,68 | 18 | 60 | |
| 8 | Klaus | 1,68 | 18 | 62 | |
| 9 | Bert | 1,72 | 30 | 70 | |
| 10 | Bert | 1,72 | 30 | 71 | |
| 11 | Klaus | 1,68 | 30 | 65 | |
| 12 | Klaus | 1,68 | 30 | 63 | |
| 13 | Klaus | 1,68 | 30 | 67 | |
| 14 | Kathi | 1,7 | 40 | 70 | |
| 15 | Kathi | 1,7 | 40 | 71 | |
| 16 | Rolf | 1,78 | 40 | 72 | |
| 17 | Rolf | 1,78 | 40 | 70 | |
| 18 | Rolf | 1,78 | 40 | 74 | |
| 19 | Klaus | 1,68 | 61 | 80 | |
| 20 | Klaus | 1,68 | 61 | 82 | |
| 21 | Klaus | 1,68 | 61 | 79 | |
| 22 | | | | | |
| 23 | | | | | |

Flache Sortierte Tabelle in EXCEL: Die Zwischenergebnisse umfassen mehr Datenzeilen nachdem die Liste der Gewichte aufgelöst ist

In dieser Tabelle wird eine neue Spalte BMIx eingefügt, in deren Zellen die BMI-Werte nach der Formel $=D2/B2^2$ berechnet werden. Die Formel aus Feld E2 muss in alle Zellen der Spalte E kopiert werden. Dabei werden die Formeln entsprechend der gewählten Zeile automatisch angepasst.

Die Spalte BMI3 wird nach der Formel $=MITTELWERT(E2:E3)$ berechnet. Zu beachten ist die richtige Auswahl der Bereiche für die Berechnung des Mittelwertes.

Analog werden die Werte für BMI2 für die betreffenden Bereiche und BMI berechnet.

| | A | B | C | D | E | F | G | H |
|----|-------|--------|-------|---------|------------|------------|------------|------------|
| 1 | NAME | LAENGE | ALTER | GEWICHT | BMIx | BMI3 | BMI2 | BMI |
| 2 | Bert | 1,72 | 18 | 66 | 22,3093564 | 22,1403461 | 20,8776213 | 23,2622421 |
| 3 | Bert | 1,72 | 18 | 65 | 21,9713359 | | | |
| 4 | Kathi | 1,7 | 18 | 55 | 19,0311419 | 18,5121107 | | |
| 5 | Kathi | 1,7 | 18 | 52 | 17,9930796 | | | |
| 6 | Klaus | 1,68 | 18 | 61 | 21,6128118 | 21,6128118 | | |
| 7 | Klaus | 1,68 | 18 | 60 | 21,2585034 | | | |
| 8 | Klaus | 1,68 | 18 | 62 | 21,9671202 | | | |
| 9 | Bert | 1,72 | 30 | 70 | 23,6614386 | 23,8304489 | 23,3502068 | |
| 10 | Bert | 1,72 | 30 | 71 | 23,9994592 | | | |
| 11 | Klaus | 1,68 | 30 | 65 | 23,0300454 | 23,0300454 | | |
| 12 | Klaus | 1,68 | 30 | 63 | 22,3214286 | | | |
| 13 | Klaus | 1,68 | 30 | 67 | 23,7386621 | | | |
| 14 | Kathi | 1,7 | 40 | 70 | 24,2214533 | 24,3944637 | 23,3924276 | |
| 15 | Kathi | 1,7 | 40 | 71 | 24,567474 | | | |
| 16 | Rolf | 1,78 | 40 | 72 | 22,7244035 | 22,7244035 | | |
| 17 | Rolf | 1,78 | 40 | 70 | 22,0931701 | | | |
| 18 | Rolf | 1,78 | 40 | 74 | 23,3556369 | | | |
| 19 | Klaus | 1,68 | 61 | 80 | 28,3446712 | 28,462774 | 28,462774 | |
| 20 | Klaus | 1,68 | 61 | 82 | 29,053288 | | | |
| 21 | Klaus | 1,68 | 61 | 79 | 27,9903628 | | | |
| 22 | | | | | | | | |

Tabelle mit BMI-Werten in EXCEL: Die Spalten BMIx, BMI3, BMI2 und BMI werden in dieser Reihenfolge hinzugefügt und ermittelt

Jetzt können noch die Anzahl der Dezimalstellen mit „Zellen formatieren“ festgelegt werden und in der Ergebnisdarstellung nicht mehr benötigte Spalten ausgeblendet werden. Das Berechnungsergebnis erscheint in EXCEL dann wie folgt:

| | A | C | F | G | H |
|----|-------|-------|-------|-------|-------|
| 1 | NAME | ALTER | BMI3 | BMI2 | BMI |
| 2 | Bert | 18 | 22,14 | 20,88 | 23,26 |
| 4 | Kathi | | 18,51 | | |
| 6 | Klaus | | 21,61 | | |
| 9 | Bert | 30 | 23,83 | 23,35 | |
| 11 | Klaus | | 23,03 | | |
| 14 | Kathi | 40 | 24,39 | 23,39 | |
| 16 | Rolf | | 22,72 | | |
| 19 | Klaus | 61 | 28,46 | 28,46 | |
| 22 | | | | | |

Finale Ausgabe in EXCEL: Das Berechnungsergebnis ist eine formatierte Tabelle

2.3. Lösung mit Excel (automatisiert)

Die Lösung unterscheidet sich erst mit Übergang von ‚gib-help-sort‘ zu ‚gib‘. Ein weiteres Arbeitsblatt ‚gib-avg-help‘ wird benutzt, um Zeilen mit Zellen zu kennzeichnen, welche Informationen über ALTER und NAME enthalten z. B. CONCATENATE (TEXT (\$BMI . A2 ; "##, ##") ; \$BMI . B2) .

Es ergeben sich als neue Formeln für die BMI-Werte:

- `BMI = AVERAGE ($BMI.E2:E50)`
- `BMI2 = IF (B2=B1; ""; AVERAGEIF (BMI.A2:A50; B2; BMI.E2:E50))`
- `BMI3 = IFERROR (IF (AND (B2=B1; D2=D1); ""); AVERAGEIF ('gib-avg-help'. B2:B50; CONCATENATE (TEXT (B2; "##, ##"); D2); BMI.E2:E50)); "")`

Das Arbeitsblatt enthält noch redundante Zeilen ohne BMI-Werte (Arbeitsblatt ,gib-holes').

Diese können analog zu den Leerzeilen in ,avec-holes' markiert und selektiert werden (Arbeitsblatt ,gib'). Danach wird wie gehabt gerundet (Arbeitsblatt ,rnd').

| | A | B | C | D | E |
|----|-------|-------|-------|-------|-------|
| 1 | BMI | ALTER | BMI2 | NAME | BMI3 |
| 2 | 23,26 | 18 | 20,88 | Klaus | 21,61 |
| 3 | | | | Kathi | 18,51 |
| 4 | | | | Bert | 22,14 |
| 5 | | 30 | 23,35 | Klaus | 23,03 |
| 6 | | | | Bert | 23,83 |
| 7 | | 40 | 23,39 | Rolf | 22,72 |
| 8 | | | | Kathi | 24,39 |
| 9 | | 56 | 28,46 | Klaus | 28,46 |
| 10 | | | | | |
| 11 | | | | | |

Arbeitsblatt BMI2 Output: Die finale Ausgabetable mit gerundeten Werten

3. Auswertung

Im Vergleich zur Arbeitsweise mit Tabellenkalkulationen, die eine Folge von sorgsam formatierten Arbeitsblättern und Zwischenergebnissen und eine hohe Anzahl von Benutzereingaben erfordert, wie im obigen Beispiel gezeigt, kann o++o diese Aufgabe mit einem aus 4 Zeilen bestehenden Programm lösen, welches modifizierbar ist.

Tabellenkalkulationen sind beliebt, weil Inhalte interaktiv modifiziert werden können und das Ergebnis sofort sichtbar ist. Selbst einfache Anwendungen wie in dem beschriebenen Beispiel erfordern einen hohen Arbeitsaufwand und genaue Kenntnis der Funktionsweise der Software. Mit wesentlich geringerem Zeit- und Arbeitsaufwand erzeugt o++o ein interaktiv modifizierbares Programm und liefert sofort sichtbare Ergebnisse. Darüber hinaus sind die möglichen Modifikationen wesentlich weitreichender.

o++o ermöglicht eine hohe Flexibilität der Berechnungen mit geringem Anpassungsaufwand. Beispielsweise muss in der Zeile

```
gib BMI, (ALTER, BMI, (NAME, BMI m) m) BMI := GEWICHT : LAENGE : LAENGE ! ++ :
```

nur ALTER und NAME vertauscht werden, um eine ganz andere Anfrage zu realisieren.

Die **manuelle Lösung mit EXCEL** für das erweiterte Beispiel zur Berechnung der BMI-Werte erfordert folgende Schritte:

1. Anlegen des Arbeitsblatts mit Tabellenspalten und Eintragen der Werte für NAME, LAENGE, ALTER und GEWICHT
2. Manuelles Filtern, da die Menüfunktion Filtern nicht ausdrucksstark genug ist
3. Auffüllen leerer Zellen, damit die Menüfunktion Sortieren angewandt werden kann mit anschließendem Sortieren
4. Eingabe der Formel zur BMI-Berechnung in die oberste Zelle der Spalte BMIx
5. Kopieren dieser Formel in alle Zellen dieser Spalte (im gegebenen Beispiel sind das 20 Zellen)
6. Eingabe der Formel zur Berechnung des Mittelwertes BMI3 und Kopieren dieser Formel in 7 weitere Zellen
7. Anpassung der Bereiche in diesen Zellen
8. Eingabe der Formeln für BMI2 und BMI sowie Kopieren der Formel für BMI2 in 3 weitere Zellen und zusätzlich Anpassen der Bereiche
9. Anwendung der Menüfunktion „Zellen formatieren“
10. Ausblenden von 3 zur Ergebnisdarstellung nicht benötigten Spalten

Das erstmalige Anlegen dieses Arbeitsblattes erfordert also die Anwendung von 3 Menüfunktionen und die Eingabe von 4 Formeln in zu berechnende Zellen. Diese Formeln müssen in 29 weitere Zellen kopiert werden. Darüber hinaus müssen Gültigkeitsbereiche der Formeln in 12 Zellen angepasst werden.

Wird dieses Beispiel um weitere Werte ergänzt, müssen im vorliegenden Arbeitsblatt die Gültigkeitsbereiche der Formeln angepasst werden.

Die **automatisierte Lösung mit EXCEL** erfordert analoge Schritte über mehrere Arbeitsblätter, bedarf allerdings keiner nachträglichen Anpassung von Formeln bei Veränderung der Eingabedaten.

Die **Lösung mit o++o** erfordert dagegen nur:

1. Eingabe der Tabelle
2. Eingabe eines aus 2 bzw. 3 Zeilen bestehenden o++o-Programms
3. Das Ergebnis kann in verschiedenen Darstellungsformen, einschließlich grafischer Ausgabe, mit jeweils einem (bzw. zwei) Klick(s) aufgerufen werden.

Unser Aufwand für die Lösung dieser Aufgabe mit o++o betrug weniger als 20 % der mit EXCEL benötigten Zeit. Vorausgesetzt wurden für diese Annahme Kenntnisse der Systeme EXCEL und o++o. Für eine genauere und allgemeinere Schätzung des unterschiedlichen Arbeitsaufwandes müssen jedoch umfangreichere Studien durchgeführt werden.

4. Allgemeine Gegenüberstellung

Der Vergleich am gewählten Beispiel lässt viele weitere Aspekte unberührt. Wir stellen unabhängig vom obigen Beispiel folgende relevante Unterschiede zwischen EXCEL und o++o fest:

1. EXCEL arbeitet mit 2-dimensionalen Tabellen, wobei beide Dimensionen gleichberechtigt sind. o++o Tabellen besitzen stets einen Tabellenkopf (für die horizontale Dimension) mit dessen Spalten typischerweise gerechnet wird. Die horizontale Dimension wird durch Tupel (Verallgemeinerung eines Paares) beschrieben, wogegen die vertikale Dimension durch Kollektionen (Liste, Menge, Multimenge ...) spezifiziert wird.
2. Die Dateneingabe in EXCEL kann schneller erfolgen als die Datenerfassung in tab-Dateien von o++o, da letztere die Eingabe entsprechender Leerzeichen erfordert. Das wird mit hsq-Dateien vermieden. Diese sind für den Anfänger jedoch etwas schwerer lesbar.
3. Man muss in EXCEL jede Zahl in eine separate Zelle schreiben. Damit ist der Bildschirm (eines Smartphones) schnell überfüllt. In o++o besteht ein analoges Problem bei der Eingabe im tab-Format. Einsteigerfreundliche Darstellungen tendieren dazu, das Tabellenformat direkt darzustellen und nehmen entsprechenden Platz in Anspruch.
4. Da EXCEL keine entsprechenden Schemen kennt, sind XML-Daten nicht unmittelbar verarbeitbar.
5. Man kann mit den EXCEL-Konzepten keine Anfragen an Datenbanken, XML, ... stellen. Man muss dafür völlig andere Sprachen (SQL, XQuery, ...) erlernen.
6. EXCEL-Arbeitsblätter zu verstehen ist sehr schwierig, da einer o++o-Formel in der Regel mehrere EXCEL-Formeln entsprechen und diese EXCEL-Formeln über viele Zellen verteilt sind und mit Referenzen verbunden sein können. Durch die geringe Überschaubarkeit sind EXCEL Arbeitsblätter fehleranfällig und existierende Fehler schwer zu korrigieren.
7. o++o basiert auf mathematischen Konzepten und mengenweisen Operationen, wie Menge, Liste, Tupel, Strichlistenoperation,.... Die Handhabung von EXCEL kennt wenige Konzepte, verlangt dafür aber viel Detailwissen (wann und wie werden kopierte Formeln angepasst; findet eine Anpassung auch beim Einfügen oder Löschen von Zeilen oder Spalten statt usw.).
8. Da ein o++o-Programm wenige Formeln enthält, ist es leichter lesbar und modifizierbar.
9. Daten und Programm sind in o++o in der Regel getrennt. Daher können Daten problemlos von mehreren Programmen genutzt werden.
10. o++o-Bedingungen selektieren beispielsweise nur Daten und keine Formeln. Für eine einfache Bedingung „avec LAENGE>1.60“ angewandt auf unsere Ausgangstabelle, gibt es beispielsweise keine äquivalente Filterbedingung in EXCEL, da ganze Personensätze (jeweils mehrere Zeilen) einschließlich der untergeordneten (ALTER,GEWICHT)-Paare gelöscht werden. Eine Bedingung „avec NAME! GEWICHT>70“ kann auch nach dem „Flachmachen“ der Tabelle nicht abgebildet werden, da sie einen Existenzquantor „enthält“ (Gesucht sind alle Personen, die einen Gewichtseintrag größer 70 enthalten.). Obwohl o++o-Bedingungen ausdrucksstärker als EXCEL-Filterbedingungen sind, sind sie leichter zu überschauen, da keine Kollisionen mit Zellformeln auftreten können.
11. Bei Aggregationen (Summen, Durchschnitte, Maximum,...) pro Wert muss in EXCEL vorsortiert bzw. gruppiert werden, in o++o nicht.

12. o++o kann die inhaltlichen Aspekte strukturierter Tabellen unmittelbar verarbeiten. Diese sind dem gewünschten Druckbild in der Regel näher als flache Tabellen. Um spezifischere Druckbilder zu erzeugen, kann man die Daten über XML ausgeben und eine Stylesheetsprache benutzen. In EXCEL sind inhaltliche Fragen und Formprobleme dagegen nicht getrennt.
13. o++o basiert auf einem abstrakten Tabmentbegriff für Daten. Ein Tabment (=TAbelle+dokuMENT) lässt sich vielfältig (tab, xml, graphik, diagram, ...) und kompakt (hsq) konkret repräsentieren. EXCEL basiert auf **einer** zweidimensionalen konkreten Sicht eines Arbeitsblattes. Diese Sicht umfasst Daten und Formeln.
14. EXCEL-Formeln sind schwerer zu verstehen als o++o-Operationen. Eine einzelne Formel kann mehr Analyse erfordern als ein volles o++o-Programm.