

Programmierung für OttoNormalVerbraucher

Klaus Benecke

Facebook und Co. arbeiten daran Nachrichten so aufzubereiten, dass sie emotional noch mehr ansprechen, als ob die gesellschaftliche Situation nicht schon aufgeheizt genug ist. Wir arbeiten daran dem Endnutzer Werkzeuge bereitzustellen um seine rationale Urteilskraft mit Hilfe des Computers zu stärken. Dafür benötigt man möglichst einfache aber dennoch leistungsstarke Programmiersprachen und umfangreiche, vertrauenswürdige, öffentlich zugängliche Informationen in Form von vielgestaltigen großen Tabellen und Dokumenten ähnlich der Wikipedia.

Auch wenn die entwickelte Sprache so einfach wie möglich ist, wird sie im Gegensatz zum Facebookansatz einen gewissen Lernaufwand erfordern.

Eine solche Programmiersprache in Kombination mit vertrauensvollen Daten könnte ein großer Schritt in Richtung einer weiteren Demokratisierung der Gesellschaft werden. Viele Falschnachrichten könnten leicht von jedermann durch entsprechende Fakten oder statistischen Auswertungen paralyisiert werden.

Vielleicht kann man die Schaffung einer solchen Programmiersprache mit der Schaffung des ersten Alphabets durch die Phönizier oder der Schaffung des ersten Alphabets mit Vokalen durch die Griechen vergleichen. Hätten diese Völker solche Leistungen vollbringen können ohne diese Voraussetzungen. Ich vermute ohne dieses Alphabet hätte es keine griechische Wissenschaft und Kultur gegeben; vielleicht auch keine griechische Demokratie.

Entwurfskriterien für eine solche Sprache:

1. Eine mathematische Fundierung ist erforderlich.
2. Methodisch-didaktische und pragmatische Fragen stehen zunächst vor Effizienzproblemen.
3. Kurze, lesbare Programme; die wichtigsten Schlüsselworte sollten kurz sein
4. Einfache, unstrukturierte Programme; Schleifen und allgemeine Rekursionen führen häufig zu schwer lesbaren und schwer änderbaren Programmen;
5. Universelle Anwendbarkeit; sie muss nicht nur für Relationen (flache einfache Tabellen) sondern auch für strukturierte Tabellen und Dokumente nutzbar sein; sie muss nicht nur für Anfragen an die wichtigsten Systeme sondern auch für vielfältige Berechnungen geeignet sein
6. Um im Schulunterricht einsetzbar zu sein, muss sie die verschiedenen mathematische Teilgebiete unterstützen, sowie Nutzen für die anderen Fächer bieten
7. Sie sollte so mächtig sein, dass sie andere Systeme und Sprachen wie Tabellenkalkulation und SQL ersetzen kann.
8. Aus Endnutzersicht darf es nur ein einheitliches System mit einheitlicher Syntax (Schreibweise) für die Verarbeitung von Massendaten geben, genau wie die Operationen der Einzeldatenverarbeitung (+ - * :) standardisiert sind.

Einführung in o++o:

A. Merkel „Jeder Schüler soll neben lesen, rechnen und schreiben auch programmieren können.“

o++o (ausführlich ottoPS) ist eine tabellenorientierte Programmiersprache mit funktionalen Möglichkeiten, die auf Schleifen verzichtet. Dennoch ist o++o sehr ausdrucksstark und man kann mit ihr nicht nur kompakte Anfragen sondern auch vielfältige Berechnungen für strukturierte Tabellen und strukturierte Dokumente bewerkstelligen.

o++o benutzt viele mathematische Konzepte, daher sehen wir die Hauptvorteile der Vermittlung im Mathematikunterricht, genau wie die wesentlichen Fähigkeiten für die Nutzung des Taschenrechners in Mathematik vermittelt werden. o++o verwendet insbesondere folgende Konzepte: Kollektion (Menge, Multimenge, Liste); Gleichheit und Inklusionsbeziehungen dieser; Tupel; leistungsfähige Operationen zum Selektieren; Berechnen; Restrukturieren; Sortieren und Aggregieren (Summe; Durchschnitt; ...),... .

Tabellenkalkulationsprogramme wie EXCEL und die Datenbankstandardabfragesprache SQL kennen keine strukturierten Schemen und Tabellen. Erste Tests mit Vorschulkindern lassen vermuten, dass man mit strukturierten Tabellen leichter rechnen kann als mit Dezimalzahlen. Wir wollen einige o++o-Beispielprogramme anfügen:

1. Berechne den Wert eines einfachen Terms.

`2*3+4`

* und + haben jeweils 2 Inputwerte. Zunächst wird `2*3` (6) berechnet. Die 6 ist erster Inputwert von +, so dass sich insgesamt 24 ergibt. Hier wird also einfach von links nach rechts gerechnet.

2. Schreibe den Term $\cos^3(\sin^2(3.14159))$ in o++o.

`pi sin hoch 2 cos hoch 3`

Unserer Meinung nach ist der Ausgangsterm für Otto Normalverbraucher schwer zu lesen. Man beginnt mit pi geht nach links bis zum sin dann nach rechts zum hoch 2 jetzt bewegt man sich wieder nach links zum cos und abschließend nach rechts zum hoch 3. Diese Schreibweise wurde sicher eingeführt um Klammern zu sparen. Eigentlich müsste der Ausgangsterm unmissverständlich zu sein, folgendes Aussehen haben:

`(cos((sin(3.14159))2))3`

Das ist sicher noch schwerer zu lesen und man bewegt sich noch mehr von links nach rechts und umgekehrt.

3. Schreibe den Term $\sin^2(x)+\cos^3(y)$ in o++o.

`X sin hoch 2 + (Y cos hoch 3)`

oder

`X sin hoch 2
+ Y cos hoch 3`

Man könnte alle Terme in o++o ohne Klammern schreiben, allerdings müssten dann bestimmte Terme mehrzeilig geschrieben werden.

4. Wie berechnet man den Term $2+3:4*5$?

`2+(3:(4*5))=2 3/20`

`2+((3:4)*5)=5 3/4`

`o++o: ((2+3):4)*5=6 1/4`

Man erkennt, dass man mit der Schulweisheit Punktrechnung geht vor Strichrechnung noch nicht auskommt. Man benötigt die Regel „von links nach rechts“ zusätzlich.

5. Berechne den Durchschnitt mehrerer Noten.

`1 2 3 1 2 ++:`

Vom methodischen Standpunkt kann man dieses Programm noch verbessern, indem man die Klammern für Listen hinzufügt: `[1 2 3 1 2] ++:`

Man erkennt jetzt, dass die Durchschnittsoperation ++: einen Inputwert, nämlich eine Liste besitzt und dass ++: diesem einen Inputwert nachgestellt wird. Da die Nutzer in der Regel nicht viel tippen wollen, gehen wir davon aus, dass die erste Notation in Praxis häufiger benutzt werden wird.

6. Berechne die Durchschnitte einer strukturierten Tabelle `noten.tab` für jedes Fach.

`noten.tab`

`DUR:=NOTE1 ++:`

`noten.tab` könnte so aussehen:

`FACH,NOTE1 1`

`Ma 1 2 1 3 1 2`

```
Phy 4 3 2 2 1
```

...

Hierbei kürzt l Liste ab. D.h., noten.tab ist eine einfache strukturierte Tabelle (Liste), die zu jedem Fach eine Liste von Noten enthält. Um Platz zu sparen, wählen wir auch hier die methodisch nicht optimale Darstellung. Wie FACH ist auch NOTE ein Spaltenname, so dass noten.tab eigentlich so dargestellt werden müsste:

```
FACH,NOTE1 1
```

```
Ma 1
```

```
2
```

```
1
```

```
3
```

```
1
```

```
2
```

```
Phy 4
```

```
3
```

```
2
```

```
2
```

```
1
```

Das Ergebnis der Anfrage wieder im „tab-Format“:

```
FACH, DUR, NOTE1 1
```

```
Ma 1.6666666666667 1 2 1 3 1 2
```

```
Phy 2.4 4 3 2 2 1
```

7. Bilde die Summe der Zahlen von 1 bis 100 (Aufgabe von Gauß Klasse 5).

```
1 .. 100 ++
```

Wie die Addition und die Multiplikation besitzt .. zwei Inputwerte (1 und 100). Als

Zwischenergebnis entsteht die Liste

```
ZAHL1
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

```
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
```

```
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
```

```
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
```

```
84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
```

,deren Zahlen dann aufsummiert werden, so dass sich 5050 ergibt.

8. Berechne näherungsweise das Maximum der Sinus-Funktion im Intervall [1 2].

```
1 ... 2!0.001 sin max
```

... benötigt 3 Inputwerte: 1. den Anfangswert 1, den Endwert 2 und die Schrittweite 0.001. Es entstehen hierbei die Zahlen 1 1.001 1.002 1.003 ...1.999 2.

Auf jede der Zahlen wird die Sinusfunktion angewandt, sodass wieder 1001 Zahlen entstehen. Auf diese Liste wird dann die Funktion max (Maximum) angewandt. Obwohl es sich hierbei um ein Näherungsverfahren handelt, kommt der exakte Wert 1 heraus, wenn die Schrittweite weiter verfeinert wird. sin und max haben jeweils einen Inputwert (hier eine Liste) aber der Outputwert von sin ist wieder eine Liste und max erzeugt lediglich eine Zahl, da es sich hier um eine Aggregationsfunktion handelt. Der zweite und der dritte Inputwert einer dreistelligen Operation (oben ...) wird jeweils durch ein „!“ getrennt. Das ist in o++o nötig, da das Komma für die Paarbildung bereits vergeben ist und das Leerzeichen bereits Listenelemente trennt.

9. Berechne näherungsweise das Minimum des Polynoms $X^3 + 4 X^2 - 3 X + 2$ im Intervall [0 2] mit zugehörigem X-Wert.

```
[X! 0 ... 2!0.001]
```

```
Y:= X polynom [1 4 -3 2]
MINI:= Y1 min
avec Y = MINI
```

avec ist französisch und bezeichnet eine Selektion. Ein konkretes Polynom von einer Variablen X hat stets nur einen Inputwert, der für X eingesetzt wird. `polynom` in Zeile 2 ist dagegen allgemeiner und hat 2 Inputwerte:

1. Den Inputwert für X, der hier alle Zahlen, die in der ersten Zeile generiert wurden, annimmt.
2. Eine Liste von Zahlen, die den Koeffizienten des konkreten Polynoms entspricht.

Durch die ersten Zeile entsteht eine Liste von Zahlen, die alle den Namen X bekommen haben. Das erkennt man am besten in der xml bzw. ment-Repräsentation:

```
<X>0.</X>
<X>0.001</X>
<X>0.002</X>
```

...

Gesamtergebnis:

```
MINI, (X, Y 1)
1.481482037 0.333 1.481482037
```

10. Berechne eine Nullstelle der Cosinus Funktion im Intervall [1 2] näherungsweise.

```
[X! 1 ... 2!0.0001]
avec X cos < 0
avec X pos = 1
```

Hier verbleiben nach der ersten Selektion nur die X-Werte mit Funktionswert kleiner 0. Von diesen wird im zweiten Schritt der erste Wert ausgewählt. Da wir wissen, dass `cos` nur eine Nullstelle im betrachteten Intervall besitzt, wird diese durch das Ergebnis angenähert. `pos` kürzt Position ab, so dass das erste Paar der verbliebenen Paare selektiert wird.

11. Berechne das Gesamtwachstum, wenn 5 Jahreswachstumszahlen gegeben sind. Runde das Ergebnis auf eine Stelle nach dem Komma.

```
[W! 0 1.5 2.1 1.3 0.4 1.2]
ACCU:= first 100. next ACCU pred *(W:100+1) at W
rnd 1
```

Die Ergebnistabelle:

```
W, ACCU 1
0. 100.
1.5 101.5
2.1 103.6
1.3 105.
0.4 105.4
1.2 106.7
```

Der erste ACCU-Wert ergibt sich durch den Ausdruck hinter `first (100.)`. Für den zweiten Wert wird für `ACCU pred` der Wert `100.` eingesetzt und der Term nach `next` bewertet. Es ergibt sich 101.5. Diese Zahl wird wieder in `ACCU pred` eingesetzt und der `next`-Term erneut berechnet (rund 103.6),... bis der letzte W-Wert erreicht ist. `pred` ist der predecessor (Vorgänger).

12. Berechne die Fläche unter der Sinuskurve im Intervall [0, pi] näherungsweise.

```
0 ... pi!0.0001 sin*0.0001 ++
```

Hierbei werden nacheinander alle Zahlen zwischen 0 und pi generiert, dann von jeder Zahl der Sinus berechnet und anschließend jede Zahl mit 0.0001 multipliziert. Es entstehen 31415 Rechteckflächen, die abschließend addiert werden.

13. Berechne den DurchschnittsbMI pro Alter und den BMI pro Person und Alter für alle Personen über 20.

```
<TAB!  
NAME,          LAENGE,  (ALTER, GEWICHT 1) 1  
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  
  
!TAB>  
avec NAME! 20<ALTER  
BMI:= GEWICHT : LAENGE : LAENGE  
gib ALTER,BMIAVG,(NAME,BMI m) m BMIAVG:= BMI ! ++:  
rnd 2 #rundet alle Zahlen der Tabelle auf 2 Stellen nach dem Punkt
```

Die TAB-Klammern deuten an, dass die eingeschlossenen Daten der TAB-Darstellung entsprechen. Die obige Bedingung selektiert Personen-Sätze, d.h. NAME, LAENGE, (ALTER, GEWICHT 1) Tupel (strukturierte Tupel bzw. Strupel). Da eine Personen mehrere ALTER-Angaben besitzt, muss quantifiziert werden. NAME! 20 <ALTER selektiert demnach alle Personen, die einen entsprechenden Alterseintrag besitzen. D.h., der Existenzquantor wird nicht geschrieben, gehört aber zu jeder Bedingung. In diesem kleinen Beispiel könnte man die Selektion natürlich auch per Hand realisieren.

Resultat:

```
ALTER, BMIAVG, (NAME, BMI m) m  
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  
56      28.34   Klaus 28.34
```

Das Endergebnis kann beispielsweise durch einfaches Klicken als Säulendiagramm dargestellt werden. Das Beispiel zeigt, dass man eine Hierarchie einfach durch Angabe des gewünschten Schemas umkehren kann. Im Ergebnis ist der Name dem Alter untergeordnet.

Es wird insbesondere deutlich, dass die Aufgaben ohne Kenntnisse der Differential- und Integralrechnung gelöst werden können. Mit o++o kann der Mathematikunterricht in vielfältiger Weise unterstützt werden. Das reicht von Klasse 7 oder tiefer bis zur Klassenstufe 12. Es betrifft: Rechnen mit natürlichen Zahlen, Dezimalzahlen, näherungsweise Berechnung von Nullstellen beliebiger Funktionen, Ableitung, Flächen unter Kurven, Extremwerte (kann wahrscheinlich bereits in der Sekundarschule gelehrt werden), Wahrscheinlichkeitsrechnung, Mit o++o können Dinge in einfacher Weise berechnet werden, die sonst nur theoretisch abgehandelt werden. Dadurch kann das Verständnis der Konzepte wesentlich verbessert, erweitert und vertieft werden. Weitere Informationen zu o++o finden Sie unter ottops.de (Z.B. „o++o auf 8 Seiten“ ist eine kurze Einführung).

Wir glauben, dass o++o besondere Vorteile für den Mathematik- und Informatikunterricht bietet aber auch in den anderen Fächern sinnvoll genutzt werden kann.