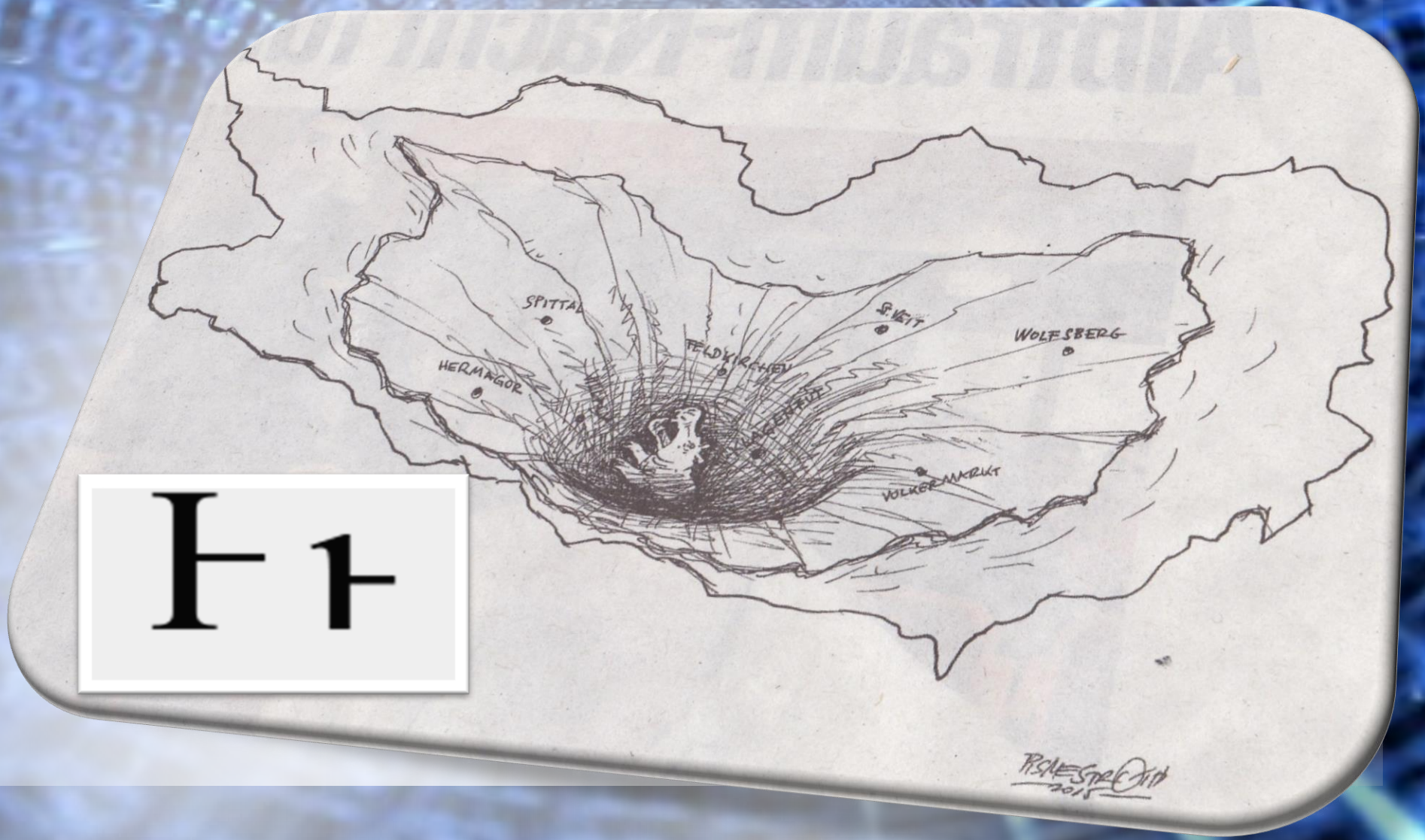
„Work“Shop

Vom programmierten Schüler zum programmierenden Schüler …

01000111 01111001 01101101 01101110
01000010 11100110 11010010 11101010
11011010 01000000 10101101 10000111
01101100 11011000 11010110 11001010
11100100 11011010 11000010 11100100
11010110 1110100

Griechenland darf nicht Kärnten werden



Programmierter Schüler?

```
5 3,8 8 6 : 8,4 =
5 3 8,8 6 : 8 4 = 6,4 1 5
  3 4 8
    1 2 6
      4 2 0
        0 0 R
```

7,23 : 5 = 1,446
-5
22
-20
23
-20
30
-30
0

5 4

Beispiel 1: 840 : 4 = 210
-8
04
-4
00
-0
0

Beispiel 5: 455 : 12 = 37 Rest 11
-36
95
-84
11

$f(x):g(x)=(2x^4 - x^3 + x^2 - 2x + 1) : (x^2 + 2x - 1) = 2x^2 - 5x + 13 = q(x)$

$2x^4 : x^2 = 2x^2$

$-(2x^4 + 4x^3 - 2x^2) \leftarrow 2x^2(x^2 + 2x - 1)$

$-5x^3 + 3x^2 - 2x + 1$

$-5x^3 : x^2 = -5x$

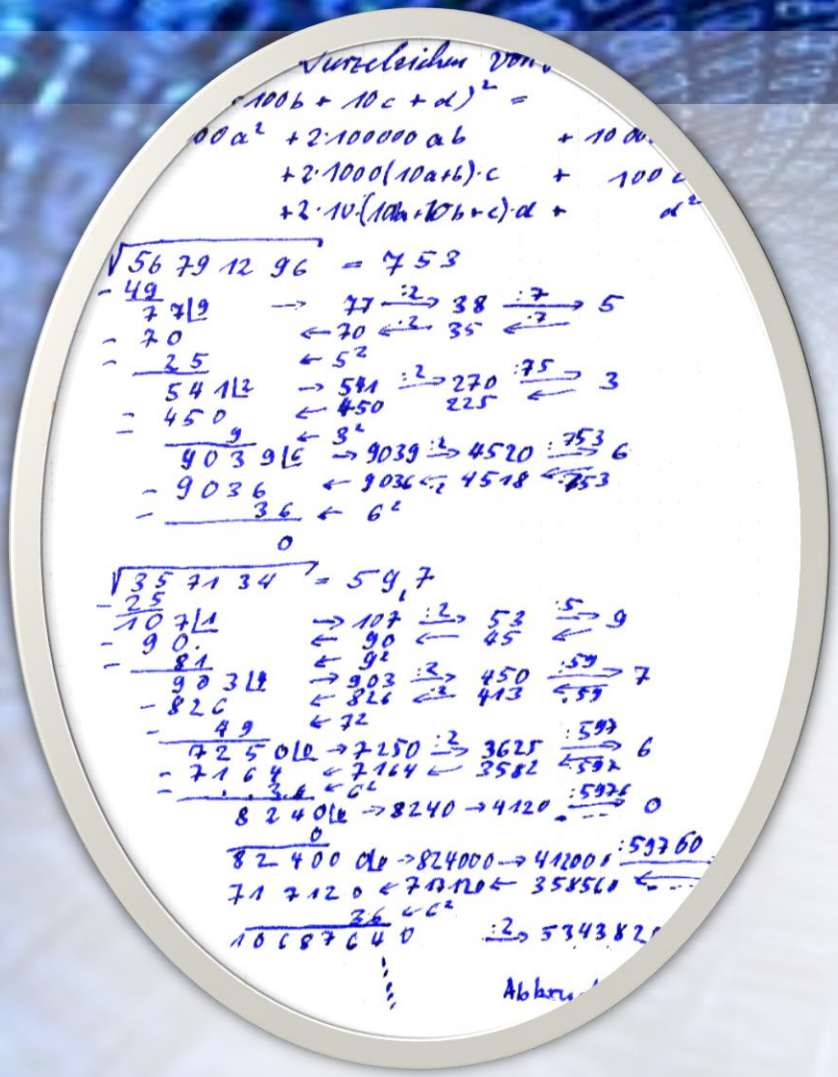
$-(-5x^3 - 10x^2 + 5x) \leftarrow -5x(x^2 + 2x - 1)$

$13x^2 - 7x + 1$

$13x^2 : x^2 = 13$

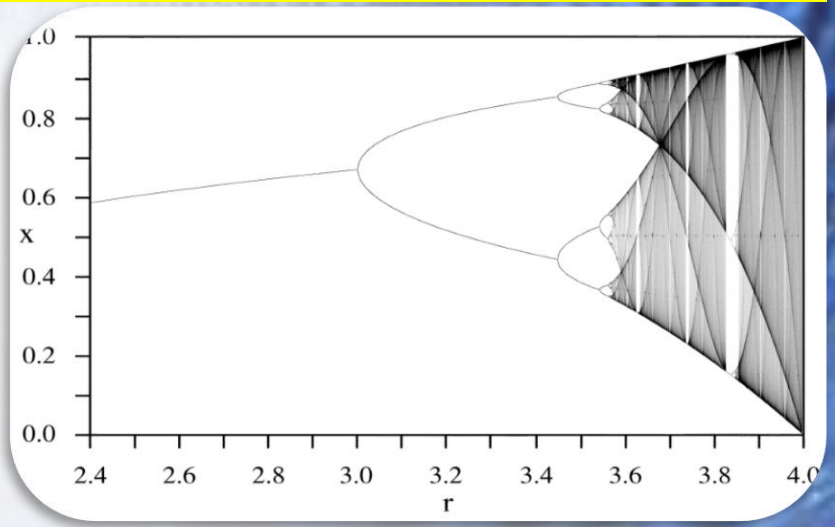
$-(13x^2 + 26x - 13) \leftarrow 13(x^2 + 2x - 1)$

Rest: $r(x) = -33x + 14$



$$x \leftarrow \tfrac{1}{2} * (x + a/x)$$

$$x \leftarrow r*x*(1-x)$$



1. Tippen Sie die folgende Tabelle ab. Die mit Formel markierten Zellen bleiben leer.

ARTIKEL	ANZAHL	PREIS	GESAMT	UST	GES. inkl. UST	Umsatz in Prozenten
Ruderboot A	5	5.000,00	Formel	20%	Formel	Formel
Ruderboot B	10	8.500,00	Formel	20%	Formel	Formel
Ruderboot C	14	9.800,00	Formel	20%	Formel	Formel
Segelboot A	18	50.000,00	Formel	20%	Formel	Formel
Segelboot B	10	125.000,00	Formel	20%	Formel	Formel
Segelboot C	1	320.000,00	Formel	20%	Formel	Formel
Motorboot A	8	60.000,00	Formel	20%	Formel	Formel
Motorboot B	9	155.000,00	Formel	20%	Formel	Formel
Motorboot C	2	230.000,00	Formel	20%	Formel	Formel
GESAMTEINNAHMEN			Formel		Formel	

2. Geben Sie der gesamten Tabelle die Schriftart Arial, 10 Pt.

3. Speichern Sie Tabelle unter BOOT.XLS ab.

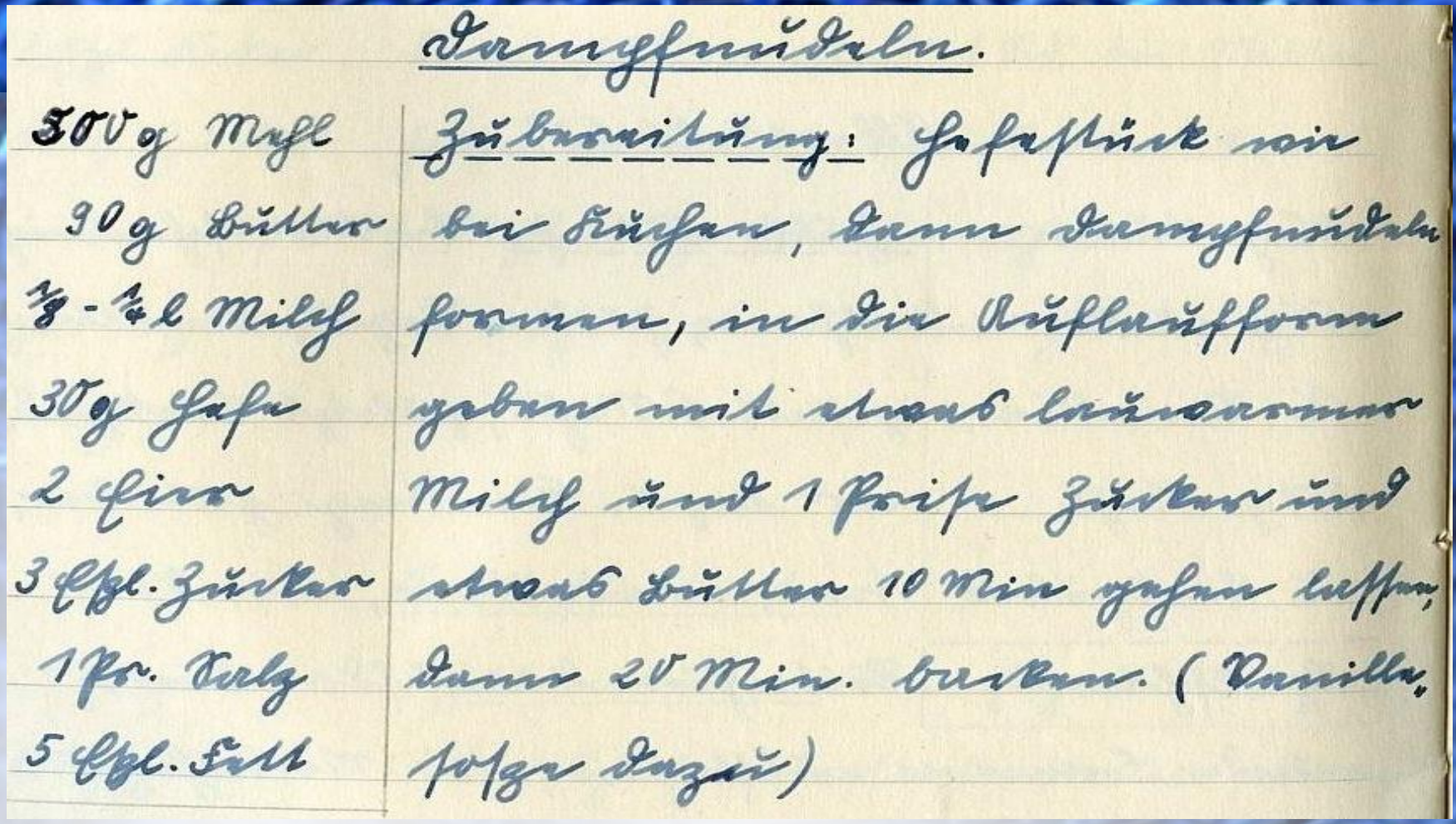
4. Berechnen Sie nun in der Spalte GESAMT den Wert aus ANZAHL mal PREIS und kopieren Sie anschließend die Formel nach unten.

5. Die Formel für die Spalte GES. inkl. UST ergibt sich aus GESAMT plus GESAMT mal UST. Kopieren Sie diese Formel nach unten.

6. Berechnen Sie nun mit der Summenfunktion die jeweiligen GESAMTEINNAHMEN.

Netzfundstück, Quelle leider überall …

http://archimeda1.ineineandrewelt.org/2015/01/rueckrufaktion-von-ikea-_lisa-ist-noch-da/



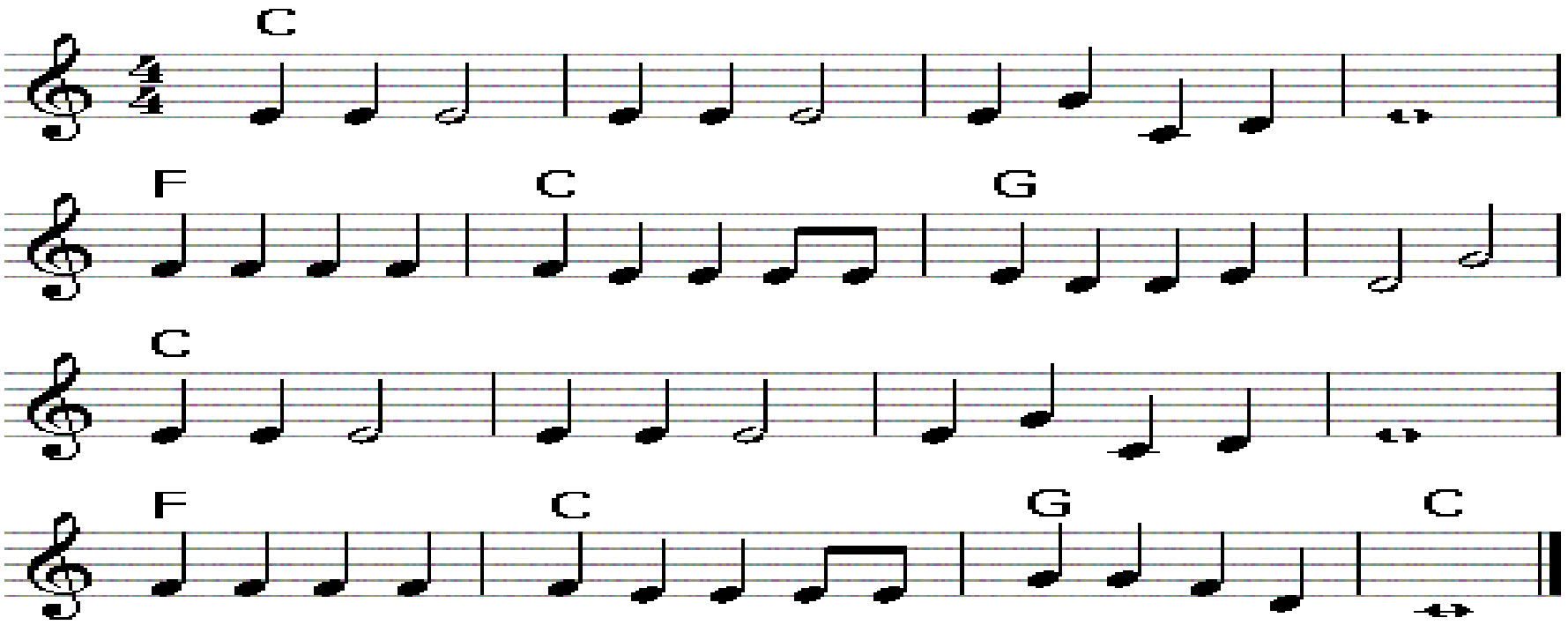
Dampfnudeln.

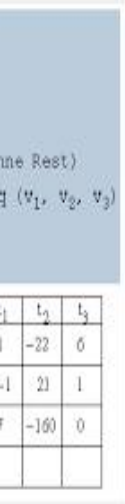
300 g Mehl
90 g Butter
1/8 - 1/4 l Milch
30 g Hefe
2 Eier
3 Efl. Zucker
1 Pr. Salz
5 Efl. Fett

Zubereitung: Hefestück wie bei Kuchen, dann Dampfnudeln formen, in die Auflaufform geben mit etwas lauwarmer Milch und 1 Prise Zucker und etwas Butter 10 Min gehen lassen, dann 20 Min. backen. (Vanillesoße dazu)

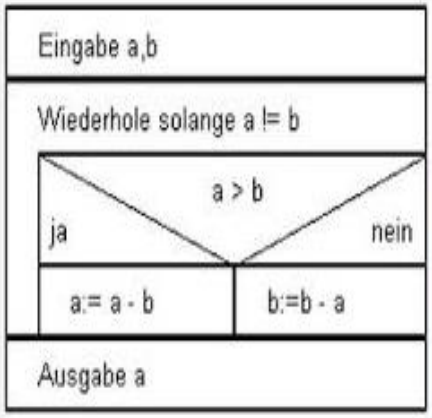
300g Mehl, 9og Butter, 1/8 - 1/4 Liter Milch, 30g Hefe, 2 Eier, 3 Esslöffel Zucker, 1 Priese Salz, 5 Esslöffel Fett.

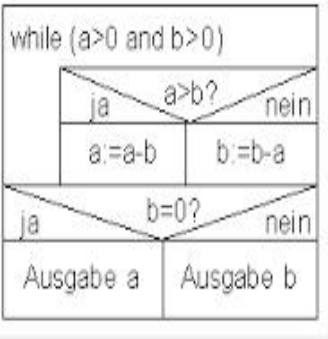
Zubereitung: Hefestück wie bei Kuchen [behandeln], dann Dampfnudeln [wie Klöße] formen, in die Auflaufform geben, mit etwas lauwarmer Milch und 1 Priese Zucker und etwas Butter 10 Minuten gehen lassen, dann 20 Minuten backen [Vanillesoße dazu]


C
F
C
G
C
F
C
G
C



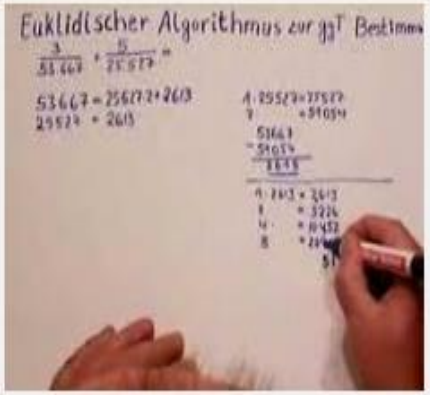
	-22	6
-1	23	1
7	-160	0

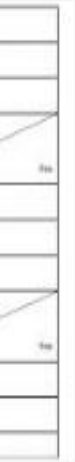


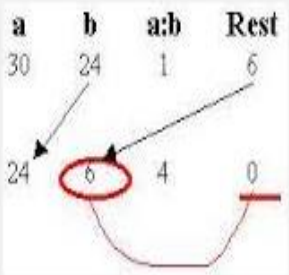


$$
\begin{aligned}
99 &= 1\cdot 78 + 21 &\iff 21 &= 1\cdot 99 - 1\cdot 78\\
78 &= 3\cdot 21 + 15 &\iff 15 &= 1\cdot 78 - 3\cdot 21 = -3\cdot 99 + 4\cdot 78\\
21 &= 1\cdot 15 + 6 &\iff 6 &= 1\cdot 21 - 1\cdot 15 = 4\cdot 99 - 5\cdot 78\\
15 &= 2\cdot 6 + 3 &\iff 3 &= 1\cdot 15 - 2\cdot 6 = -11\cdot 99 + 14\cdot 78
\end{aligned}
$$

$$
\begin{aligned}
2 &= 8 - 6\\
&= 8 - (22 - 2\cdot 8)\\
&= 3\cdot 8 - 22\\
&= 3\cdot (30 - 22) - 22\\
&= 3\cdot 30 - 4\cdot 22\\
&= 3\cdot 30 - 4\cdot (52 - 30)\\
&= 7\cdot 30 - 4\cdot 52.
\end{aligned}
$$

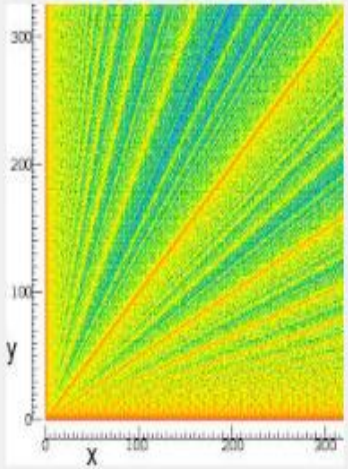




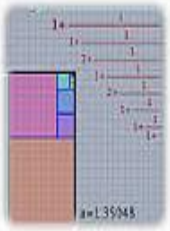


$$
\begin{aligned}
24.531 \div 10.839 &= 2 \text{ Rest } 2.853\\
10.839 \div 2.853 &= 3 \text{ Rest } 2.280\\
2.853 \div 2.280 &= 1 \text{ Rest } 573\\
2.280 \div 573 &= 3 \text{ Rest } 561\\
573 \div 561 &= 1 \text{ Rest } 12\\
561 \div 12 &= 46 \text{ Rest } 9\\
12 \div 9 &= 1 \text{ Rest } 3\\
9 \div 3 &= 3 \text{ Rest } 0
\end{aligned}
$$

$$
\begin{aligned}
99 &= 1\cdot 78 + 21\\
78 &= 3\cdot 21 + 15\\
21 &= 1\cdot 15 + 6\\
15 &= 2\cdot 6 + 3\\
6 &= 2\cdot 3 + 0
\end{aligned}
$$



$$
\begin{aligned}
99 &= 1\cdot 78 + 21\\
78 &= 3\cdot 21 + 15\\
21 &= 1\cdot 15 + 6\\
15 &= 2\cdot 6 + 3\\
6 &= 2\cdot 3 + 0
\end{aligned}
$$



SUPER PI DAY 3/14/15

1415926535897932384626433832795028841971693993751058209749445923078164062862089986280348253421170679
8214808651328230664709384460955058223172535940812848111745028410270193852110555964462294895493038196
44288109756659 0249141273
72458700660631 9415116094
33057270365759 8301194912
98336733624406 7669405132
00056812714526 6892589235
4201995611212902196086403441815981362977477130996051870721134999999837297804995105973173281609631859
5024459455346908302642522308253344685035261931188171010003137838752886587533208381420617177669147303
598253490428755468731159562863882 66130019278766111959092164201989
3809525720106548586327886593 41389124972177528347913151
557485724245415069595082 701671139009848824012
85836160356370766010 080466842590694912
933136770289891521 9927260426992279
67823547816360093 10797509302955
3211653449872027 9777727938000
816470600161452 2390739414333
454776241686251 1786085784383
8279679766814541 0222106611863
06744278622039194 39047802759009
9465764078951269468 437455305068203
49625245174939965143 01617539284681382
686838689427741559918559 262609912460805124388
439045124413654976278079771 2225828488648158456028506
016842739452267467678895252138522 5774564980355936345681743241125
1507606947945109659609402522887971089314566913686722874894056010150330861792868092087476091782493858
9009714909675985261365549781893129784821682998948722658804857564014270477555132379641451523746234364
5428584447952658678210511413547357395231134271661021359695362314429524849371871101457654035902799344
0374200731057853906219838744780847848968332144571386875194350643021845319104848100537061468067491927
8191197939952061419663428754440643745123718192179998391015919561814675142691239748940907186494231961
5679452080951465502252316038819301420937621378559566389377870830390697920773467221825625996615014215
0306803844773454920260541466592520149744285073251 8

5) Euler showed that

a) $\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \ldots$

b) $\frac{\pi^4}{90} = 1 + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \ldots$

What approximations for π are given by the first 1000 terms of these series?

3.9 Monte-Carlo method for determination of π.

The Monte-Carlo method involves the estimation of a number from the result of a probabalistic experiment. We give two examples.

(a) π from randomly-falling rain

Suppose 1000 raindrops fall randomly on the square in Fig. 3.46, and let the number of drops which fall inside the quarter circle be r. Then

$$\frac{r}{1000} \approx \frac{\pi}{4} \quad \text{i.e.} \quad \pi \approx \frac{r}{250}$$

Fig. 3.47 gives the program. The variable i counts all the "raindrops" and r counts those which fall in the quarter circle. The core of the program is line 30. We show how it works.

First a point (x,y) = (RND, RND) is chosen at random within the square.

If the point lies inside the quarter circle, then $x^2 + y^2 < 1$, i.e. $[x^2 + y^2] = 0$ and we have $r \leftarrow r + 1 - 0$ i.e. the drop is counted.

If the point does not lie inside the quarter circle, then $1 < x^2 + y^2 < 2$ i.e. $[x^2 + y^2] = 1$ and so $r \leftarrow r + 1 - 1$ i.e. the drop is not counted.

The program gives poor estimates of π. It can be shown that

$$\pi - 0{\cdot}052 \leq \frac{r}{250} \leq \pi + 0{\cdot}052 \text{ with probability } 0{\cdot}683$$

and $\pi - 0{\cdot}104 \leq \frac{r}{250} \leq \pi + 0{\cdot}104$ with probability $0{\cdot}955$

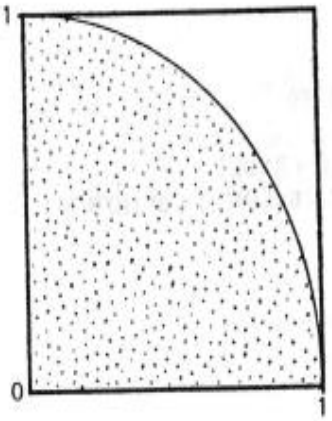


Fig. 3.46

```
10 R = 0
20 FOR I = 1 TO 1000
30       R = R + 1 - INT (RND↑2 + RND↑2)
40 NEXT I
50 PRINT R/250
60 END
```

3.104

Fig. 3.47

b) Buffon's n

A table ha

throw a ne

How great

The answe

below. I

s crossi

$\frac{s}{w} \approx$

We write

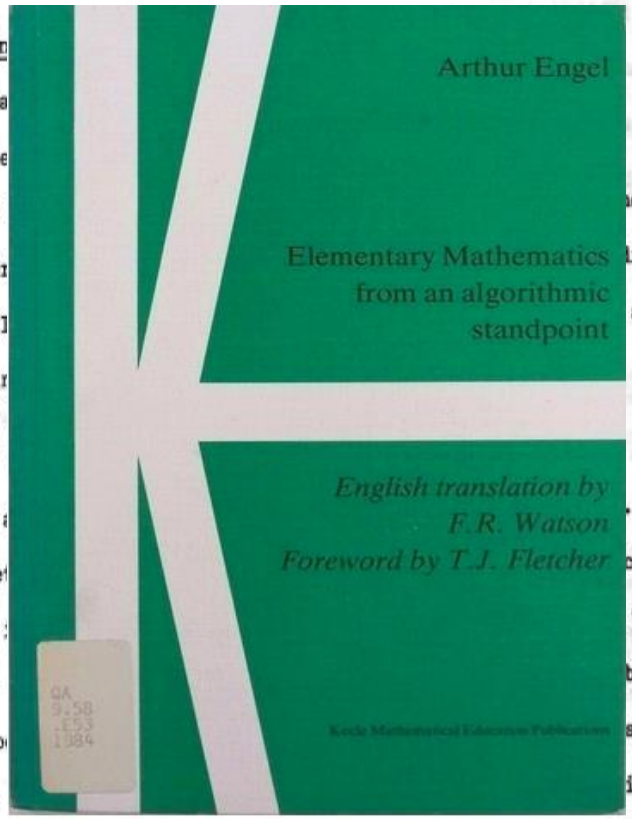
strips be

that one

abscissa

first cho

angle a



We

e lines'

lix

are

. All

consider

The

by

sing the

ite

$b = 0{\cdot}5 \cos a$, then the needle cuts one of the parallel lines precisely

Unterrichtspraxis: Monte Carlo Methode zur Bestimmung der Kreiszahl PI

Ziel dieser Aufgabe ist die näherungsweise Ermittlung der Kreiszahl PI (=Umfang / Durchmesser)
Lässt man einen Zufallsregen auf ein Quadrat prasseln und färbt die Tropfen, die in den Viertelkreis fallen schwarz, die anderen rot,
und zählt diese, so ergibt sich die Zahl PI aus folgendem Verhältnis:
Tropfen im Viertelkreis : Tropfen im Quadrat = r^2 PI/4 : r^2 bzw. PI/4.
Damit ist PI = Tropfen im Viertelkreis : Tropfen im Quadrat * 4
(Das gleiche Spiel lässt sich natürlich genausogut mit einem Vollkreis durchführen!)

Die Prozeduren:

Diese Prozedur ist zum Aufwärmen:
Die Zellen des 100x100 Quadrats werden zeilen- und zellenweise zufällig gefärbt.

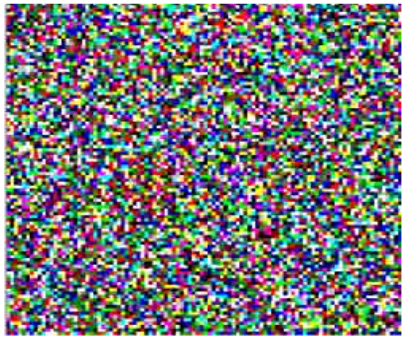
```
Sub CommandButton1_Click()
   Range(Cells(1, 1), Cells(100, 100)).Interior.ColorIndex = 2
   Randomize
   For z = 1 To 100
      For s = 1 To 100
        Cells(z, s).Interior.ColorIndex = Int(Rnd() * 16)
      Next
   Next
End Sub
```

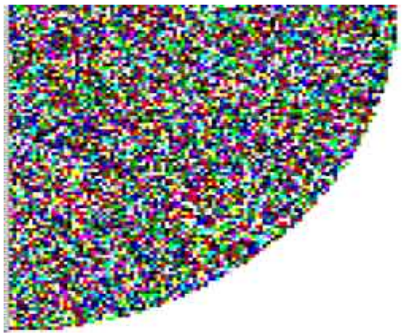
Diese Prozedur zeichnet einen bunten Viertelkreis!

```
Sub CommandButton2_Click()
   Range(Cells(1, 1), Cells(100, 100)).Interior.ColorIndex = 2
   Randomize
   For z = 1 To 100
      For s = 1 To 100
         If (z * z + s * s) < 10000 Then
            Cells(z, s).Interior.ColorIndex = Int(Rnd() * 16)
         End If
      Next
   Next
End Sub
```

Folgende Prozedur löst das Problem:
Die zufällig fallenden Tropfen innerhalb des Viertelkreises werden gezählt, der mit 4 multiplizierte Quotient (näherungsweise PI) wird am Ende ausgegeben.
Die gewünschte Tropfenanzahl wird in einer Textbox (textbox1) eingegeben.

```
Sub CommandButton3_Click()
  Randomize
  Range(Cells(1, 1), Cells(100, 100)).Interior.ColorIndex = 2
  tropfen = 0
  innerhalbkreis = 0
  anzahl = Val(TextBox1.Text)
  Do
    tropfen = tropfen + 1
    z = Int(Rnd * 100) + 1
    s = Int(Rnd * 100) + 1
    Cells(z, s) = Cells(z, s) + 1
    If (z * z + s * s) <= 10000 Then
          innerhalbkreis = innerhalbkreis + 1
          Cells(z, s).Interior.ColorIndex = 1
      Else
          Cells(z, s).Interior.ColorIndex = 3
    End If
  Loop Until tropfen > anzahl
  Label2.Caption = innerhalbkreis / tropfen * 4
End Sub
```





http://www.gym1.at/schulinformatik/unterrichtsinhalte/unterrichtsbeispiele/sprachen/vba/montecarlo/

Die Energie des Verstehens

<!selfhtml.wiki />

http://wiki.selfhtml.org/wiki/JavaScript/Objekte/Math

JavaScript/Objekte/Math

JavaScript Objekte

Mit dem Objekt `Math` können Sie Berechnungen, auch komplexe wissenschaftlicher oder kaufmännischer Art, durchführen. Dazu stehen Ihnen verschiedene mächtige Methoden und Funktionen sowie einige Eigenschaften zur Verfügung.

- E (Eulersche Konstante)
- LN2 (natürlicher Logarithmus von 2)
- LN10 (natürlicher Logarithmus von 10)
- LOG2E (konstanter Logarithmus von 2)
- LOG10E (konstanter Logarithmus von 10)
- PI (Konstante PI)
- SQRT1_2 (Konstante für Quadratwurzel aus 0,5)
- SQRT2 (Konstante für Quadratwurzel aus 2)

- abs() (positiver Wert)
- acos() (Arcuscosinus)
- asin() (Arcussinus)
- atan() (Arcustangens)
- ceil() (nächsthöhere ganze Zahl)
- cos() (Cosinus)
- exp() (Exponentialwert)
- floor() (nächstniedrigere ganze Zahl)
- log() (Anwendung des natürlichen Logarithmus)
- max() (größere von zwei Zahlen)
- min() (kleinere von zwei Zahlen)
- pow() (Zahl hoch Exponent)
- random() (0 bis 1 per Zufall)
- round() (kaufmännische Rundung einer Zahl)
- sin() (Sinus)
- sqrt() (Quadratwurzel)
- tan() (Tangens)

JavaScript Reference
Overview

JavaScript
JS String
JS Number
JS Operators
JS Statements
JS Math
JS Date
JS Array
JS Boolean
JS RegExp
JS Global

Browser BOM
Window
Navigator
Screen
History
Location

HTML DOM
DOM Document
DOM Elements
DOM Attributes
DOM Events

HTML Objects
<a>
<abbr>
<address>
<area>
<article>
<aside>
<audio>
<b>
<base>

JavaScript Math Reference

« Previous Next Reference »

Math Object

The Math object allows you to perform mathematical tasks.

Math is not a constructor. All properties/methods of Math can be called by using Math as an object, without creating it.

Syntax

```
var x = Math.PI;            // Returns PI
var y = Math.sqrt(16);      // Returns the square root of 16
```

For a tutorial about the Math object, read our JavaScript Math Tutorial.

Math Object Properties

Property	Description
E	Returns Euler's number (approx. 2.718)
LN2	Returns the natural logarithm of 2 (approx. 0.693)
LN10	Returns the natural logarithm of 10 (approx. 2.302)
LOG2E	Returns the base-2 logarithm of E (approx. 1.442)
LOG10E	Returns the base-10 logarithm of E (approx. 0.434)
PI	Returns PI (approx. 3.14)
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
SQRT2	Returns the square root of 2 (approx. 1.414)

Number of Digits: http://www.trans4mind.com/personal_development/JavaScript/longnumPiMachin.htm

1000

Add Count: ☑

Add Spaces: ☑

Calculate Pi

PI (1005)=3.
14159 26535 89793 23846 26433 83279 50288 41971 69399 37510 (50)
58209 74944 59230 78164 06286 20899 86280 34825 34211 70679 (100)
82148 08651 32823 06647 09384 46095 50582 23172 53594 08128 (150)
48111 74502 84102 70193 85211 05559 64462 29489 54930 38196 (200)
44288 10975 66593 34461 28475 64823 37867 83165 27120 19091 (250)
45648 56692 34603 48610 45432 66482 13393 60726 02491 41273 (300)
72458 70066 06315 58817 48815 20920 96282 92540 91715 36436 (350)
78925 90360 01133 05305 48820 46652 13841 46951 94151 16094 (400)
33057 27036 57595 91953 09218 61173 81932 61179 31051 18548 (450)
07446 23799 62749 56735 18857 52724 89122 79381 83011 94912 (500)
98336 73362 44065 66430 86021 39494 63952 24737 19070 21798 (550)
60943 70277 05392 17176 29317 67523 84674 81846 76694 05132 (600)
00056 81271 45263 56082 77857 71342 75778 96091 73637 17872 (650)
14684 40901 22495 34301 46549 58537 10507 92279 68925 89235 (700)
42019 95611 21290 21960 86403 44181 59813 62977 47713 09960 (750)
51870 72113 49999 99837 29780 49951 05973 17328 16096 31859 (800)
50244 59455 34690 83026 42522 30825 33446 85035 26193 11881 (850)
71010 00313 78387 52886 58753 32083 81420 61717 76691 47303 (900)
59825 34904 28755 46873 11595 62863 88235 37875 93751 95778 (950)
18577 80532 17122 68066 13001 92787 66111 95909 21642 01989 (1000)
38095

It took: 0.028 seconds

PROGRAMMIERSPRACHEN

Reden Sie mit Ihrem Computer?

Wenn Mensch und Maschine aneinander vorbeireden, kann das schon an der Definition von Sprache liegen. Denn Code ist eben nicht wie Französisch oder Chinesisch.

VON GERO VON RANDOW

3. Juni 2014 13:56 Uhr

50 Kommentare |

In Maschinen bauen wir immer ein Stück von uns selbst mit ein, meint ZEIT-Redakteur Gero von Randow, 61. Und das hat überraschende Effekte. | © Nicole Sturz/Cornelia Pflüger

Es gibt scheinbar abstrakte Fragen, die von höchst praktischer Bedeutung sind. Zu diesen Fragen gehört die, ob Programmiersprachen überhaupt Sprachen sind.

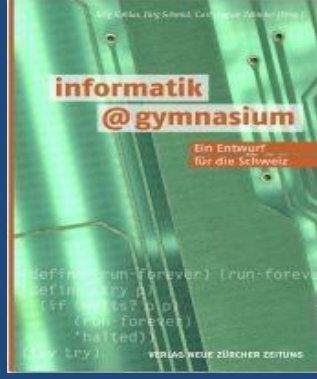
http://www.zeit.de/wissen/2014-06/programmiersprache-computersprache-software/komplettansicht

Über das Programmieren …

Die folgenden Folien können wegen ihrer vielen Buchstaben und Aussagen verwirren.

Lassen Sie Ihre Augen schweifen, ein paar Meinungen wirken und reflektieren Sie …

oder diskutieren Sie face-face mit.



Programmieren im engeren Sinn bedeutet,

- einem ******** ein gewünschtes Verhalten (die gewünschte Tätigkeit oder das Berechnungs vorgehen) unmissverständlich mitzuteilen.
- Für diese Auftragserteilung wurden *********sprachen entwickelt; sie dienen der Kommunikation zwischen Menschen und ******* zur Steuerung der *******.
- Das Programmieren nutzt ein Sortiment von grundlegenden Konstrukten wie beispielsweise Variablen, Schleifen, Verzweigungen und kennt Strukturierungsmethoden wie Objektorientierung und Nebenläufigkeit.

Programmieren im weiteren Sinn umfasst auch die Suche nach den geeigneten Algorithmen zur Lösung bestimmter Probleme.

Quelle: http://doebe.li/bibliothek/w00482.html

AUSSAGEN

> Programming, after all, is nothing if not a method of control.

von Nicholas G. Carr im Buch The Big Switch (2008)

> Programmieren war anfänglich ein intuitiv betriebenes Handwerk.

von Edsger W. Dijkstra, W. H. J. Feijen im Buch Methodik des Programmierens (1984)

> Zufälligerweise ist Programmieren eine Kunst, die relativ leicht zu erlernen ist.

von Joseph Weizenbaum im Buch Die Macht der Computer und die Ohnmacht der Vernunft (1976) im Text Gegen den Imperialismus der instrumentellen Vernunft auf Seite 362

> Programming is to CS what proof construction is to mathematics and what literary analysis is to English.

von James J. Lu, George H. L. Fletcher im Text Thinking about computational thinking (2009)

> Mit der ersten Sprache erlernt man nicht nur ein Vokabular und eine Grammatik, sondern man erschliesst sich eine Gedankenwelt.

von Niklaus Wirth im Buch Systematisches Programmieren (1972) auf Seite 8

> Programming is the sweet spot, the high leverage point in a digital society. If we don't learn to program, we risk being programmed ourselves.

von Douglas Rushkoff im Buch Program or Be Programmed (2010) im Text Purpose

> Die Vertreter des harten Programmierstils sind im Vorteil, weil sie die Fähigkeit und das Bedürfnis haben, ihren Stil theoretisch zu untermauern.

von Seymour Papert im Buch Revolution des Lernens (1993) im Text Computerspezialisten auf Seite 173

> Fast jeder, der zu einem sauberen methodischen Denken imstande ist, kann durch geringe Unterweisung und mit einiger Übung ein recht guter Programmierer werden.

von Joseph Weizenbaum im Buch Die Macht der Computer und die Ohnmacht der Vernunft (1976) im Text Gegen den Imperialismus der instrumentellen Vernunft auf Seite 362

> Der Versuch ein Computerprogramm zu schreiben, entlarvt unvermeidlich jeden ungenauen oder verschwommenen Gedanken und jeden impliziten Appell an das, was jeder für selbstverständlich hält.

von S. Dreyfus, Hubert L. Dreyfus im Buch Mind over Machine (1986) im Text Logische Maschinen und ihre Grenzen auf Seite 83

Und da das Programmieren fast unmittelbar zu Belohnungen verhilft, d. h. weil ein Computer sich sehr schnell in etwa der Weise verhält, wie der Programmierer das beabsichtigt hat, ist das Programmieren besonders für den Anfänger sehr verführerisch.

von Joseph Weizenbaum im Buch Die Macht der Computer und die Ohnmacht der Vernunft (1976) im Text Gegen den Imperialismus der instrumentellen Vernunft auf Seite 362

Außerdem spricht es in hohem Maße gerade solche Personen an, die noch nicht über genügend Reife verfügen, eine lange Zeitspanne zu ertragen, die zwischen einem bestimmten Aufwand zur Erreichung eines Ziels und dem Auftreten konkreter Erfolgsanzeichen liegt.

von Joseph Weizenbaum im Buch Die Macht der Computer und die Ohnmacht der Vernunft (1976) im Text Gegen den Imperialismus der instrumentellen Vernunft auf Seite 362

Programming is a somewhat allen activity, but it is emotionally very powerful. This power is what makes the Job of programming into something more akin to a calling, its jargon more like a distinct language, and the brotherhood of softwäre engineers into a cohesive culture.

von Alan Cooper im Buch The Inmates are Running the Asylum (2002) im Text An Obsolete Culture

Programming is a very difficult activity. Some of the difficulty is intrinsic to programming, but this research is based on the observation that programming languages make the task more difficult than necessary because they have been designed without careful attention to human-computer interaction issues.

von John F. Pane, Chotirat Ann Ratanamahatana, Brad Myers im Text Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems (2001)

Programming is used to implement algorithms on computers. While programming is a central activity in computer science, it is only a tool that provides a window into a much richer academic and professional field. That is, programming is to the study of computer science as literacy is to the study of literature.

von Fadi Deek, Jill Jones, Dennis McCowan, Chris Stephenson, Allen Tucker, Anita Verno in der Broschüre A Model Curriculum for K-12 Computer Science (2003)

Das Lernen "über den Computer", d. h. das Kennenlernen der Funktionsweise der Maschine ist - wie vieles andere auch - am besten im Zusammenhang mit der eigenen Erfahrung gewährleistet. Aus diesem Grunde ist das selbsttätige Programmieren unverzichtbar , weil es die Erfahrung vermitteln kann, da nicht "der Computer", sondern der Autor des Programms schuld ist, wenn die Ergebnisse nicht stimmen.

von Friedrich Oswald im Text Humanwissenschaftliche Aspekte in der Informationsgesellschaft

Es reicht gesunder Menschenverstand, um zu begreifen: Das Programmieren als Steuerung der Technik und Kommunikation mit dem Rechner in einer eigenen Sprache ist nicht nur der Prägnanz und Präzision im Ausdruck der Schüler förderlich, sondern trägt dadurch auch zu einer Verbesserung ihrer Kommunikationsfähigkeiten bei. Wer klar denkt und sich klar ausdrückt, kann auch kompetent kommunizieren.

von Juraj Hromkovic in der Zeitschrift Dossier des Schweizer Monat zum Thema: Einsen und Nullen - unsere Informationsgesellschaft. Juli/August 2012 (2012) im Text Digitale Analphabeten auf Seite 45

Vor meinem mathematischen Hintergrund fand ich die Programmierung sehr einfach und, weil alles sich im Endlichen bewegte, zuerst auch eher langweilig. Begeistert habe ich mich dann für die abstrakten Strukturen und Konzepte, die dahinter stehen, insbesondere für Programmiersprachen und Programmiermethodik, später für Softwarearchitektur. Dagegen habe ich nie die Faszination für den Computer selbst geteilt.

von Christiane Floyd im Buch Pioniere der Informatik (1999) im Text Christiane Floyd auf Seite 127

Programming should not, however, be essential in the teaching of computational thinking, nor should knowledge of programming be necessary to proclaim literacy in basic computer science. Just as math students come to proofs after 12 or more years of experience with basic math, and English students come to literary analysis after an even longer period of reading and writing, programming should begin for all students only after they have had substantial practice thinking computationally.

von James J. Lu, George H. L. Fletcher im Text Thinking about computational thinking (2009)

Das Programmieren ist auch ein nicht (mehr) sehr typischer Umgang mit Computern, da hier sehr viel mehr und sehr viel deutlicher strukturelles und erfindendes7 Denken gefordert und weniger gefördert wird. Die algorithmische und/oder objektorientierte Modellierung eines Problems ist jeweils nur eine spezifische Form des problemlösenden Denkens, die nicht von allen Schülern in gleicher Weise geleistet werden kann. Das Programmieren ist nicht nur eine hochspezialisierte sondern eine viele Schüler wenig motivierende Tätigkeit. Was manche mit viel Spaß betreiben, sehen andere, nach vorläufigen Eindruck die große Mehrheit der Schüler, als sinnfreies Tun.

von Dieter Engbring im Konferenz-Band Informatik in Bildung und Beruf im Text Was ist/kann/soll Informatikunterricht? (2011) auf Seite 102

Programmers are somehow different from ordinary people. Their stereotypical behavioral differences have been the subject of jokes for years: the social awkwardness, the pocket protectors, the bookish manner. Those are just the easily noticeable - and easily ridiculed - surface differences. The really Substantive differences are not only far subtler, but they have a more profound effect on the cognitive friction-rich interactive products that programmers build. Many observers of the Computer industry have taken pains to point out these differences. Robert Cringely calls programmers "stinking gods among men," referring simultaneously to their superior attitudes and their hygiene habits.

von Alan Cooper im Buch The Inmates are Running the Asylum (2002) im Text Homo Logicus auf Seite 95

Informatik darf sicherlich nicht auf Programmieren reduziert werden. Einblick in die Pro-grammierung ist jedoch nötig, um überhaupt zu verstehen, warum ein (im Wesentlichen) von-Neumann-Computer jene universelle Informationsverarbeitungsmaschine ist, die heutige Computer vom Palmtop bis zum Großrechner (ja selbst funktionsstarke Handys) nun einmal sind. Ohne Grundverständnis in Programmierung werden letztlich Konzepte der Praktischen Informatik, handle es sich um Betriebssysteme, Rechnernetze, Datenbanken, das Web oder auch um künstliche Intelligenz, kaum verstanden werden und stets mit einer Aura des Mysti-schen umgeben bleiben. Mystifizierung ist aber das Letzte, das guter Informatikunterricht erzielen sollte.

von Karin Hodnigg, Roland Mittermeir im Konferenz-Band 25 Jahre Schulinformatik (2010) im Text Konzept einer stufenweisen Fortbildung für InformatiklehrerInnen

Oft wird behauptet, bei der Medienkompetenz handle es sich um eine "Schlüsselkompetenz", "Kernkompetenz« bzw. "Kulturtechnik". Bei Licht betrachtet, sind mit Medienkompetenz jedoch weder das Programmieren noch logisches Denkvermögen (Boolsche Algebra) noch andere grundlegende mit Bildschirmmedien verbundene intellektuelle Fähigkeiten gemeint, sondern zunächst einmal nichts weiter als oberflächliche Kenntnisse verbreiteter Anwender-Software. Wer dies nicht glaubt, sollte einmal nachsehen, was im Fach "Informationstechnik" tatsächlich gelehrt wird, wenn Schüler mit dem Computer arbeiten: die Schwächen der Produkte der weltgrößten Software-Firma - Word, Excel und PowerPoint. Wer also Computer literacy mit literacy gleichsetzt, der erhebt das Beherrschen einiger Tricks und vor allem den Umgang mit vielen Problemen und Fehlern von Produkten der Firma Microsoft in seiner Bedeutung auf eine Stufe mit dem Lesen von Goethe und dem Schreiben von Aufsätzen. Dies ist ein ungeheuerlicher Vorgang!

von Manfred Spitzer im Buch Digitale Demenz (2012) im Text Was tun? auf Seite 310

Eine Gruppe von Menschen erschreckt das Programmieren niemals: Kinder. Kinder kommen zum Programmieren wie Enten zum Wasser, insbesondere wenn man sie vorsichtig hinführt. Das Programmieren ist nichts anderes als eine von vielen normalen Computeraktivitäten. Es erfordert kein geheimnisvolles Spezialwissen. Doch sobald eine Aktivität erst einmal so gebrandmarkt ist wie das rrogrammieren, wird seine Unerreichbarkeit für die normale Menschheit zu einer sich selbst erfüllenden Prophezeiung. Die Botschaft verbreitet sich, dass es eine schrecklich schwierige Sache ist und so zittern die Leute davor, anstatt es zu versuchen. Bücher über das Programmieren sind in einer Sprache geschrieben, die nur Spezialisten zugänglich ist. Natürlich dienen die Inhalte des Programmierens eher den Zielsetzungen von Spezialisten als den Bedürfnissen eines breiteren Anwenderpublikums. Doch Kinder werden von diesen Vorbehalten keineswegs erschreckt und beeinflusst. Sie wollen das Programmieren lernen, weil sie danach streben, den Computer zu beherrschen.

von Seymour Papert im Buch The Connected Family (1996) auf Seite 22

UND?
Programmieren aus Ihrer/Deiner Sicht?

PROGRAMMIEREN ALLGEMEINBILDEND?

In der Tat ist das Programmieren eine hochspezialisierte Tätigkeit, die nicht jeder beherrschen muss und schon deswegen wenig allgemein bildend ist.

von Dieter Engbring im Konferenz-Band Informatik in Bildung und Beruf im Text Was ist/kann/soll Informatikunterricht? (2011)

Heute ist es akzeptabel, dass jemand nicht mit einem Computer umgehen oder Programmcodes verstehen kann, in 20 Jahren wird das so wirken, als ob er ein Analphabet wäre.

von Clay Johnson im Text Man kann auch umfassend falsch informiert sein (2012)

Wer in der globalen Welt mit wenig Kapital ein Produkt entwickeln will, muss im Virtuellen denken und agieren, also programmieren können. Programmieren gehört zur Allgemeinbildung. Punkt.

von Ruedi Noser in der Zeitschrift Dossier des Schweizer Monat zum Thema: Einsen und Nullen - unsere Informationsgesellschaft. Juli/August 2012 (2012) im Text «Das ist wirklich vordigital!»

Ja, Programmieren gehört zur Allgemeinbildung und zwar nicht, weil man heute und in Zukunft dauernd selbst programmieren muss, sondern damit man das Potenzial von Computerprogrammen abschätzen kann.

von Beat Döbeli Honegger, erfasst im Biblionetz am 23.07.2012

The "everyone should learn to code" movement isn't just wrong because it falsely equates coding with essential life skills like reading, writing, and math. I wish. It is wrong in so many other ways.

von Jeff Atwood im Text Please Don't Learn to Code (2012)

We do not know how to program our computers, nor do we care. We spend much more time and energy trying to fi gure out how to use them to program one another instead. And this is potentially a grave mistake.

von Douglas Rushkoff im Buch Program or Be Programmed (2010) auf Seite 18

Schwer schließlich ist das Programmieren - aber das müssen nicht sehr viele Leute in dieser Gesellschaft tun, es müssen sich nur alle einmal an einem Programm versucht haben, um zu verstehen, was da vor sich geht.

von Hartmut von Hentig im Buch Die Schule neu denken (1993) im Text Schwierige Veränderungen auf Seite 69

There has been much discussion of whether the primary aim should be to teach children programming for its own sake, or to use programming in the service of some other end or discipline 'programming to learn, or learning to program'.

von Patrick Mendelsohn, T.R.G. Green, P. Brna im Buch Psychology of Programming (1990) im Text Programming Languages in Education

[Es] sollte selbstverständlich sein, dass jeder Maturand eine Programmiersprache beherrscht und damit das Abstraktionsniveau eines Computerprogramms verstehen kann. Davon sind wir in unserem Bildungssystem jedoch meilenweit entfernt.

von Ruedi Noser in der Zeitschrift Dossier des Schweizer Monat zum Thema: Einsen und Nullen - unsere Informationsgesellschaft. Juli/August 2012 (2012) im Text «Das ist wirklich vordigital!»

In the emerging, highly programmed landscape ahead, you will either create the soft ware or you will be the software. It's really that simple: Program, or be programmed. Choose the former, and you gain access to the control panel of civilization.

von Douglas Rushkoff im Buch Program or Be Programmed (2010) im Text Introduction auf Seite 7

When human beings acquired language, we learned not just how to listen but how to speak. When we gained literacy, we learned not just how to read but how to write. And as we move into an increasingly digital reality, we must learn not just how to use programs but how to make them.

von Douglas Rushkoff im Buch Program or Be Programmed (2010) im Text Introduction auf Seite 7

Die Frage "Programmieren, ja oder nein ?" ist ganz einfach falsch gestellt. Es geht im Gymnasium gar nicht darum, sondern zunächst generell um die Frage "Was gehört zur Allgemeinbildung?" und speziell auf die Informatik bezogen, "Was muss jeder Maturand über Informatik wissen ?"

von Walter Gander im Text Informatikunterricht - Informatikstudium (1992)

When human beings acquired language, we learned not just how to listen but how to speak. When we gained literacy, we learned not just how to read but how to write. And as we move into an increasingly digital reality, we must learn not just how to use programs but how to make them.

von Douglas Rushkoff im Text Why Johnny Can't Programm (2010)

I would like to stress that I do not presume to assign a universal educative core to programming. But I see programming as a representative for many of the principles that must be part of informatics education in order to prepare the learners for an increasingly information-centred world.

von Peter K. Antonitsch im Konferenz-Band From Computer Literacy to Informatics Fundamentals (2005) im Text Standard Software as Microworld? auf Seite 190

It assumes that more code in the world is an inherently desirable thing. In my thirty year career as a programmer, I have found this ... not to be the case. Should you learn to write code? No, I can't get behind that. You should be learning to write as little code as possible. Ideally none.

von Jeff Atwood im Text Please Don't Learn to Code (2012)

It assumes that coding is the goal. Software developers tend to be software addicts who think their job is to write code. But it's not. Their job is to solve problems. Don't celebrate the creation of code, celebrate the creation of solutions. We have way too many coders addicted to doing just one more line of code already.

von Jeff Atwood im Text Please Don't Learn to Code (2012)

Unterricht, der auf das Programmieren-Lernen und auf die Syntax von Programmiersprachen gerichtet ist, hat wenig mit "Allgemeinbildung" zu tun. Dies ist Spezialwissen, das für die meisten Schülerinnen nicht interessant ist, sondern zur Abgrenzung der "Expertinnen" beiträgt. Dennoch war die Praxis des Informatikunterrichts lange Zeit vom Programmieren-Lernen geprägt.

von Heidi Schelhowe im Buch Technologie, Imagination und Lernen (2007) im Text Zum (Zu)Stand von Bildung in der Wissensgesellschaft und zur Rolle des Computers

Das Lernen "über den Computer", d. h. das Kennenlernen der Funktionsweise der Maschine ist - wie vieles andere auch - am besten im Zusammenhang mit der eigenen Erfahrung gewährleistet. Aus diesem Grunde ist das selbsttätige Programmieren unverzichtbar , weil es die Erfahrung vermitteln kann, da nicht "der Computer", sondern der Autor des Programms schuld ist, wenn die Ergebnisse nicht stimmen.

von Friedrich Oswald im Text Humanwissenschaftliche Aspekte in der Informationsgesellschaft

Im Zeitalter der Anwenderpakete wird das Programmieren nicht mehr primär als Werkzeug benötigt, sondern als Gedankengut, das den vernünftigen Einsatz der Werkzeuge ermöglicht, die von anderen erstellt wurden. Eine ähnliche Aussage gilt für jede Art von Allgemeinbildung. Allgemeinbildung beinhaltet ein Gedankengut, das man selten für direkten Nutzen einsetzt, das einem aber erlaubt, Detailkenntnisse von transienter Bedeutung im Tagesgeschäft vernünftig einzusetzen.

von Raimond Reichert, Jürg Nievergelt, Werner Hartmann im Buch Programmieren mit Kara (2003) im Text Programmieren im Unterricht - warum und wie? auf Seite 6

Im Zeitalter der Anwenderpakete wird damit das Programmieren nicht mehr nur als Werkzeug benötigt, sondern als Gedankengut, das den vernünftigen Einsatz der Werkzeuge ermöglicht, die von anderen erstellt wurden. Eine ähnliche Aussage gilt für jede Art von Allgemeinbildung. Denn darunter versteht man Gedankengut, das man selten für direkten Nutzen einsetzt, das einem aber eine Geisteshaltung erlaubt, um Detailkenntnisse von transienter Bedeutung im Tagesgeschäft vernünftig einzusetzen.

von Jürg Nievergelt im Text Roboter programmieren - ein Kinderspiel (1999)

I suppose I can support learning a tiny bit about programming just so you can recognize what code is, and when code might be an appropriate way to approach a problem you have. But I can also recognize plumbing problems when I see them without any particular training in the area. The general populace (and its political leadership) could probably benefit most of all from a basic understanding of how computers, and the Internet, work. Being able to get around on the Internet is becoming a basic life skill, and we should be worried about fixing that first and most of all, before we start jumping all the way into code.

von Jeff Atwood im Text Please Don't Learn to Code (2012)

Teaching everyone on campus to program is a noble goal, put forth by Alan Perlis in 1962. Perlis, who was awarded the first ACM A.M. Turing Award, said that everyone should learn to program as part of a liberal education. He argued that programming was an exploration of process, a topic that concerned everyone, and that the automated execution of process by machine was going to change everything. He saw programming as a step toward understanding a "theory of computation," which would lead to students recasting their understanding of a wide variety of topics (such as calculus and economics) in terms of computation.

von Mark Guzdial im Text Paving the Way for Computational Thinking (2008)

Der politische Diskurs um Informationstechnologie und Bildung dreht sich seit Jahrzehnten im Kreis. Mal ist Informatik in, mal wieder out. Das nervt Döbeli Honegger. Er wünscht sich einen Informatikunterricht, der allen Schülern die, nun ja, Basics beibringt: "Zu wissen, wie Computer programmiert werden, gehört heute schlicht zur Allgemeinbildung. So wie man in der Schule ins Chemielabor geht – und eben nicht nur Bücher darüber liest oder Filme schaut – und trotzdem nicht zum Chemiker wird, genauso sollte man heute auch selbst mal programmiert haben." Im Alltag müsse man heute nicht programmieren können, "aber um die heutige Welt zu verstehen".

von Pavel Lokshin im Text Erziehung zur digitalen Mündigkeit (2014)

Das Programmieren ist auch ein nicht (mehr) sehr typischer Umgang mit Computern, da hier sehr viel mehr und sehr viel deutlicher strukturelles und erfindendes7 Denken gefordert und weniger gefördert wird. Die algorithmische und/oder objektorientierte Modellierung eines Problems ist jeweils nur eine spezifische Form des problemlösenden Denkens, die nicht von allen Schülern in gleicher Weise geleistet werden kann. Das Programmieren ist nicht nur eine hochspezialisierte sondern eine viele Schüler wenig motivierende Tätigkeit. Was manche mit viel Spaß betreiben, sehen andere, nach vorläufigen Eindruck die große Mehrheit der Schüler, als sinnfreies Tun.

von Dieter Engbring im Konferenz-Band Informatik in Bildung und Beruf im Text Was ist/kann/soll Informatikunterricht? (2011) auf Seite 102

Wenn ein Schüler nicht ein eigentlicher Computer-Freak ist und später Informatiker werden will, wird er auch später eher ein Anwender von fertigen Programmen sein. Er wird weder die Fähigkeiten noch genügend Zeit haben, um Programme auszuarbeiten, die mit professioneller Software Schritt halten. Für ihn ist es deshalb wichtiger, die Struktur dieser Disziplin zu kennen, nämlich die grundlegende Denkweise, welche dem Programmieren zugrundeliegt. Die Details der Programmiertechnik helfen ihm dagegen kaum viel weiter. In diesem Sinne, aber auch mit den dadurch gekennzeichneten Grenzen, wäre ein elementares Programmieren sicher Teil jeder Grundausbildung mit Computern.

von Heinz Moser im Buch Der Computer vor der Schultür (1986) im Text Tipps und Ratschläge auf Seite 133

„Jeder Zehnjährige ein Programmierer!?" Es ist der Reflex gegen Diktate „der Wirtschaft" an die Schule. Und wäre Programmieren eine Technik wie Dentalhygiene oder Betonmischen, der Reflex hätte komplett Recht. Nur ist Informatik keine solche Technik, sie steuert Technik, sie ist die neueste Einstellung zur Welt, eine junge Wissenschaft, das sogenannte Computational Thinking, das sich in unwahrscheinlich kurzer Zeit durchgesetzt hat, unser Leben durchdringt, den Alltag regelt, Wirtschaft und Konsum steuert, globale Kommunikation lenkt. So dass uns nur diese Wahl bleibt: Entweder wir lassen uns willfährig lenken durch Programme einer Denkweise, die uns schleierhaft bleibt – oder wir lernen kennen, was uns lenkt. Schmeckt nach Entweder - Oder: Unmündigkeit oder (mögliche) Freiheit.

von Ludwig Hasler im Text Informatik und Bildung - eine philosophische Annäherung (2013)

Programmieren als notwendiger Bestandteil der täglichen Arbeit mit dem Computer, das war einmal, und kommt nicht wieder zurück. Darin gehen wir mit der heute weit verbreiteten Meinung einig. Wir denken aber, dass in der Ausbildung vielerorts das Kind mit dem Bade ausgeschüttet wurde, als deswegen das Programmieren aus den Lehrplänen gleich ganz entfernt wurde. Wenn ein Werkzeug so allgegenwärtig geworden ist, wie es der Computer heute ist, und wenn wir ihm so viele wichtige Entscheide überlassen, dann ist es angebracht, dass gebildete Menschen etwas über die Kernideen verstehen, mittels denen wir dem Computer unsere Wünsche beibringen. Programme sind die Formulierungen solcher Wünsche, und eigene Erfahrungen im Programmieren als Teil der Allgemeinbildung ist der Weg zum Verständnis der Arbeitsweise von Computern.

von Werner Hartmann, Jürg Nievergelt, Raimond Reichert im Buch Programmieren mit Kara (2003) im Text Und die Moral von der Geschicht? auf Seite 125


As we see it, digital fluency requires not just the ability to chat, browse, and interact but also the ability to design, create, and invent with new media, as BalaBethany did in her projects. To do so, you need to learn some type of programming. The ability to program provides important benefits. For example, it greatly expands the range of what you can create (and how you can express yourself) with the computer. It also expands the range of what you can learn. In particular, programming supports "computational thinking," helping you learn important problem-solving and design strategies (such as modularization and iterative design) that carry over to nonprogramming domains. And since programming involves the creation of external representations of your problem-solving processes, programming provides you with opportunities to reflect on your own thinking, even to think about thinking itself.

von Mitchel Resnick, John Maloney, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, Yasmin B. Kafai, Andrés Monroy-Hernández im Text Scratch: Programming for All (2009)

Programming is more than just a useful skill of a computer scientist. Learning programming means learning a language of communication with technical systems, learning to tell a machine what activity we would like to have from it. Since machines do not have any intelligence, our instructions must be so clearly and unambiguously formulated that no mistake can arise. In this way, the pupils learn to describe ways and methods for achieving aims that can be correctly followed by everybody without needing to provide the knowledge why they successfully achieve these goals. The development of this skill essentially contributes to the pupils' natural language skills by motivating pupils to properly think about how to best express what they would like to communicate. After supplementing the programming courses with some elementary data structures and algorithms, we propose to switch to the fundamentals.

von Juraj Hromkovic im Konferenz-Band Informatics Education - The Bridge between Using and Understanding Computers im Text Contributing to General Education by Teaching Informatics (2006)

Nico Lumma

Ja, das klingt merkwürdig, aber lassen Sie die Forderung nach einer Programmiersprache als zweite Fremdsprache einmal sacken. Ich habe in der Schule Latein gehabt und das hat mir trotz aller Probleme beim deklinieren und bei der Grammatik insgesamt wenigstens für mein Studium der Politikwissenschaft und Geschichte ein gewisses Rüstzeug mitgegeben.

Aber, für den Alltag hat es abgesehen vom Lesen der Asterix-Hefte meiner Kinder für mich kaum noch einen praktischen Nutzwert. Lediglich der Vokabelschatz hilft ab und zu mal, kurze Sätze auf Französisch, Italienisch oder Spanisch zu verstehen. Nichts, was Google Translate und andere nicht viel besser könnten – bei Facebook, dem Ort an dem wir mit Freunden und Bekannten kommunizieren, ist bei Postings in anderen Sprachen schon jetzt die Möglichkeit vorhanden, sich den Text übersetzen zu lassen.

Für Kinder und Jugendliche allerdings wird künftig immer weniger eine Rolle spielen, die Klassiker der Vergangenheit im Original lesen zu können, sondern sie sollen den Code der Zukunft besser verstehen. Vor einigen Jahren hiess es: "Lern spanisch, das ist das neue Englisch!" – Mittlerweile entwickeln Kinder Apps für das iPhone.

Marc Andreesen, der uns als Student mit der Entwicklung des ersten massentauglichen Browsers namens Mosaic die bunte Welt des World Wide Web erst zugänglich gemacht hatte und investiert aktuell als Venture Capital Geber in Internet-basierende Geschäftsmodelle, schrieb kürzlich in einem viel beachteten Essay im Wallstreet Journal: "Software is eating the world."

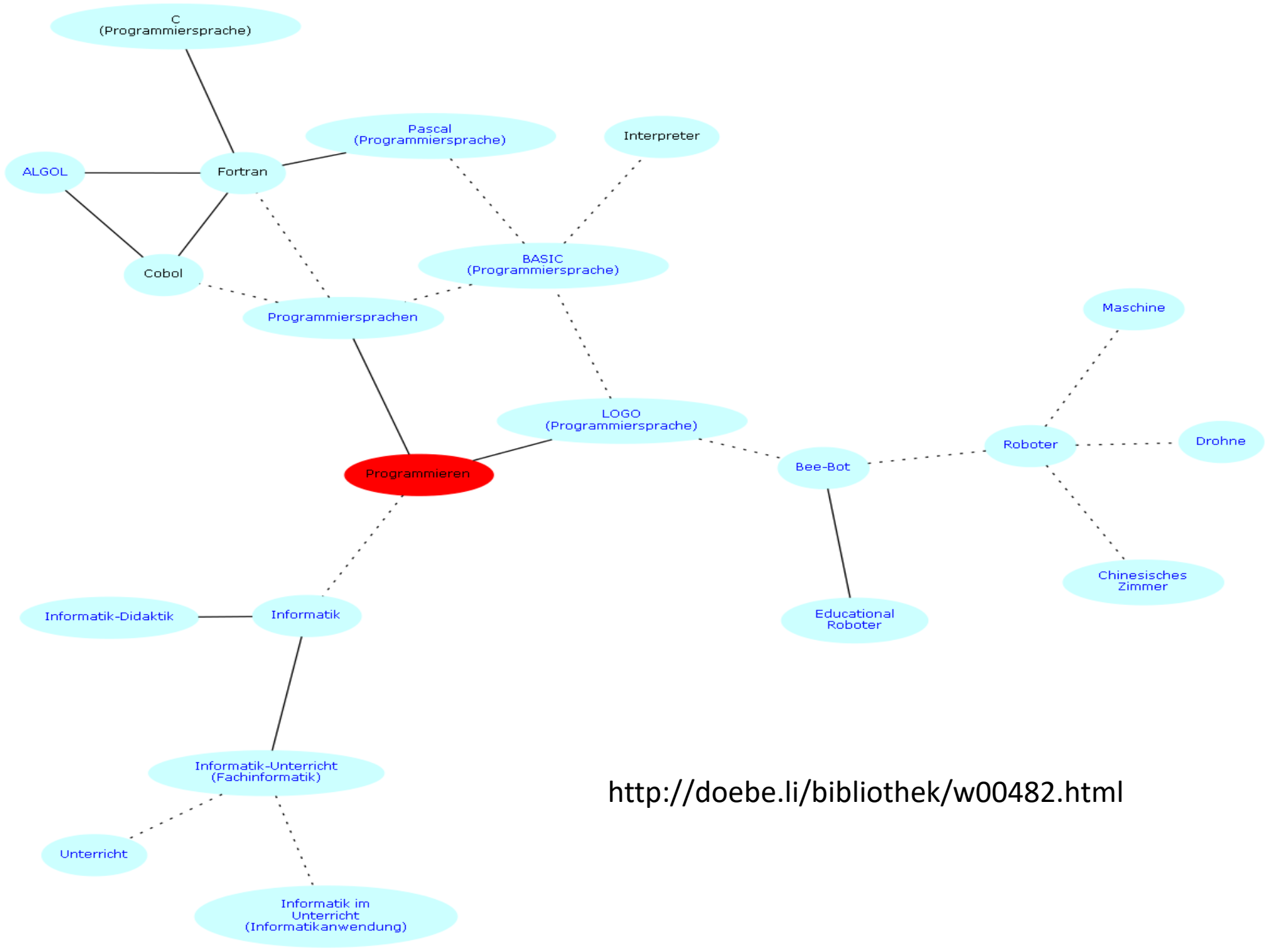
Software ist das geworden, was die Welt zusammenhält, aber auch immer das Potential hat, sie auseinander zu reissen, meine Damen und Herren.

Daher ist es für Kinder und Jugendliche essentiell, zu lernen, wie Software funktioniert, also wie man programmiert. Ich erwarte jetzt nicht ein Volk von Programmierern, aber ebenso wie uns das Erlernen einer Sprache den Zugang zu anderen Kulturen eröffnet und damit unseren eigenen persönlichen Horizont erweitern kann, ermöglicht das Erlernen einer modernen Programmiersprache ein besseres Verständnis für die komplexen Abläufe der Zukunft, die allesamt ohne Software nicht mehr möglich sind.

Der Autor Douglas Rushkoff hat in seinem Buch "Program or be programmed" darauf hingewiesen, dass wir als Gesellschaft darauf achten müssen, nicht bloß zum Objekt zu werden, sondern dass wir selber in der Lage sein müssen, zu verstehen, was um uns herum passiert.

Javascript ist das neue Latein, meine Damen und Herren, wir sollten uns zügig daher überlegen, wie wir die Wissensvermittlung bei Kindern und Jugendlichen dahingehend verändern, dass wir das Erlernen einer modernen Programmiersprache mit in die Lehrpläne aufnehmen – denn wir wollen doch alle, dass die nachwachsenden Generationen das Rüstzeug für die Zukunft erhalten.

Jetzt denken etliche von Ihnen im Saal "Was für ein Quatsch, wie soll das denn gehen?" – aber werfen Sie bitte einen Blick nach Estland, das jetzt bereits jedem Schüler anbietet, in der Schule eine Programmiersprache zu erlernen. Es ist machbar, man muss es nur wollen.



http://doebe.li/bibliothek/w00482.html

Informatik auf SwissEduc
» Programmiersprachen

Karatojava
Turtles (verschiedene Sprachen)

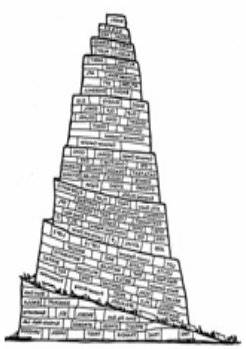
GameGrid: Spiele in Java programmieren
Game-Apps für Android-Smartphones
Handys programmieren in Java
Turtlegrafik in Java
Android - Turtlegrafik
Lego-Robotik mit Java
TigerJython

Scratch - Werkstatt
Pledge-Algorithmus
Java-Kurs für Einsteiger
Java: Tic Tac Toe
Povray: Grafik programmieren
Rekursive Programme

Geschichte der Programmiersprachen
Sprachen: Übersicht
Programmier-Paradigmen
Corewars - Duell der Programme
Einführung in Prolog
Brainfuck - esoterische Programmiersprache
Dynamische Webseiten
XML: Einführung

Programmiersprachen – gestern, heute, morgen

Verfasst von Hermann Lehner



Turm zu Babel der Programmiersprachen,
angelehnt an Sammet 1969

Inhalt	Geschichte der Programmiersprachen
Voraussetzung	mindestens ein Jahr Programmiererfahrung; grundlegende Begriffe wie Syntax, Semantik, Compiler, Interpreter, Assembler, HW-Architektur, etc.
Schultyp	Fachhochschule, Hochschule
Dauer	ca. 6 Lektionen

Worum geht es?

Sie kennen sicher die heute gängigen Sprachen wie C, Java oder Pascal. Doch wie kam es dazu? Wer hat diese Programmiersprachen erfunden? Und – noch viel interessanter – warum? Über die Entstehung von Programmiersprachen nachzudenken wirft viele Fragen auf. Betrachten wir zum Beispiel die Sprache C. Eine ziemlich betagte Sprache aus dem Jahre 1971! Im Internet gibt es heftige Diskussionen darüber, wie der aller erste C-Compiler geschrieben wurde. Wurde er in C geschrieben? Wie hätte er kompiliert werden sollen, wenn es der erste Compiler für C ist? Das typische Huhn-Ei Problem. Nun, der Compiler könnte in einer noch älteren Sprache geschrieben worden sein. Doch wie wurde der erste Compiler der ersten Programmiersprache geschrieben? Und welche Sprache war das? Was war davor?

Downloads

Information für die Lehrperson	PDF [154 KB] · Word [145 KB]
Information für die Studierenden	PDF [298 KB] · Word [286 KB]
Literatur	PDF [30 KB] · Word [29 KB]

Eine andere Einordnung

Was heißt Programmieren?
Was bedeutet Codieren?
Was bedeutet Modellieren?
Was ist Implementieren?

Fachsprache ?!

Nachdem wir das Klassendiagramm erstellt haben, wollen wir im nächsten Schritt die Klasse implementieren

Achten Sie beim Programmieren besonders auf Groß- & klein-Schreibung der Klassen und Variablen.

Implementieren ist die Umsetzung der theoretisch geplanten Konzepte aus der Modellierung heraus.
→ kompletter Handlungsschritt

Programmieren ist das Erzeugen von Quellcode
→ Handlung

Modellieren Sie die Lösung der Aufgabe in UML!

Nutzen Sie die ASCII-Tabelle Tabelle um den folgenden Satz \ das Zeichen A zu kodieren!

Modellieren ist die versuchte Abbildung eines Problems aus der Realität

→ UML: Entwurfsdiagramm
Objektdiagramm
Sequenzdiagramm
:

Kodieren ist die Umwandlung von einer Darstellung in eine andere Darstellung

→ z.B ASCII
UTF8
Hex

„Kreuz"

Modellieren:

Abbildung der Wirklichkeit
Entwicklung von Konzepten
Entwurf der Software

„Modellieren Sie die benötigten Daten als Klassenmodell in Uml."

Implementieren:

Umsetzung der Planung
Umsetzung des Softwareentwurfs

„Implementieren Sie das Klassenmodell in C# / Java."

Programmieren:

Erzeugung von Programmcode

„Programmieren Sie den Gauß-Algorithmus."

Codieren:

Umsetzung von Struktogrammen / PAP's in Programmcode

—

Was bedeutet das?

programmieren:

- alles was zur Umsetzung in ein Programm nötig ist. (eher umgangssprachlich)
- „Programmiert das "Nim"-Spiel."

implementieren:

- Ein Modell konkret in eine Programmiersprache umsetzen (mit Tests).
- „Wir implementieren Bubble Sort in Java."

modellieren:

- Sachzusammenhang analysieren, abstrahieren und beschreiben/darstellen.
- „Wir modellieren die Bücherleihe als ER-Diagramm."

codieren:

- Umsetzen von einer Darstellung in eine andere
- „[illegible] Codiere "A" im ASCII-Code.

Satz

Implementieren Sie die vorhandenen textlichen Befragen in die ~~vorhandene~~ Seiten-struktur.

Modellieren Sie einen Algorithmus mithilfe des Struktogramms

Wir wollen 23 als Binärzahl codieren!

Beim Programmieren soll man unbedingt auf die korrekte Syntax achten

Definition

1 implementieren:
Ein Modell in einer Programmiersprache umsetzen!

modellieren:
Das vereinfachen eines realen Problems in eine abstrakte Darstellung.

4 codieren
Die Übersetzen einer Information in eine andere Darstellung!

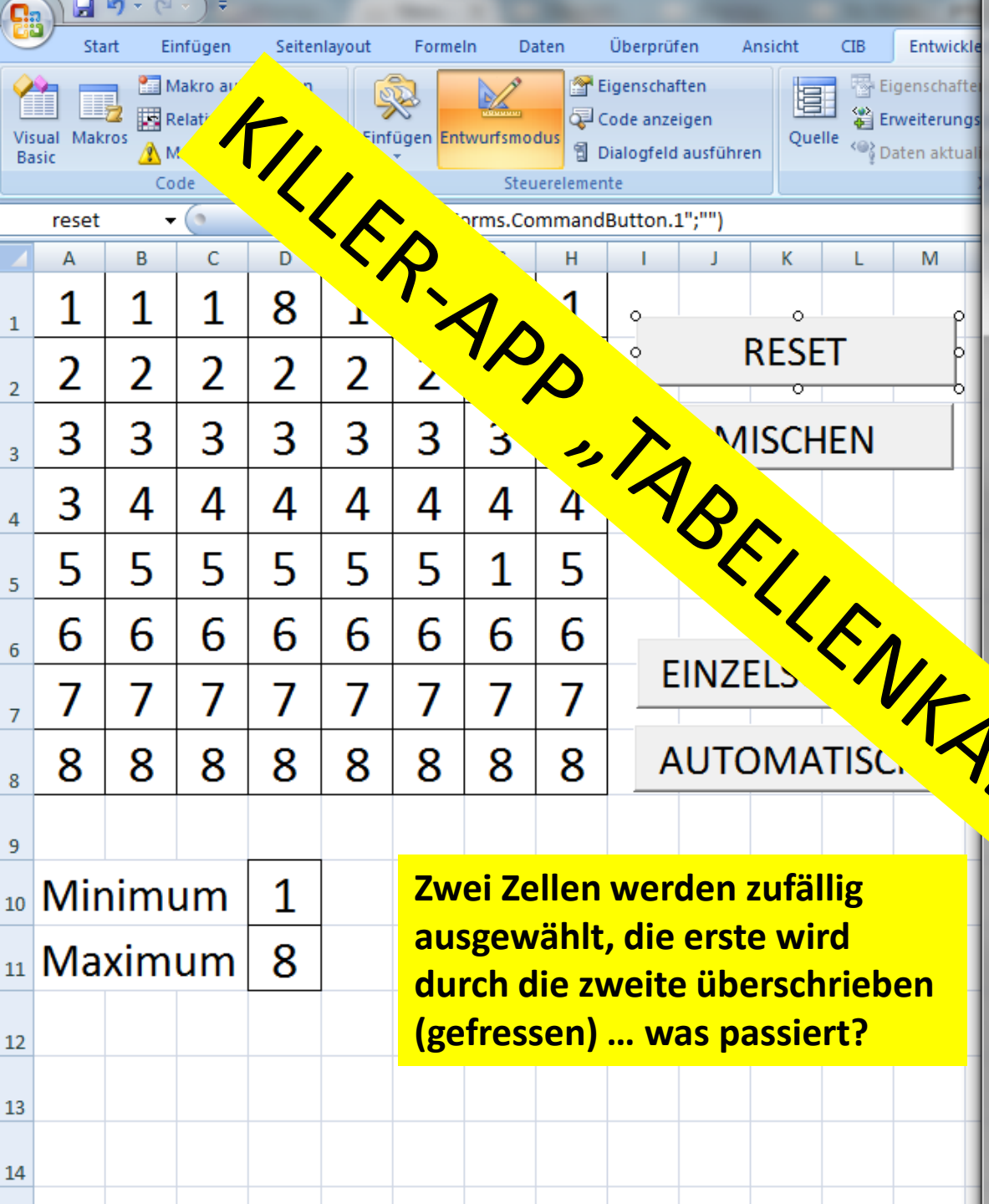
3 programmieren
Programmieren ist das Umsetzen eines Modells in eine Anwendung.

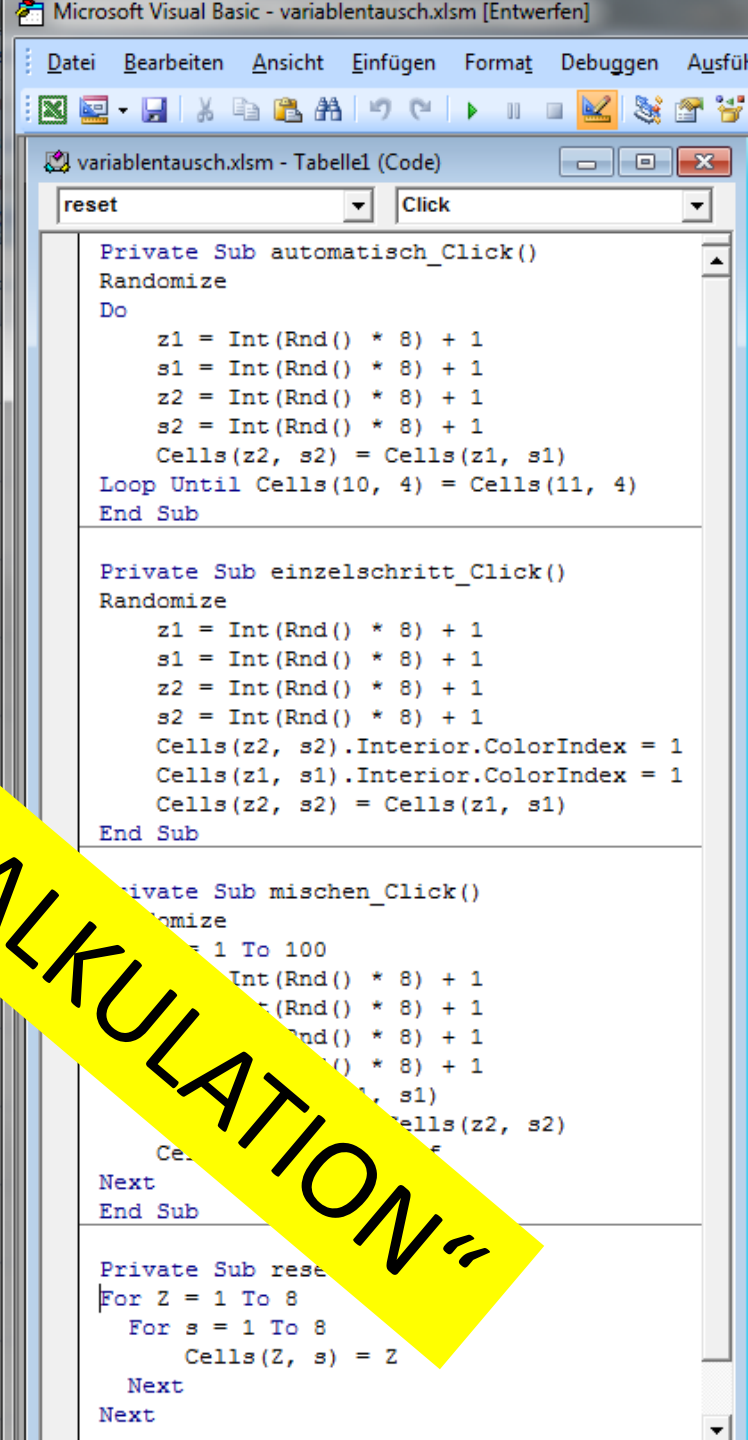
PROGRAMMIERWERKZEUGE

Ob für Biker, Angler, Golfer, Pfadfinder oder Wanderer. Das Sammlermesser Giant verfügt über **87 Werkzeuge mit 141 Funktionen**.

Bitte auf Amazon selber suchen …

KILLER-APP „TABELLENKALKULATION“

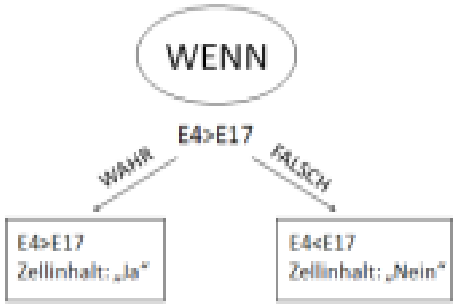




Zwei Zellen werden zufällig ausgewählt, die erste wird durch die zweite überschrieben (gefressen) … was passiert?

EIN MATURABEISPIEL

Thema: FUNKTIONALE MODELLIERUNG

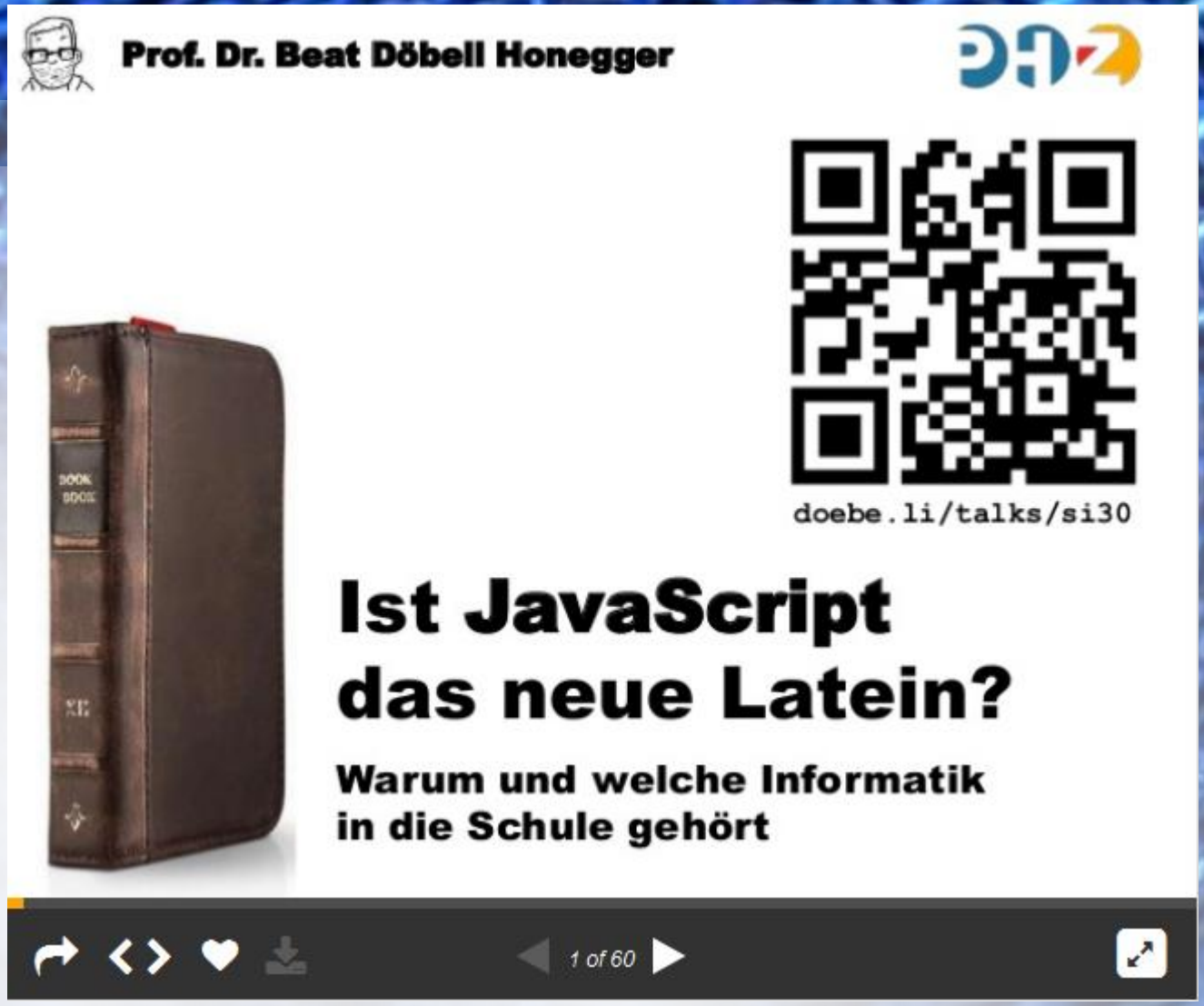


Empirische Untersuchungen im Wartezimmer eines Arztes haben ergeben, dass im Schnitt einer von zehn Patienten die eCard nicht mit hat. In einem Wartezimmer sitzen 10 Personen. **Modelliere** diese Situation in einer Tabellenkalkulationssoftware, in dem du geeignete Funktionen verwendest.

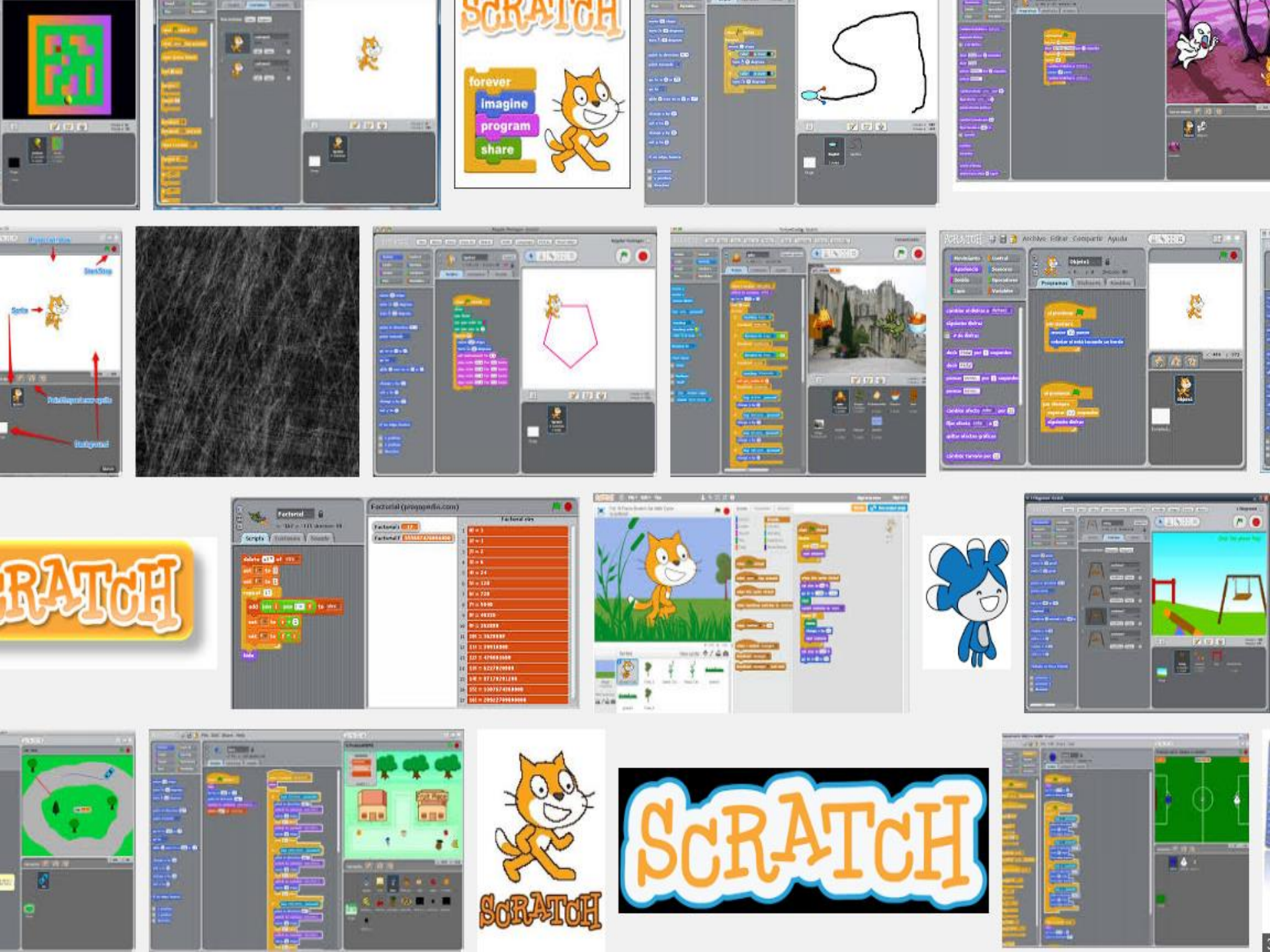
Führe diese Simulation angemessen oft (z.B. 1000 mal) **durch** und **visualisiere** die möglichen Szenarien (0, 1, 2, 3, ... haben die eCards mit) in Form eines aussagekräftigen Diagramms, das die Anzahl der „säumigen" Patienten von 0 bis 10 veranschaulicht. Dabei soll es auf die Sitzanordnung der Patienten im Wartezimmer nicht ankommen.

Ein Transferproblem: Ein Zahnarzt sieht den Zahnstatus in Form eines Bitmusters (z.B. 11111011100 ... 1110111) seiner Patienten auf dem Bildschirm. Dabei gibt es für jeden der 32 Zähne die Zustände 1 (für OK und unversehrt) und 0 (für jeden anderen Zustand). **Begründe**, warum es mit der zeilenweisen Darstellung aller möglichen verschiedenen 32-stelligen Bitmuster (aller möglichen Gebisse) in einer Tabellenkalkulation Probleme gibt bzw. geben könnte. **Finde** eine **Erklärung** dafür, was dies mit der IPv4-Adressierung im Internet zu tun hat.

Das moderne Gesundheitswesen wäre ohne die Segnungen der Digitaltechnologien und die Errungenschaften der Medizintechnik unvorstellbar. Dennoch gibt es im Zusammenhang mit der bereits seit langem etablierten eCard und der bevorstehenden Einführung von ELGA (Elektronische Gesundheitsakte) immer wieder (berechtigte?) Vorbehalte. **Kläre auf**, was es mit der eCard auf sich hat und **schätze** aus deiner Sicht die (vermeintlichen) Gefahren und die Sinnhaftigkeit von ELGA **ein.**



http://www.slideshare.net/beatdoebeli/ist-javascript-das-neue-latein-warum-und-welche-informatik-in-die-schule-gehrt


SCRATCH
forever
imagine
program
share
SCRATCH
SCRATCH
SCRATCH

RASPBERRY PI

EIN MATURABEISPIEL

Thema: VARIABLEN UND DATENSTRUKTUREN

Panini steht für ein italienisches Geschäftsmodell, das nicht zuletzt dadurch erfolgreich ist, dass es die archaische Sammelleidenschaft der Kunden anspricht und sozial in dem Sinne ist, dass es zum Tauschhandel anregt.
Der Quellcode links gibt nach Eingabe der Platzhalter eines Panini-Albums aus, wieviele Sticker gekauft werden müssen, um das Album zu komplettieren. Zufallsbedingt kann dies eine teure Angelegenheit werden.

```
import random
print("Wieviele Sticker?")
a=input()
l=[]
e=0
for i in range(1,int(a)+1):
    l.append(i)
while len(l)!=0:
    x=random.randint(1,int(a))
    if l.count(x)!=0:
        l.remove(x)
    e=e+1
print("Zu kaufen:")
print(e)
```

Implementiere diesen Code in einer Python-Entwicklungsumgebung, und schätze, wieviele Sticker für das Fußball-WM 2014-Album (640 Sticker) notwendig sind.
Erkläre den Aufbau und die Programm- und Datenstrukturen dieses Algorithmus und führe exemplarisch Python-Spezifika an.
Modifiziere das Programm so, dass es zufällige Stickernummern anzeigt, solange sie verschieden sind und erst dann stoppt, wenn das erste Duplikat erscheint.

Bewerte die Programmiersprache Python im Zusammenhang mit anderen dir bekannten formalen Sprachen und Entwicklungsumgebungen. Das Panini-Problem ist in der Literatur auch als Sammlerproblem bekannt. **Führe** ein weiteres Beispiel an.

Weitere Spielereien?

BLOCKLY (Neil Fraser, Google)

https://blockly-games.appspot.com/

https://developers.google.com/blockly/

ÖSTERREICH-PREMIERE PENCILCODE

Nette Sache, oder ein „more of the same“?

http://pencilcode.net