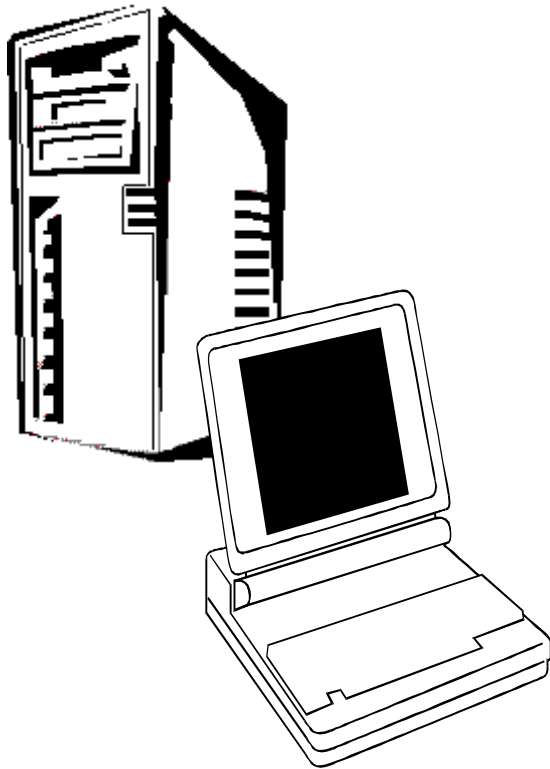
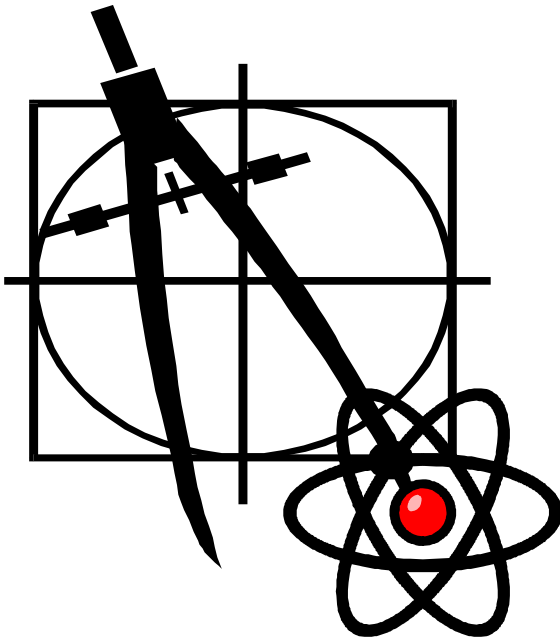


VIERTE DIMENSION

für Wissenschaft und Technik, für kommerzielle EDV,
für MSR-Technik, für den interessierten Hobbyisten.



In dieser Ausgabe:

Forth-Tagung 1998

Leserbriefe und Berichte von der Tagung und der Mitgliederversammlung 1998 in Neukirchen-Vluyn

Interessantes aus dem Netz

Auszüge aus DE\COMP\LANG\FORTH

“Embedded” - eine Einmannzeitschrift

Fred Behringer stellt eine lesenswerte Arbeit von Raffael Deliano vor.

Prelude - OO Konzept, ‘late binding’

Klaus Schleisiek stellt das Konzept von Manfred Mahlow und seine Implementierung vor.

Objekte in HolonForth

Ein ‘einfaches’ und effizientes Konzept, vorgestellt von Wolf Wejgaard

Objektorientiertes Programmieren in comFORTH 4

Der erste von drei Teilen eines Aufsatzes zu einem technischen Leckerbissen, von Egmont Woitzel

Dienstleistungen und Produkte fördernder Mitglieder des Vereins

FORTH - Shirt



Räumungsverkauf

T - Shirt: hellgrau / grün
in Größe M-L-XL **15 DM**
Sweat-Shirt: grau / grün
in Größe M-L-XL **25 DM**
(+ Porto)

ForthWORKS

Ulrike Schnitter
Nelkenstr. 52
85716 Unterschleißheim
fon/fax 089-310 33 85

Dipl.-Ing. Arndt Klingenberg

Tel.: ++32 +87 -63 09 89 (Fax: -63 09 88)
Waldring 23, B-4730 Hauset, Belgien
akg@aachen.forth-ev.de

Computergestützte Meßtechnik und Qualitätskontrolle, Fuzzy, Datalogger, Elektroakustik (HiFi), MusiCassette HighSpeedDuplicating, Tonband, (engl.) Dokumentationen und Bedienungsanleitungen

Forth Engineering Dr. Wolf Wejgaard

Tel.: +41 41 377 3774 - Fax: +41 41 377 4774
Neuhöflirain 10
CH-6045 Meggen <http://holonforth.com>

Wir konzentrieren uns auf Forschung und Weiterentwicklung des Forth-Prinzips und offerieren HolonForth, ein interaktives Forth Cross-Entwicklungssystem mit ungewöhnlichen Eigenschaften. HolonForth ist erhältlich für 80x86, 68HC11 und 68300 Zielprozessoren.

KIMA Echtzeitsysteme GmbH

Tel.: 02461/690-380
Fax: 02461/690-387 oder -100
Karl-Heinz-Beckurtz-Str. 13
52428 Jülich

Automatisierungstechnik: Fortgeschrittene Steuerungen für die Verfahrenstechnik, Schaltanlagenbau, Projektierung, Sensorik, Maschinenüberwachungen. Echtzeitrechnersysteme: für Werkzeug- und Sondermaschinen, Fuzzy Logic

FORTech Software GmbH Entwicklungsbüro Dr.-Ing. Egmont Woitzel

Joachim-Jungius-Straße 9 D-18059 Rostock
Tel.: (0381) 4059472 Fax.: (0381) 4059471
E-Mail: EWOI@FORTECH.DE

PC-basierte Forth-Entwicklungswerkzeuge comFORTH für Windows und eingebettete und verteilte Systeme. Softwareentwicklung für Windows und Mikrocontroller mit Forth, C/C++, Delphi und Basic. Entwicklung von Gerätetreibern und Kommunikationssoftware für Windows 3.1, Windows95 und Windows NT. Beratung zu Software-/Systementwurf.

Ingenieurbüro Dipl.-Ing. Wolfgang Allinger

Tel.: (+Fax) 0+212-66811
Brander Weg 6
D-42699 Solingen

Entwicklung von µC, HW+SW, Embedded Controller, Echtzeitsysteme 1-60 Computer, Forth+Assembler PC / 8031 / 80C166 / RTX 2000 / Z80 ... für extreme Einsatzbedingungen in Walzwerken, KKW, Medizin, Verkehr / >20 Jahre Erfahrung.

Ingenieurbüro Klaus Kohl

Tel.: 08233-30 524 Fax: —9971
Postfach 1173
D-86404 Mering

FORTH-Software (volksFORTH, KKFORTH und viele PD-Versionen). FORTH-Hardware (z.B. Super8) und -Literaturservice. Professionelle Entwicklung für Steuerungs- und Meßtechnik.

IMPRESSUM

Name der Zeitschrift

Vierte Dimension
Organ der Forth-Gesellschaft e.V.

Herausgeberin

Forth-Gesellschaft e.V.

Postfach 161204
D-18025 Rostock
Tel.: 0381-4007872

E-Mail:
SECRETARY@ADMIN.FORTH-EV.DE
Bankverbindung: Postbank Hamburg
BLZ 200 100 20
KTo 563 211 208

Redaktion & Layout

Friederich Prinz
Hombbergerstraße 335
47443 Moers
Tel.: 02841-58 3 98
E-Mail: F.PRINZ@MHB.GUN.DE
FRIEDERICH.PRINZ@T-ONLINE.DE

Anzeigenverwaltung

Büro der Herausgeberin

Redaktionsschluß 1997

März, Juni, September, Dezember
jeweils in der letzten Woche

Erscheinungsweise

1 Ausgabe / Quartal

Einzelpreis

DM 10,- zzgl. Porto u. Verp.

Manuskripte und Rechte

Berücksichtigt werden alle eingesandten Manuskripte. Leserbriefe können ohne Rücksprache gekürzt wiedergegeben werden. Für die mit dem Namen des Verfassers gekennzeichneten Beiträge übernimmt die Redaktion lediglich die presserechtliche Verantwortung. Die in diesem Magazin veröffentlichten Beiträge sind urheberrechtlich geschützt. Übersetzung, Vervielfältigung, Nachdruck sowie Speicherung auf beliebigen Medien ist auszugsweise nur mit genauer Quellenangabe erlaubt. Die eingereichten Beiträge müssen frei von Ansprüchen Dritter sein. Veröffentlichte Programme gehen - soweit nichts anderes vermerkt ist - in die Public Domain über. Für Fehler im Text, in Schaltbildern, Aufbauskißzen u.ä., die zum Nichtfunktionieren oder eventuellem Schadhafwerden von Bauelementen oder Geräten führen, kann keine Haftung übernommen werden. Sämtliche Veröffentlichungen erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.



Liebe Leser,

Die Forth-Tagung in diesem Jahr in Neukirchen-Vluyn, ausgerichtet von der Moerser Forthgruppe, war ein großer Erfolg. Die Teilnehmer haben sich, wie Sie den Leserbriefen zur Tagung entnehmen können, am Niederrhein ausgesprochen wohl gefühlt. Eine angenehme, entspannte Atmosphäre,

ausreichend Zeit zur Diskussion zwischen den Vorträgen, interessante Workshops und Besucher, gut vorbereitete Direktoren und eine reibungslose Organisation haben dazu beigetragen, daß ein Teilnehmer geäußert hat, er habe den Eindruck, wir Forther seien erwachsen geworden. Die Moerser Forther fassen das als großes Kompliment auf.

Im Zeichen der Tagung steht auch die vorliegende Ausgabe der VD, von der wir hoffen, daß sie ebenso erwachsen wird, wie es die Forth-Gesellschaft ist. Als 'Vereinszeitschrift' will die VD versuchen, den Mitgliedern, die nicht an der Tagung teilnehmen konnten, so viel wie möglich von ihren Inhalten zu vermitteln. Darum nimmt die Jahrestagung einen relativ breiten Raum in dieser Ausgabe ein.

Leserbriefe zur Tagung, der Bericht der Direktoren, das Projekt Forth(buch), Aufsätze zu einigen, ausgewählten Vorträgen der Tagung und Vieles mehr, sollen Ihnen das Geschehen in Neukirchen-Vluyn näher zu bringen. Gleichzeitig soll Ihnen das Appetit auf die Jahrestagung in 1999 machen, die Heinz Schnitter ausrichten wird. Tagungsort wird dann (vermutlich) Oberammergau sein.

“Unser Mann in Bonn”, Peter Enders, MdB (SPD), von den Menschen am Niederrhein mit einem Direktmandat ausgestattet, war ein interessanter und wichtiger Gast der Tagung. Eine glückliche Hand bei der Vortragsauswahl und der zeitlichen Zusammenstellung der Vorträge, hat Herrn Enders einem Vortrag beiwohnen lassen, der ihn auch persönlich sehr interessiert hat. Das macht ‘dem Forth’ und der Forth-Gesellschaft noch keine Lobby in Bonn, aber wir dürfen sicher sein, daß ‘man’ sich in Bonn zu verschiedenen Themen in der nächsten Zeit immer wieder an Forth und an eine kleine, aber feine Veranstaltung am Niederrhein erinnern wird.

Die “Embedded”, die ihm Herbst diesen Jahres stattfinden wird, wird erstmals eine eigene Messezeitung haben. Die Forth-Gesellschaft hat von den Organisatoren der Messe die Gelegenheit bekommen, einen Text in dieser Zeitung zu veröffentlichen. Selbstverständlich hat der Verein diese Gelegenheit wahrgenommen. Interessant ist hieran vor allem, daß der Text zur Messezeitung nicht als Einzelleistung eines interessierten Mitgliedes verfaßt wurde, sondern ebenfalls während der Tagung in einem ‘Workshop’ von den Teilnehmern gemeinsam verfaßt wurde. Um das Publikum der “Embedded” gezielt anzusprechen, haben sich die ‘Autoren’ des Textes jeweils entsprechend ihren Erfahrungen und ihrem speziellen, auf dieser Messe gefragtem Fachwissen in diese für den Verein wichtige Arbeit eingebracht.

Wir, die ‘Forther’, tun etwas miteinander und füreinander - füreinander und für Forth, dessen Förderung und Verbreitung unser satzungsgemäßes Ziel ist.

Für das “Editoriat”
Friederich Prinz



Quelltext Service

Die Quelltexte in der VD müssen Sie nicht abtippen. Sie können diese Texte auch direkt bei uns anfordern und sich zum Beispiel per E-Mail schicken lassen. Schreiben Sie dazu einfach eine E-Mail an die Redaktionsadresse.

fep



Leserbriefe

Die Forth-Tagung in Neukirchen-Vluyn hat den Teilnehmern ausnahmslos gefallen, so gut, daß sie sich auf die nächste Tagung freuen. Um allen Mitgliedern, die in diesem Jahr nicht an der Tagung teilgenommen haben, 'Appetit' zu machen, stellen wir Ihnen hier einige Leserbriefe und 'Lobesschreiben' vor...

Absender : All@business.forth-ev.de (**Wolfgang Allinger**)

War alles super, auch das Hotel hat mir gut gefallen. Und der Besuch des Azubi Bergwerkes war prima.

Tschüß Wolfgang

Zur Erklärung: Friederich Prinz hatte einer 14-köpfigen Gruppe angeboten, mit ihm das Lehrbergwerk (Ausbildungsstätte) der Ruhrkohle Bergbau AG zu 'befahren'. Aus Platzgründen mußte die Teilnehmerzahl hier beschränkt werden. Die Fahrt fand bereits am Freitag morgen statt. Die Mehrzahl der Teilnehmer mußte dazu einen Tag früher anreisen. Weiteren Interessenten an einem Besuch des Lehrbergwerks sei an dieser Stelle empfohlen, sich mit Friederich Prinz in Verbindung zu setzen.

Die Redaktion

Liebe Redaktion,

die diesjährige Tagung war wieder ein voller Erfolg. Dank der guten Organisation von Friederich Prinz verlief alles reibungslos. Die Unterkunft war ausgezeichnet, das Essen war gut, die Vorträge und vor allem die Mitgliederversammlung verliefen in sehr lockerer Atmosphäre.

Das Direktorium hatte bestens vorgearbeitet. Es gab keine Probleme, die nicht gelöst werden konnten. Der Tagungsort lag nicht direkt im Zentrum der Welt, aber das war vielleicht, vom ruhigen Arbeitsklima her gesehen, gar nicht so schlecht. Der kurze Abstecher nach Moers mit dem Einblick in die Arbeitsumgebung der Moerser Gruppe hat mir gefallen. Der Abholdienst von und nach Duisburg (Dank an Ulrich Richter) für die Teilnehmer, die mit der Bahn kamen, war eine glänzende Idee. Ein hohes Lob nochmal dem Fritz, dem Organisator, und seinem Moerser Team!

Fred Behringer, München

Absender : uho@informatik.uni-kiel.de (**Ulrich Hoffmann**)

Ich bin auch wieder heil nach Hause gekommen. Es war ein hervorragendes Treffen. Ich freue mich da gewesen zu sein. Vielen Dank an alle, die dafür gesorgt haben, insbesondere die Tagungsorganisatoren. Ich freu' mich schon auf das nächste Treffen.

Viele Gruesse,

Ulrich

Absender : michael@malente.forth-ev.de (**Michael Kalus**)

Hi, Friedrich

Hat prima geklappt! Möchte mich hier noch mal ganz herzlich bei allen aus eurer Gruppe bedanken, dafür das ihr das Treffen ausgerichtet habt. Es hat mir gut gefallen, wie ihr das gemacht habt.

Grüße aus Malente,

Michael

ABS: wejgaard@acm.org (**Wolf Wejgaard**)

Hallo Fritz,

es war eine feine Tagung, gelöst, interessant, ermutigend. Ich danke Dir nochmals für die gute Betreuung inklusive Transport von/nach Duisburg. Und auch vielen Dank an Deine Freunde und Kollegen in Moers.

Leider waren meine Beiträge zur Tagung etwas jenseits der Klarheit. Ich war permanent zu müde, um meine Gedanken zu sammeln. Jetzt tauche ich langsam wieder hoch.

Ich hoffe, dass Du mit einem sehr guten Gefühl im Bauch an das Wochenende zurückdenkst. Immerhin hat Dir die Tagung auch Gelegenheit zu einem netten "MdB-Spiel" gegeben, dass sich nun in mehr Rückenwind für Euer feines MALZ auswirken möge! Und Dein "Gutes-Forth" Projekt hast Du ja auch zum Laufen gebracht! Hier wirkt sich wohl auch die neue Lage und gute Stimmung aus, die Ihr, das Direktorium, lobenswerterweise in die FG gebracht habt.

Möge es so weiter gehen.

Gruss, Wolf

Das 'gute Gefühl' war in den ersten Tagen eine große Müdigkeit. 'Beruflich' bin ich an recht frühes Aufstehen gewöhnt, und mit fortschreitendem Alter auch entsprechend frühes zu Bett gehen. Letzteres war während der Tagung natürlich nicht möglich.

Von dem Besuch von Peter Enders, MdB (SPD) auf unserer Tagung erhoffe ich mir natürlich nicht nur 'Rückenwind' für das Moerser Arbeitslosenzentrum, sondern auch für Forth und die Forth-Gesellschaft. Welcher 'Computer Club' hat denn schon eine Lobby in Bonn ? :-)

fep

Absender : EWoi@fortech.de (**Egmont Woitzel**)

Die Rostocker schließen sich hiermit den Dankesgrüßen an. Durch den optionalen Tag 'vor' der Tagung für die Bergwerksbesichtigung und die reichliche Zeit für die einzelnen Vorträge war das die entspannteste (und erholsamste) Forth-Tagung, an der ich bisher teilgenommen habe.

Viele Grüße

Egmont Woitzel

Absender : kschleisiek@send.de (**Klaus Schleisiek**)

Hallo Fritz,

ich fand es ganz toll bei der Tagung, danke für Eure Mühe. Und auch für MALZ: Gratulation. Es tut gut, ein Beispiel lebendiger Demokratie zu sehen.

Von Eduard (unserer Stoffmaus) soll ich Euch herzlich grüßen. Seit er das Bilderbuch mit SWAPs Erlebnissen gesehen hat, war er in den Ferien nicht mehr zuhause zu halten und durfte sogar mit in den Grand Canyon...

Klaus

Sind Sie neugierig geworden - auf das Bilderbuch von SWAP? ... auf die tolle Stimmung während der Tagung ? Es ist 'noch etwas hin', aber merken Sie sich trotzdem den April 1999 vor !

fep

<u>Leserbriefe</u>	4, 6
<u>Interessantes aus dem Netz</u>	12
Typ-Prüfungen in Forth, Ein modernes Forth (<i>Claus Vogt</i>), Diverses	
<u>“Embedded” - Besprechung einer Einmannzeitschrift</u>	15
Vorstellung einer lesenswerten Arbeit, <i>Fred Behringer</i>	
<u>Prelude - Objekte und Methoden mit Fehlerprüfung</u>	16
OO und ‘late binding’ nach Manfred Mahlow, <i>Klaus Schleisiek</i>	
<u>Objekte in HolonForth</u>	19
Das Objektkonzept von Holon, <i>Wolf Wejgaard</i>	
<u>Objektorientiertes Programmieren in comFORTH 4, Teil - 1 -</u>	22
Wie comFORTH 4 welche Objekte zur Verfügung stellt, zeigt <i>Egmont Woitzel</i>	
<u>FDUMP - Nachtrag</u>	28
Korrekturen zum Beitrag aus der VD 1/98, <i>Fred Behringer</i>	
<u>Gehaltvolles</u>	30
...aus den USA, Großbritannien und den Niederlanden, <i>Fred Behringer</i>	
<u>Bericht des Direktoriums</u>	33

In der nächsten Ausgabe finden Sie voraussichtlich:

CRT für DUMMIES - Wolfgang Allinger beschreibt ausführlich, wie er den Kampf um seine serielle Schnittstelle gewonnen hat.

Objektorientiertes Programmieren in comFORTH 4, Teil - 2 -, die Fortsetzung des Aufsatzes von Egmont Woitzel, plus Sourcen

Ihre Programmierbeispiele zum Zählen und Sortieren von Äpfeln...



Leserbriefe

Nachfolgende **Meldung** hat die Redaktion am **1. April 1998** erreicht. Mangels entsprechender Detailkenntnis gehen wir trotzdem davon aus, daß ihr Inhalt ernst gemeint ist... ;-))

Kurzmeldung zu MARC4:

Nachdem die TEMIC an Atmel verkauft wurde und diese Firma bereits ein reichliches Angebot an Controllern hat, war die Zukunft des MARC4 etwas ungewiß. Die Entscheidung scheint inzwischen gefallen zu sein, daß alle vorhandenen Produkte für die nächsten Jahre verfügbar bleiben. Der Einsatz des MARC4 wird in Zukunft jedoch auf Standardzellen ausgerichtet, in denen er zusammen mit passender Peripherie für bestimmte Anwendungen integriert wird. Betreuung des MARC4 erfolgt von Eching aus, wo insbesondere auch der Bereich Transponder angesiedelt ist.

R. Deliano

Kurzmeldung zu MPE

Produkte: MPE stapelt PowerNet auf IP

Eine oft gesuchte, aber selten gefundene Lösung bietet MPE. Warum eigentlich nicht die verteilten kleinen Systeme per IP an den Steuer-PC koppeln? Für alle vom MPE-Targetcompiler unterstützten Systeme gibt es jetzt IP und den passenden Paket-Treiber UDP unter dem Namen PowerNet. Auch ICMP wird unterstützt, sodaß der Test mit PING-Programm gleich losgehen kann. Zur Übertragung kommen Ethernet und SLIP in Frage. Vom Forth aus wird KEY und EMIT zur Übertragung benutzt. Das Paket umfaßt die kompletten Quellen, Dokumentation und Terminalemulatoren für Win95 bis NT 5.0.

Weitere Informationen bei MPE - MicroProcessorEngineering Ltd; 133 Hill Lane; Southampton SO15 5AF UK; mpe@mpeltd.demon.co.uk

clv, 3/98, nach Firmeninformationen

Claus Vogt

Nachtrag zum Artikel "Real-Mode-32-Bit-Erweiterung für Turbo-Forth" von **Fred Behringer** aus der VD 2/98

Im Druck sind leider ein paar Zeilen verlorengegangen. Nicht weiter schlimm, da diese lediglich das System schon beim Einladen des Zusatzprogramms auf die erweiterte Segmentgrenze umschalten und eine Sicherheitsabfrage über den Zustand der A20-Leitung durchführen.

CR .(32-Bit-System einschalten)

SIZEUP 0 FS! \ Segmentgrenzen hochschalten. Vorgabe für \ FS =3D 0

```
:XXX ( -- ) FREE-A20 CR      \ A20 freischalten
  ." Adreßleitung A20" A20?  \ Wirklich frei ?
  IF ." jetzt frei."
  ELSE ." noch blockiert."
  THEN ;
```

XXX \ XXX aus=FChren

FORGET XXX \ und weg

ABS: MP7@wanadoo.fr (**Marc PETREMANN**)

Ich habe mein selbst web site, wo mann can Informationen über FORTH finden.

Adresse:

<http://ourworld.compuserve.com/homepages/mp7>

In Forth Rübrik, werden Sie finden etwas über der Jupiter ACE:

- Jupiter ACE emulator für DOS,
- mehre Programmen für Jupiter ACE.

Und meine web site ist sehr "sexy".

Salutations.

*Ein Schelm, wer Böses dabei denkt - diese kleine Notiz aus dem Netz haben wir **nicht wegen des letzten Satzes** von Marc Petremann übernommen ;-)*



Gruesse!

Nun da dieser freundliche Professor aus Mittweida sich der Mühe unterzieht meine 'Berichte' zu uebersetzen, denke ich, daß ich etwas mehr Aufmerksamkeit auf die Qualität anstatt die Quantität meiner Briefe richten sollte. Das macht es etwas mehr zu meinem Job, da ich, wie ich ich schon bemerkte, seit über einem Monat seit Ankunft meiner VD gezögert habe. Daher denke ich, ist es am Besten, ich setze mich gleich hin und mache mein WYGIWYWW für euch ohne darüber zu klagen. ("What You Get Is What You Work With" - Was Du bekommst, ist daß womit Du arbeitest. Wie z.B. Windows, VCR's, Leihwagen und andere Beispiele der "Hi-Tech-Nutzerfreundlichkeit".)

Auf dem Januar-Treffen der SVFIG hatten wir vier Teilnehmer namens John und etwa noch 16 andere, die im Laufe des Tages kamen. Es gibt mindestens noch vier weitere "John's", die aber nicht da waren. Gibt Euch das eine Vorstellung wie viele Leute zur Silicon-Valley Forth-Interest-Group gehören? Auf der Morgensitzung, die von John Carpenter geleitet wurde, versuchten wir die Anforderungen an Forth für das 21. Jahrhundert herauszufinden, indem wir nach Antworten für solche Fragen wie "Was ist gut und was ist schlecht an Forth? Wer sind seine Nutzer und was wollen sie in Forth finden?" suchten. Wenn Ihr die volle Liste sehen wollt (etwa eine Seite voll), schickt eine Nachricht an John (jdc@leland.stanford.edu).

Falls Ihr findet, daß Ihr das alles schon kennt und nach besseren Antworten sucht, dann probiert doch noch **PoncedeLeon@Bimini.com**. Ich sehe, daß ich ein paar Notizen über einige Kommentare gemacht habe, die mir interessant erschienen: "Wandle Forth von einer Sprache zu einem Befehlssatz für einen Superchip um"; "Bringt Forth auf den USB (Universal Serial Bus)"; "Software zu schreiben wird immer teurer"; "Festplatten kosten heute nur noch 4 Cents pro MB."

Am Nachmittag beschrieb Dwight Elvey seinen ADSP2100



Forth Compiler, den er für ein preiswertes Analog Devices Board entwickelt. John Rible gab einen Bericht über ein Buch, welches mit einem Satz von Nitinol-Drähten zusammen kommt, mit denen man seinen eigenen winzigen, laufenden Roboter bauen kann, der bis zu 3 cm/min schafft. (Nitinol ist ein Akronym aus Nickel, Titan und Naval Ordnance Laboratory.)

Nun, einen Monat später, finden wir uns auf dem Februar-Treffen der SVFIG wieder. Californien hat inzwischen eine Menge Regen gesehen. Einige Häuser sind die Hügel hinunter gerutscht, einige sind sogar in den Pazifik gefallen. Unsere entfernt lebenden Verwandten rufen an und machen sich Sorgen, ob wir noch auf trockenem Land sind. Oh ja, das ist so wie bei den Erdbeben: Der Prozentsatz der davon betroffenen Leute ist ziemlich klein. Wir bedauern die, die ihr Heim verloren haben und wir sind verärgert, wenn einige Straßen geschlossen sind und es noch länger dauert zur Arbeit oder zurück zu fahren. Aber Mutter Natur muß wohl zeigen, daß sie stärker ist als die City Hall (das Rathaus???)

Somit haben wir mit etwa einem Dutzend Forthern am Morgen begonnen, aber als dann die Sonne auf uns herunter schien, zählte ich wieder mindestens 20. Der unermüdliche Dr. Ting füllte die Morgensitzung mit einer detaillierten Beschreibung der Internals des 8051 und des kommenden 8051 eForth Version 2.01. Subroutine Threading ist wieder aktuell, obwohl, wie sich John Rible erinnerte, vor 20 Jahren subroutine-threaded Forth KEIN Forth war, nach Ansicht der Gründungsväter. Dr. Ting denkt, daß eForth eventuell sogar ähnlich wie ein Basic Stamp nutzbar ist, falls es in 4 kB des internen 8751-Speichers gepresst werden kann (Wartet auf den nächsten Bericht!)

Am Nachmittag sprach John Rible ausführlicher über eForth's Datenstrukturen und über Multi-Tasking in eForth. Er erinnerte uns auch daran, daß der ANSI Forth-Standard kurz vor der fünfjährigen Ratifizierung steht. Es gibt Vorschläge was zum Standard hinzugefügt werden sollte, z.B. die Benennung von Gruppen von Annahmen (z.B. eine Adresse ist ein CELL, Returnstack ist ein CELL, Fädellung läuft über den R-Stack usw.). Wir stimmen wohl überein, daß der Standard eine bessere Beschreibung des eigenen Systems erlaubt, indem man aufzeigt wie das eigene System vom Standard abweicht.

Zum Schluß bekam Bill Ragsdale, einer der FIG Gründer, etwas Zeit (allerdings nicht genug) um über einige Dinge

von allgemeinerem Interesse zu sprechen, wie z.B. das Jahr-2000 Problem und finanzielle Analyse und Vorhersage im Aktienmarkt. Kalman Filter und Chaos Theorie für den Aktienmarkt? Wow! Ich hoffe das Bill bald wieder kommt. Ich war froh ein wenig Geschichte der FIG von ihm zu erfahren. Sie ist jetzt um die 20 Jahre alt. Bill führte Forth in den Homebrew Club (Selbstbau-Klub) ein, der in Menlo Parks, Californien, 1975 startete (Bob Reiling, der Editor der Klubzeitschrift war, ist der Organisator der FORML Konferenzen und regelmäßiger Teilnehmer der SVFIG-Treffen). Wie Ihr wißt, nutzte Chuck Moore sein Forth für das Radioteleskop auf dem Kitt Peak Observatorium. Der Rest der Welt hat es von da übernommen. Es gab eine Gruppe in der Schweiz am Genfer Observatorium (siehe auch den Artikel von Paul Bertholdi in Forth Dimensions, Dec.78 / Jan.79) die ebenfalls Forth förderten. Es gab zwei FORML Konferenzen in 1979, die erste in London. Daher haben wir dieses Jahr die 20. Konferenz und werden zu Recht FIG's 20. Geburtstag feiern. Die ersten Forth Dimensions kamen im Juni 1978 heraus, dank Mr. Ragsdale's Schreibmaschine. Ihr könnt die Namen der 'Gründungsväter' am unteren Ende von Seite 2 dieser Ausgabe sehen.

Leider fliegt die Zeit, und einige der Old-Timer sind nicht länger unter uns. Erinnern wir uns an Gary Kildall, den Erfinder von CP/M? Ich habe gerade das Buch "Programmers at Work" von Susan Lammers geöffnet und lese was Kildall über Forth und LISP zu sagen hatte. Da ist auch noch Jonathan Sachs der über sein STOIC spricht, eine Variante von Forth. Das ist ein sehr interessantes Buch, Interviews mit 19 Programmierern, noch vor 1986. Aber, ich schweife ab... , das ist kein Bericht über Bücher.

Aber da sind noch ein paar mehr davon, die ich für Wert fand zu lesen. Das nächste SVFIG-Treffen ist nur noch eine Woche entfernt. Hoffentlich bin ich dort und ich werde versuchen einen KURZEN Bericht direkt danach zu schreiben. Mit den besten Wünschen,

Henry

Guten Tag!

Ich war etwas faul. Es ist nun schon fast zwei Monate seit dem März-Treffen der SVFIG. Schauen wir mal, ob meine Notizen noch Sinn machen, da mein Gedächtnis nicht so leicht aufzufrischen ist...

Wir waren die üblichen 20 bis 25 Leute auf dem Treffen am 28. März. Dr. Ting war nicht in der Lage gewesen, daß

Date: Thu, 26 Feb 1998 11:05:10 +0100

From: Fred Behringer

<behringe@mathematik.tu-muenchen.de>

...

der Alan Wenham hat mich auf einen Fehler in meinem Zen-Floating-Point-Artikel aus der VD 98/1 aufmerksam gemacht. Könnten wir die folgende Korrekturmeldung in die VD 98/2 setzen?

...aber selbstverständlich ! Dafür sind wir doch da ;-)

Korrektur zu Zen Floating Point aus VD 98/1

die Redaktion



eForth soweit zu reduzieren, daß es in die 4 kB des 8751 Chips paßt, aber er gab uns eine zusammengefaßte Erläuterung über seine heldenhaften Bemühungen es doch zu schaffen. Es gab zu viele Schwierigkeiten mit @ (und !) im Suchmechanismus. EFORTH belegt gegenwärtig 6 bis 7 kB, mit 203 Woertern. Der 68HC11 wird es jedoch schaffen, da er zusätzlichen RAM bereitstellt. Anregungen aus dem Auditorium betrafen die Nutzung serieller EPROMs, die Abtrennung und Verlagerung des Compilers in den Terminal Emulator und weitere Ideen, die über mein Verständnis hinaus gingen.

Dr. Ting ist es gelungen, das 8086 eForth auf etwa 3,9 MB (???) zu reduzieren - er nennt es ZF fuer ZenForth. Das Zen schreitet voran.

Unser FIG Präsident, Skip Carter, hielt die zweite Präsentation des Tages. Er demonstrierte ein diskloses Linux-System, welches ein Boot-ROM nutzt, das auf der Netzwerkkarte installiert ist. In Skip's eigenen Worten: "Dieses ROM bringt das System dazu, auf dem lokalen Netzwerk nach einem Boot-Server zu suchen. Dieser Boot-Server liefert dann das Abbild des Betriebssystems und das disklose System startet Linux unter Nutzung der vom Server exportierten Laufwerke." Warum hat Skip das gemacht? Er war beim Aufrüsten seines Motherboards und "Warum packe ich das alte nicht in eine 30\$ Box, besorge eine 3-COM 3C509 Netzwerkkarte und lasse Linux Software laufen, die dann auch wirklich läuft?" (Das alte Motherboard war ein 80486 66 MHz.)

Skip erwähnte weiterhin daß die FIG Mitgliederzahl momentan knapp unter 1000 beträgt. Die meisten Newcomer sind Ingenieure und es gibt noch eine Reihe russischer Emigranten.

John Hall übernahm das Podium um Skip mit dem prestigehafteten Figgy-Preis fuer 1997 auszuzeichnen. Er erinnerte außerdem alle daran, daß die nächste FORML Konferenz die 20. sein wird.

Meine Notizen für das nächste SVFIG Treffen (25. April) sind noch kürzer -- hauptsächlich, da ich 'Multi-Tasking' machen mußte, d.h. zuhören und außerdem meine Kopie der VD lesen, die am 15. April angekommen war. Es ist interessant, daß genau wie Friederich's Aufruf für den Bau "eines guten Forth", ein Aufruf von Chris Passauer zur Entwicklung eines Forth-basierten Betriebssystems auf dem Treffen vorgetragen wurde. Darüber aber später. "Ja zu tun gäbe es für alle reichlich auch auf dieser Seite des großen Teiches." Es ist gut zu hören, daß der Enthusiasmus für Forth weiterlebt.

So long,

Henry

Hallo, Friederich!

Ich sagte Dir, das ich dieses Mal pünktlich sein werde. Das letzte Treffen war vor vier Tagen, während eines Drei-Tage Wochenendes wenn die meisten Leute aus der Stadt fortfahren, aber ungeachtet dessen waren wir doch einige Leute. Ich habe 27 um die Mittagszeit gezählt. Am Morgen allerdings waren wir nicht sicher, ob wir das Treffen im Freien abhalten müssen, da niemand mit einem Schlüssel da war. Die Studenten wollen während der Ferien ihre Zeit nicht in der Schule verbringen.

Peter Milford präsentierte den ersten Vortrag auf der Veranda außerhalb des College Gebäudes. Er gab uns einige gute Tips was man von der Entwicklung neuer Betriebssysteme erwarten kann, speziell aufbauend auf seinen Erfahrungen mit BEOS, ein Betriebssystem der Firma "Be", die soweit ich es verstanden habe, von einigen früheren Apple-Mitarbeitern gegründet wurde. Peter hat auch noch einige Zeit in sein eigenes Geisteskind investiert, eine Sprache namens MCO. Diese basiert auf STOIC, welches ähnlich wie Forth aber mit zusätzlichen und veränderten Worten ist. Es gibt eine Menge Erfindungen, aber wann ist der Markt groß genug für eine Vermarktung?

Als dann der College Vertreter mit dem Schlüssel kam, war Chris Passauer bereit, sein neues Betriebssystem anzupreisen, welches in Forth geschrieben sein sollte. Er ist so davon begeistert, daß er weit bis nach Mittag redete und schließlich um 12:30 Uhr vor den grollenden Mägen kapitulieren mußte. Dieses Betriebssystem wird in Forth geschrieben sein, aber wird sich nicht mit "OK" melden. Es wird eine Target Applikation sein, die einen Datenfluß umgibt, um HTML, Java und was man noch so hat auszuführen. Es soll 100 % in Forth-Stil erweiterbar sein. Man soll in der Lage sein, alle Module die man braucht zu laden. Falls ich es richtig abgeschrieben habe, könnt ihr mehr darüber unter WWW.WENET/~LGICWURS erfahren.

Nach dem Essen berichtete John Rible über einige Neuigkeiten über den Picaro Chip, der in Circuit Cellar's 98er April Ausgabe beschrieben ist.

Mitchell kam mit einer Anzahl von Proben und Demos seiner Arbeitsweise. Er hat einige wirklich interessante und praktische Anwendungen die kommerzielles Interesse gefunden haben. Schaut ihn Euch unter WWW.AMRESEARCH.COM an.

Natürlich war Dr. Ting hier -- mit einem "kurzen Gespräch über ein wirklich kleines Forth". Er hat es geschafft, einen Forth-Interpreter auf einem 8051 Chip laufen zu lassen. Durch Verwendung von Zwei-Byte Tokens -- momentan um die 170 -- und der Eliminierung des Linkfeldes, ist er der Realisierung einer Forth-Briefmarke (s. BASIC-STAMP) näher gekommen.

Ich mußte die Tagung verlassen bevor John Rible seine Story über "Squeak" beendete -- ein Smalltalk in Smalltalk geschrieben, wobei der Smalltalk Kernel in C übersetzt ist. Ähnlich wie Fifth, verfügt Smalltalk über on-the-fly Neukompilation. Schaut es Euch unter WWW.RESEARCH.APPLE.COM/RESEARCH/PROJ./LEARNING-CONCEPTS/SQUEAK an.

Ich hoffe, daß auf der anderen Seite des großen Teiches alles in Ordnung ist. Und ich will Euch nicht einen guten Sommer wünschen, da es um die Zeit, da ihr dies lest wohl schon Herbst oder Winter ist...

Alles Gute, anyway!

Henry

Übersetzung: Thomas Beierlein


Kontakte zu Forthern 'irgendwo' in der weiten Welt, haben viele von uns - wie Henry Vinert's Berichte aus der FIG Kalifornien ebenso deutlich zeigen, wie Fred Behringer's Zusammenfassungen z.B. der ForthWrite und des VIJGEBLAADJE. Beiden sei an dieser Stelle dafür gedankt, daß Sie uns schon eine ganze Weile an ihren Erfahrungen und ihren Informationen teilhaben lassen. Beide gehören zu den fleißigsten 'Arbeitern' für die VD...

Internationale Kontakte zu unseren 'Kollegen in Forth' zu knüpfen und, wo bereits vorhanden, zu intensivieren, ist eine der Aufgaben, die sich das Direktorium der FG gemeinsam mit Teilnehmern der Jahrestagung '98 gegeben hat. Wir können von Forthern 'irgendwo in der Welt' lernen, Ideen sammeln, Anregungen bekommen - oder einfach nur über aktuelle Themen auf dem Laufenden gehalten werden. Dazu bedarf es gar nicht so sehr viel. Wie immer braucht es nur Jemanden, der sich die Arbeit macht - in diesem Fall eine Arbeit, die sicher eine sehr umfangreiche Korrespondenz bedeutet.

Diese Arbeit hat für die FG Fred Behringer übernommen, der seine persönlichen Kontakte der Gesellschaft sozusagen zur Verfügung stellt.

Herrn Behringers Bemühungen sind dabei auf ausgesprochen fruchtbaren Boden gefallen, wie die nachfolgende Übersetzung des Editorials der jüngsten Forth Dimension zeigt. Mr. Marlin Ouverson weist darin auf die Vielzahl der unterschiedlichsten Forth-Aktivitäten hin, die sich weltweit beobachten lassen - wenn man nur will. Seinem Votum läßt sich weder etwas hinzufügen, noch läßt es sich schöner artikulieren.

Und auch die Kollegen aus dem Vereinigten Königreich wünschen sich einen verstärkten Austausch mit uns und mit allen anderen Forthern im 'globalen Dorf'. Mr. Alan Wenham hat dazu für die ForthWrite den Part übernommen, die jeweils jüngste Ausgabe der VD in der ForthWrite inhaltlich kurz vorzustellen.



EDITORIAL *

Rund um die Welt

Daß Forth ein internationales Phänomen ist, wird von niemandem bestritten. Größere Tagungen wurden in verschiedenen Städten Europas und in Australien abgehalten und vor einigen Jahren besuchte eine Mannschaft von Technikern und Ingenieuren Orte in China, wo man sich mit Forth beschäftigt. Es gibt eine Forth-Webseite in französischer Sprache und ein interessantes DFÜ-Experimentierlabor in Brasilien, über das man mit seinem Web-Browser eigene Programme auf einem dortigen 8051-Microcontroller ausführen lassen kann. Es arbeitet mit Camel-Forth von Brad Rodriguez, einem Kanadier, und ist nicht etwa nur eine auf das Netz gestützte Simulation. Ein beträchtlicher Schatz an Forth-Informationen steht in Deutschland zur Verfügung, darunter eine inhaltsreiche Club-Zeitschrift, Ausdruck langjähriger Forth-Anstrengungen von Experten dieses Landes. Großbritannien ist Sitz mehrerer Forth-Firmen und vieler Forth-Anwender. Dort gibt es eine ebenfalls bewundernswerte Zeitschrift, *Forthwrite*, von deren Autoren einige auch in unserer eigenen Zeitschrift publizieren.

Natürlich kann dieser flüchtige Blick nicht jeden unserer geschätzten Leser und Autoren erfassen, solche aus Holland, Österreich, der Schweiz, Mexiko, Japan und anderen Ländern, in denen Forth Fuß gefaßt hat. Wir grüßen sie alle und heißen ihre ständige Teilnahme an der internationalen Forth-Gemeinschaft willkommen!

Wir können nicht Forth und jene Forth-Anwender ignorieren, die zufällig in anderen Regionen wohnen, durch geographische oder politische Grenzen von uns getrennt. Da gibt es solche, die aufregende Forth-Forschung betreiben (wenn auch manchmal in einer Sprache, die wir nicht so gut lesen können) und solche, die uns daran erinnern, wie man Forth für Anfänger schreibt. Und wiederum andere entdecken immer wieder von neuem die Stärke der Sprache Forth in Umgebungen mit beschränkten Mitteln, etwas, auf das wir, die wir unseren Lebensunterhalt mit Forth verdienen (oder verdienen wollen), auch größeren Wert legen sollten.

Von der mehr persönlichen Seite her freuen wir uns natürlich immer wieder, Gemeinsamkeiten mit jenen zu finden, die anderen Lebensgewohnheiten und Kulturen verhaftet sind. Wir schöpfen Stärke aus unserer Verschiedenheit und achten das Anderssein des anderen.

Forth Dimensions ruft alle Forth-Anwender auf, schriftliche Beiträge einzureichen, ganz besonders würden wir es aber begrüßen, von unseren internationalen Lesern zu hören. Auch die diesjährige FORML-Tagung (siehe hintere Umschlagseite) würde sich über mehr internationale Teilnehmer freuen. Es ist der zwanzigste Geburtstag unserer Tagung. Kommen Sie also nicht nur zu den Fachveranstaltungen, sondern auch, um mit uns zu feiern!

---Marlin Ouverson

* Aus der *Forth Dimensions* vom März/April 1998 übertragen von Fred Behringer

Die Welt wird größer...

Sich einander vorzustellen, in persona, in den jeweiligen Ansichten und Anschauungen und - nicht zuletzt - natürlich auch im 'forthigen' Schaffen, macht die Welt im positiven Sinne ein wenig kleiner - und für den gelegentlich doch "isolierten" Forther um Vieles größer. Man muß nur von einander wissen...

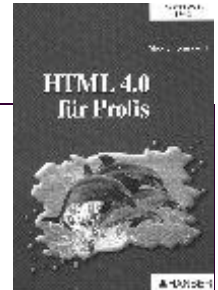
Dieses Wissen von einander zu fördern, soll ein Anzeigentausch forcieren, der zwischen der VD, der Forth Dimensions und der ForthWrite - bzw. den jeweiligen Gesellschaften und Chaptern - vereinbart wurde. Darüber hinaus sollen an dieser Stelle die Empfehlung und die Bitte stehen, zu überlegen, ob zukünftige Aufsätze und Beiträge zur VD nicht gleich in die englische Sprache übersetzt werden können. Das Interesse an diesen Arbeiten ist groß !

Das Interesse an den deutschen Forthern und ihrem Schaffen ist sogar so groß, daß **Glen Haydon** sich entschlossen hat, Mitglied der FG zu werden, bevor er eine entsprechende Anzeige in der Forth Dimensions lesen konnte. Dr. Glen B. Haydon ist den meisten Forthern Deutschlands sicher bekannt als der Schöpfer von MVP-Forth. Daß Dr. Haydon zudem Patho-

loge ist und Inhaber des Verlages Mountain View Press, ist sicher nicht allen Forthern auf dieser Seite des großen Teiches bekannt. Von Henry Vinerts wissen wir, daß Glen Haydon zu den versiertesten und aktivsten Forthern in den USA gehört. Willkommen, Mr. Haydon !

fep

Stephan Lamprecht
HTML 4.0 für Profis
316 Seiten, broschiert, Hanser
DM 59,- / ÖS 431,- / SFr 54,80
ISBN 3-446-19247-6



Alle reden von HTML (Hypertext Markup Language), kaum einer zeigt wie es geht ! Was verbirgt sich da eigentlich hinter meiner schönen Webseite, was steht im Quelltext und wie funktioniert es ?

Stephan Lamprecht zeigt in "HTML für Profis" wie es geht

und was sich hinter den Kulissen so verbirgt. Der Einsteiger und vor allem die zahlreichen ungeduldigen Quereinsteiger, erhalten mit dem Buch eine sachliche und nüchterne Referenz, die sich auf das Wesentliche konzentriert. Dazu gehören die Grundlagen zur Erstellung von Texten, Tabellen, Listen und interaktiven Formularen sowie Hypertext, die Vernetzung von Dokumenten oder das Einbetten von Multimedia Dateien. Was ist neu in der Welt der Tags ? Beschreibt die Abteilung Style Sheets und Layer sowie Ausblicke auf neue Technologien wie XML, Dynamic HTML und ActiveX. Mit der CD im Buch (enthält alle aufgeführten Listings sowie Editoren, Grafiktools und Referenzen) wird diese Referenz zu einer empfehlenswerten Grundlagen Lektüre, nicht nur für den Einsteiger.

mkm

Forth Interest Group International (FIG USA)

Wollen Sie mit der ganzen Welt verbunden sein und dabei Ihr Englisch perfektionieren? Amerika ist ein wesentlicher Teil der ganzen Welt. Zumindest, was Forth betrifft. Über tausend Mitglieder aus allen Ländern sind bei uns.

Werden auch Sie Mitglied in der Amerikanischen Forth-Gesellschaft (FIG USA).

Für 45 Dollar im Jahr (DM 72,-) bekommen Sie 6 Hefte unserer Vereinszeitschrift Forth Dimensions und genießen auch sonst verschiedene Vorteile. In den Heften erfahren Sie Forth-Neuigkeiten aus aller Welt, neue Produkte, Literatur, Forth-Ideen, fundiertes Wissen, Artikel auch für Einsteiger, Projekte, Leser-Diskussionen, Quelltexte, Hinweise auf Internet-Verbindungen, kostenlose Forth-Systeme und vieles mehr. (Für Übersee-Porto müssen wir leider noch 15 Dollar (DM 24,-) hinzurechnen.)

Unmittelbare Informationen über uns bekommen Sie, wenn Sie auf der Homepage der Deutschen Forth-Gesellschaft "Links zu anderen Forth-Organisationen" und dann "Forth Interest Group (USA)" anklicken.

Ansonsten bekommen Sie Auskünfte über das amerikanische Forth-Büro:

Forth Interest Group
100 Dolores Street, suite 183
Carmel, California 93923
USA

E-Mail: office@forth.org

oder auch vom Redakteur, *Marlin Ouverson*, unter der E-Mail-Adresse editor@forth.org



FORTH, Inc. bietet das

Forth Programmers's Handbook

an. Geschrieben wurde es von Edward K. Conclin und Elizabeth D. Rather. Zu beziehen ist das Buch über: FORTH, Inc.; 111 N. Sepulveda Blvd., Suite 300, Manhattan Beach, CA 90266-6847 USA. \$ 49,95 plus \$ 20,- für den Überseeversand soll es kosten. Die Beschreibung klingt so interessant, daß ich es mir im Herbst kommen lassen will, allerdings per Sammelbestellung, um die Frachtkosten zu drücken. Bei Interesse, bitte bei der Redaktion melden.

fep

FIG UK

Join our group of "Forthers"
Get access to our extensive library
Receive our bi-monthly magazine
(back issues available)
Access our web site
(www.users.zetnet.co.uk/aborigine/forth.htm)
Get free joining pack

Membership is £10
for which you get 6 magazines

Contact:
Dr.Douglas Neale
58 Woodland Way, Morden Surrey, UK, SM4 4DS
Voice: (44) 181-542-2747
Email: dneale@w58wmorden.demon.co.uk

Bericht über die Sitzung des Drachenrates auf der Jahrestagung 1998 in Neukirchen-Vluyn in der Nacht zum 26. April 1998

Wie in jedem Jahr, so tagte auch 1998 wieder der Drachenrat in den späten Stunden der Nacht vor der Mitgliederversammlung, um einen neuen Würdenträger für den nun im zweiten Jahr in Bronze gegossenen Swap-Dra-chen und gleichzeitig ein weiteres Mitglied des Drachenrats zu finden. Um die Auswahl-Zeremonie straff zu halten, wurde in diesem Jahr die Beschleunigung mittels Fernet-Branka-Runden eingeführt, die im 5-10 Minutentakt geschmissen würden. Würden - denn in diesem Jahr war nach einer Runde bereits einstimmig ein Würdiger gefunden. Mit der Verleihung der Swap-Trophäe an Bernd Paysan wird ein Forth-Enthusiast geehrt, der mit big-FORTH/MINOS und mit gForth gleich an der Entwicklung von zwei hervorragenden Systemen nennenswerten Anteil hat. Ich freue mich, Bernd im kommenden Jahr bei einer hoffentlich wieder so entspannten Sitzung des Drachenrates begrüßen zu dürfen.

Ulrich Hoffmann
<http://www.informatik.uni-kiel.de/~uho/swap/>

Ralf Lankau
Handbuch für Grafiker
447 Seiten, Hardcover, Hanser
DM 79,- / ÖS 577,- / SFr 72,-
ISBN 3-446-19071-6



Webdesign und Webpublishing ist ein ungewöhnlich gelungenes Buch zur Umsetzung von grafisch orientierter Werbung im Internet. Es besticht vor allem durch die Informationsdichte sowie durch die sprachliche Aufbereitung der Inhalte. So macht Wissensvermittlung Spaß !

Wenn es darum geht eine Gestaltungsaufgabe für das Internet zu realisieren, findet der Leser hier alle Tücken, Hürden und Fallstricke aufgezeigt, die bei der Bearbeitung zu berücksichtigen sind. Die verständliche Einführung zu dem was das Internet ist, wie es funktioniert oder welche Arbeitsmethoden und Werkzeuge es gibt, bilden einen sehr guten Einstieg zu der Frage: Wie kann ich Grafiken bzw. Seiten für das WWW optimal und professionell gestalten ?

Diese Frage wird durch unzählige Praxisbeispiele beantwortet. Dabei steht immer im Vordergrund, was kann ich wie umsetzen ? Worauf muß ich achten ?

Als Zuckerguß auf der Sahnetorte, enthält das Buch eine CD mit Browser, Editoren und Grafikprogrammen für Windows und Mac. Also probieren Sie mal ein Stück, ich bin überzeugt auch Nicht-Grafiker kommen auf den Geschmack. Gönnen Sie sich mal eine Runde klassisches Offline lesen, bevor sie Online Werben. Wer lesen kann ist hier klar im Vorteil.

mkm



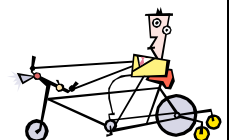
Urlaub werden die meisten von Ihnen bereits gehabt haben, wenn die Redaktion der VD in den selbigen, wohlverdienten geht. Der August ist dafür "ausgeguckt".



Selbstverständlich müssen Sie deswegen nicht auf den Kontakt zu Ihrer VD Redaktion verzichten. Ihre Beiträge schicken Sie bitte einfach - wie gehabt - an die EMAIL Adresse der Redaktion. 'Notfalls' steht auch Ulrich Richter unter U.RICHTER@MHB.GUN.DE zum Empfang Ihrer Aufsätze und Beiträge bereit. Und in ganz dringenden Fällen können Sie ihn sogar anrufen: **02841-22729**.

Falls Sie die VD gerade mit in den Urlaub genommen haben, wünschen wir Ihnen angenehme und erholsame Ferien. Wenn Sie Ihre Ferien bereits hinter sich gebracht haben, dann waren diese hoffentlich schön.

Ihr Editoriat





Den nachfolgenden Text haben wir aus dem Netz 'gefishcht', wo in DE\COMP\LANG\FORTH eine interessante **Diskussion** über **Typen in Forth** geführt wurde. Wir können solche Diskussionen nur auszugsweise wiedergeben und stellen hier Claus Vogt's Ansicht zu diesem Thema vor.

Die Frage der Typüberprüfung würde ich gerne in den Zusammenhang der Überprüfung allgemein stellen. Da wo ich Typüberprüfung kennen (und schätzen) gelernt habe, ging es mir in den seltensten Fällen wirklich um die Frage von Typen.

Für meine Begriffe ist Typüberprüfung eine von mehreren Möglichkeiten, Tippfehler und kleinere Denkfehler abzufangen. Wenn ich z.B. in C versehentlich zwei Parameter verkehrt herum angebe, meldet der Compiler oft (aber absolut nicht zwingend) einen Typfehler. Wenn ich versehentlich eine Routine aufrufe, die eher für interne Zwecke gedacht war, als für 'End'-User, weil ich das falsche Kapitel im Handbuch erwischt habe, habe ich wieder gute Chancen, das angemerkert zu kriegen. Oder wenn ich wieder lange Zeit kein C gemacht habe und sowas schreibe, wie 'if (a=b) ...', wo eigentlich ein doppeltes '=' hin muß, dann hat der Compiler wenigstens eine Chance, das anzuwarnen.

In Forth würde es mir durchaus gefallen, wenn ein Compiler meine kleinen Nachlässigkeiten anmäkeln würde. Z.B. gelingt es mir auch nach 10 Jahren Forth immer noch, Date und Adresse beim '!' (Store) verkehrt herum zu übergeben. Oder ich verhakele mich in den nicht unbedingt gut dokumentierten Schnittstellen fremder Programme, zumal wenn ich unter Zeitdruck einige hundert Zeilen zu produzieren habe. Am härtesten finde ich, in einem sehr großen Programm ein paar kleine Features hinzuzufügen.

Da muß man an sehr vielen Stellen sehr kleine Änderungen machen und hat meist nicht die Zeit, sich in den alten (vielleicht damals schon etwas unordentlichen) Code hineinzudenken. Meine Programmhauptschleife prüft regelmäßig den Stack. Sonst summieren sich kleine marginale Fehler (ein vergessendes 'drop') in großen Programmen zu unregelmäßig auftretenden stochastischen Abstürzen.

Ich weiß nicht, ob solche Fehler mit Dummheit des Programmierers (also meiner Person) zu tun haben. Mein Ehrgeiz war noch nie sonderlich stark darauf gerichtet, daß jeder Tastendruck und jedes kleinste Programmfragment hundertprozentig perfekt ist. Mein Ehrgeiz liegt eher in den Konzepten, die hinter Programmen stecken. Es mag bessere Programmierer geben, die außer guten Konzepten auch 500 Zeilen Assemblercode runterschreiben können, und das läuft dann gleich beim ersten Versuch. Ich gehöre nicht dazu.

Und verstärkend: Die Programme werden nicht kleiner, sondern größer. Ich halte es durchaus für denkbar, daß Programmierertätigkeit in ein paar Jahren überhaupt nichts mehr mit dem zu tun hat, was für mich mal die eigentliche Programmierung war: Nämlich die Implementierung von klugen Al-

gorithmen in Routinen von überschaubarer Größe.

Vermutlich wird Programmierertätigkeit eher das schnelle Anpassen vorhandener Beispiele (vielleicht von mehreren tausend Zeilen Quellcode) an ein ganz gering anders gelagertes Problem sein. Unterstützt vermutlich durch Browser, Case-Tools, Programmgeneratoren, automatische Internet-Update- und was weiß ich. Vielleicht geht es dann nur noch darum, halbwegs treffsicher mit der Maus das richtige Tool am richtigen Ende zu ziehen und ein paar Zeilen Code an siebzehn Stellen auszutauschen, die ohne Suchmaschine kaum noch zu finden sind.

Wenn meine Vermutung richtig ist, sollten wir **mehr** Prüfungen in Forth-Systemen vorsehen. Typprüfungen wäre mir nicht so wichtig. Das können die anderen besser mit ihren typisierten Sprachen.

Vielleicht mal als Frage formuliert:

- Welche Programmier-Fehler werden in Forth häufig gemacht?
- Welche Programmier-Fehler haben besonders fatale Folgen?
- Welche Fehler-Prüfungen wären in Forth leicht zu implementieren?

Claus Vogt

EMP: /DE/COMP/LANG/FORTH

BET: Forth und Modernität

ABS: clv@clvpoint.forth-ev.de (Claus Vogt)

Hallo,

eine Frage, die immer wieder gestellt wird und noch nie so richtig befriedigend beantwortet wurde, möchte ich hier mal aufwerfen:

Was ist für ein MODERNES Forth-System nötig ?

(Die Betonung liegt auf modern)

Ich gestehe, daß ich durchaus persönliche Gründe habe, diese Frage zu stellen. Einerseits habe ich mir heute ungefähr 23 mal im Fernsehen angehört, daß Modernität das Gebot der Stunde ist und wieder nicht verstanden, was das eigentlich sein soll, und das ging mir schon in den letzten 20 Jahren so. Und andererseits habe ich gerade ein paar Worte in comp.lang.forth dazu gelesen, und da wollte ich auch mal was sagen.

Präambel: Ein modernes Forth ist eines, das alle (!) soziologischen Ansprüche an Modernität auch und gerade in technischer Sicht erfüllt. Eines, das alle dergestaltigen heutigen und zukünftigen Erwartungen der Benutzer nicht nur erfüllt, sondern sogar in den Schatten stellt (die Erwartungen natürlich, nicht die Benutzer). Ein modernes Forth hat einen modernen Massenmarkt, der sich in heutzutage üblicher Schichtung aus Käufern, Nutzern, Studenten, Raubkopierern und unverschämten Hotline-Anrufern zusammensetzt und sich



(hoffentlich!) auch wieder in diese aufteilen läßt. Daraus ergibt sich zwingend, daß es zu einem sinnvollen Preis verkauft wird, der eine adäquate Bekleidung der Hotline-Damen gewährleistet, damit diese bei Auseinandersetzungen mit der Marketing-Abteilung auch mal Chancen haben, die Käuferwünsche durchzusetzen. Daraus ergibt sich aber ebenso zwingend, daß es mindestens drei Kaufvarianten (z.B. Professional, Enterprise, Star Wars) geben muß, denn sonst könnte der Käufer nie danebengreifen und die Trennung in Käufer und Nutzer würde obsolet. Außerdem würde das den Raubkopierern das schmutzige Geschäft erleichtern, mal von den Studenten ganz zu schweigen!

Das Anforderungsprofil eines modernen Forth ...

... hat 32 bit. Mindestens.
... hat 3-Dimensionale Pushbuttons (mindestens), die auch auf älteren 3-D-Karten und bei voll aufgedrehter Soundunterstützung noch zu hören sind!
... bringt mindestens 3-mal pro Sitzung eine Message 'in a box', auf der eine 'unzulässige allgemeine Schutzverletzung' bemäkelt wird und zwar bevor (!) es dann wirklich richtig abstürzt.
... die andere Meldungsbox (die mit der 'mangelnden Speicherausstattung' des verwendeten Rechners) scheint mir seit etwa zwei Jahren aus der Mode gekommen zu sein. Das war etwa der Zeitpunkt, wo die Provisionen der Halbleiterhersteller auf diese jämmerlichen Margen fielen. Um aber deren Preispolitik reaktiv-rezipierend zu flexibilisieren, würde ich mal einen Optionsbutton vorsehen, wo man die Box an und aus macht.
... der Optionsbutton sollte von mindestens drei unabhängig voneinander arbeitenden Usability-Labors in etwa 17 verschiedenen Menüs versteckt werden. Die Unabhängigkeit erhöht die Akzeptanz durch Objektivität. (Aber vor allem ermöglicht mir das natürlich die Vorlage mehrere Varianten, über die dann die Managementkonferenz entscheiden kann. Die anderen Punkte wären mir nämlich zu wichtig, um den Chefs Alternativen vorzulegen. Man will ja keine schlafenden Löwen wecken. Und dabei wird sowieso der Optionsbutton gewinnen, der so tief in den Menüs versteckt ist, daß die Chefs ihn auch nach stundenlangem Brainstorming nicht wiederfinden. Ich selbst werde ihn also garantiert im Laufe meines Lebens nie mehr zu Gesicht kriegen. Stattdessen werde ich befriedigt die endlos gefüllten Tips&Tricks-Spalten der Illustrierten überfliegen und mich für meine Unterstützung der Pressearbeit loben lassen.)
Ein angemessener Termin für die Managementkonferenz wäre wohl Ostern.
... hat einen Namen, der trendy, hip und cool ist. Ich denke an "Visual++"
... erfährt regelmäßige Updates mindestens zweimal pro Jahr. Das ist für den Support sehr wichtig, denn es gibt nur einen einzigen Satz, den wirklich jeder Hotline-Mitarbeiter bereits nach 5 Minuten Einarbeitungszeit ebenso verbindlich, wie in der Sache überzeugend, fachlich glaubwürdig mit dem nötigen Respekt für die

Wichtigkeit des Kundenanliegens flüssig und ohne Überheblichkeit von sich geben kann.

Der Satz lautet: 'Oh, das tut mir leid. Mit **der** Version kenne ich mich nun leider nicht so gut aus.'
... wird natürlich von Microsoft hergestellt.

Geschichtlicher Hintergrund und zukünftige Umsatzstrategien

Ich denke, die meisten dieser Vorschläge lassen sich recht einfach in die Tat umsetzen, wenn sie denn erst mal als Gesamtlösung begriffen werden.

Beispielsweise wurde die erforderliche Updatefrequenz sogar über einige Jahre bei weitem übertroffen. Gerade in der Anfangszeit von Forth war die Updatefrequenz oft höher als die Prozessorfrequenz :-). (Dieses ulkige Zeichen heißt, daß das jetzt witzig gemeint war). Da aber kein Geld reinkam (das nannte man damals Public Domain) konnte dieser Vorteil nie richtig durch den Support genutzt werden. Diesen unmodernen geschichtlichen Hintergrund gilt es kurz zu beleuchten:

Der Support wurde meist als persönliches Problem des Programmierers wahrgenommen. Da Programmierer leider oft der Ansicht sind, von ihrem Produkt Ahnung zu haben, gab es viele nervenzerfetzende Streitereien zwischen Käufern und Support, (wenn ich das mal Support nenne). Bekannterweise sind ja Leute, die Ahnung von irgendwas haben, meistens doppelt so engstirnig wie ein übellauliger Chef und ein abgedrehter Dumm-User zusammen. Dadurch wurde das Verhältnis von Käufern und Support sehr stark belastet. Das war doof, weil es für beide Seiten belastend war.

Außerdem hat es auch das Verhältnis zwischen ihnen belastet. Das lag daran, daß außer diesen unangenehmen persönlichen Gesprächen praktisch kein Verhältnis bestand. Es wurde damals kein Geld oder Produkt hin- und hergeschoben. Als letzter ideeller Faden zwischen Käufer und Produzent blieb die Nutzung des Programms. Und die ist ja offenbar schon gestört, wenn der Käufer den Support beansprucht.

Ach ja, so unsinnig das klingt: Oft wurde nicht nur kein Geld hin- und hergeschoben. Auch das Produkt wurde nicht übergeben. Das wird im Bürgerlichen Recht gemeinhin als typischer Vertragsbruch gesehen, aber irgendwie gaben die Käufer das Produkt untereinander weiter, teilweise sicher auch, um den Kontakt mit den o.g. Programmierer-Produzenten zu vermeiden.

Die Produzenten wunderten sich über die mangelnde Verbreitung. Die Käufer hätten es wissen können, hatten aber guten Grund zu schweigen. Und eine ernstzunehmende Marktanalyse war gar nicht denkbar. Ach so, dazu muß ich vorausschicken, daß die Bezeichnung "Käufer" natürlich damals zumindest unexakt war. Jemand, der kein Geld bezahlt, ist schon von erfahrenen Juristen nur schwer von einem Nichtkäufer zu unterscheiden. Wie soll das dann eine Marketing-Abteilung leisten? Käufer, Nutzer und Raubkopierer sind nicht mehr voneinander zu unterscheiden! (Die Studenten sind sowieso schwer voneinander zu unterscheiden, weil sie alle picklig



sind und lange Haare haben.)

Kurz und gut: Aufgrund der Egalisierung der Käuferschichten war keine detaillierte Marktanalyse möglich. Die Marktanalyse schrumpfte gewissermaßen auf die Boolesche Entscheidung zusammen: Gibt es einen Markt, oder gibt es ihn nicht? Und egal, wie die Antwort lautet: Auf einer Ein-Bit-Marktanalyse kann man nunmal kein modernes Management, keine Vertriebsstrategie, keine Produktionsauslastung, keine Supportrentabilität und keine Liquiditätsplanung machen. Damit kann nichtmal der begnadetste Werbefachmann den dümmsten Frühstückspräsidenten einlullen. Auch nicht wenn er das eine vorhandene Bit in 120 Punkt halbfett und mit einem Wordart-Button 'keep-it-simple' und dreifacher RTF-Konversion auf 20 Megabyte aufpustet.

Einige weitere der o.g. Vorschläge möchte ich nur kurz streifen. Sicher wird es wegen der 32 bit Streit mit Chuck Moore geben, dem vermutlich schon ein Bit zu kompliziert ist. Sicher wird es mit Bill Gates wegen irgendwas Streit geben. Sicher! Aber Streit sind wir doch gewohnt. Da müssen wir halt durch. Es ist ja für einen guten Zweck. Auch mit dem Namen und dem Optionsbutton wird es Streit unter uns Forthern geben. Macht nix, da fallen dann vielleicht Chuck Moore und Bill Gates im allgemeinen Gewusel nicht mehr so auf. Bei diesem Streit werde ich mich übrigens vehement auf die Seite derjenigen stellen, die "Visual++" für einen doofen Namen halten, weil er erstens zu mainstreamig ist und weil zweitens weder Java noch Jit dabei angemessen assoziiert werden.

Am technisch anspruchsvollsten scheint mir der dritte Punkt in der Liste zu sein. Der mit der 'Unerlaubten Schutzverletzung'. Deshalb habe ich da auch dieses (!) hingemacht, obwohl ich sonst nicht zu solchen Gefühlsausbrüchen neige.

Also ich denke, es ist durchaus zu leisten, in Forth so eine Meldungsbox aufzumachen. Dann denke ich auch, daß Forth-Systeme irgendwann abstürzen. Da waren sie ja gewissermaßen schon immer sehr eng mit Windows verwandt. Wo ich mir nicht so ganz sicher bin, und da sollten mal die Techniker und Windows-Freaks was zu sagen. Also, ich bin nicht sicher, ob man wirklich mit einem Forth-System zuverlässig (!) garantieren kann, daß die Message-Box immer schon vor dem Absturz kommt. Kann mir da vielleicht jemand einen Tip geben, der sich da besser auskennt?

Also, das war jetzt mein Beitrag zum Thema Modernität. Falls jemand was dazu sagen möchte: Es ist nicht so eilig. Ich bin ein geduldiger Mensch und das Thema Modernität ist ja schon seit einigen Jahren ein Dauerbrenner. Das wird nur noch aktueller, je länger man wartet.

Ciao,

Claus Vogt, 1.3.98

Face it. FORTH will never be popular.

So sprach Simon am 4.4.98 auf comp/lang/forth. Zu deutsch: "Macht euch nichts vor. FORTH wird nie weitverbreitet sein." Und er begründete es wie folgt. Ich übersetze:

"C wurde geschaffen, um große Gruppen kaum kompetenter Programmierer in die Lage zu versetzen, große Quasi-Echtzeit Datenbanken heraus zu bringen. Und dafür ist es erstklassig. Die fähigen Individuen dagegen schreiben Treiber. Die weniger Erleuchteten machen 'production code' - Auftragsarbeiten. FORTH ist geschaffen, um dem fähigen Individuum die volle Kontrolle über sein Computersystem zu geben. Macht euch nichts vor. Es gibt nicht genug fähige Individuen, um Forth weit zu verbreiten."

Michael Kalus

Leserbrief zu Fred Behringer:

Real-Mode-32-Bit-Erweiterung für Turbo-Forth in VD 2/98, S.10

Fred Behringer stellt eine Möglichkeit vor, mit Turbo-Forth unter DOS im Real-Mode 4 Gigabyte RAM zu adressieren. Er benutzt dabei den von Harald Albrecht in der c't 1990 vorgestellten Trick. Dabei werden die Segment-Cache-Register so verändert, daß mittels der 32-Bit-Indexregister (ab 80386) auch im Real-Mode auf 4 Gb zugegriffen werden kann.

Für F-PC-Nutzer wird es vielleicht interessant sein, daß ich eine Anpassung des Albrechtschen Codes besorgt habe. An dieser Stelle nochmals vielen Dank an den Heise-Verlag für die Genehmigung zur Weiterverbreitung. Ich nutze die F-PC-Anpassung seit 1993 unter F-PC 3.54, 3.56 und 3.60, um Meßdaten im Extended Memory abzulegen, und habe bisher keine Probleme beobachtet, abgesehen von der von Fred bereits beschriebenen Inkompatibilität mit dem Speicherverwaltungsprogramm EMM386 von MS-DOS.

Außer dem Albrechtschen Code namens FASTMOVE habe ich für F-PC eine Schnittstelle zu HIMEM.SYS namens XMS entwickelt sowie etwa 20 sinnvolle High-Level-Operationen (@G C@G GFILL CMOVEG GDUMP ...) als G@386 zusammengefaßt. Die entsprechenden Dateien finden sich auf der VD-Weihnachts-CD 1995 in der Directory \CLV\F\CLV\ oder auf der KBBS im Archiv F356CLV1.ZIP.

Claus Vogt, Mai'98

Hello All,

Update four (4) for **Win32Forth**, filename 35UPDT04.EXE is on my web page at:

<http://pw1.netcom.com/~zforth/zimmer.html>

I have also uploaded it to taygeta, so Skip should have it online shortly.

There have been numerous fixes, including a class to handle dialog box window, as has been inquired about lately in this news group. I also added forget handling to several classes that had been overlooked previously.

The WinView editor has been enhanced by removing a bunch of bugs, thanks to Bruno Gauthier.

I hope you find these fixes useful.

Tom Zimmer



"embedded"

Besprechung einer Ein-Mann-Zeitschrift

Fred Behringer, München

Mit "Hi there" begann der Brief (an mich?), den Rafael Deliano einem in zeitschriftenartiger Form gehefteten Pack seiner Arbeiten über "embedded controller und Forth" beilegte. Meine Nachfrage, was ich damit solle, wurde von Herrn Deliano zwar etwas ausführlicher, aber wieder mit "Hi there" beantwortet. Ich fühle mich mit "Hi there" nicht direkt angesprochen und werde wohl beim nächsten "Hi there" nicht mehr reagieren. Herrn Deliano kann man das vielleicht insofern nachsehen, als seine Anstrengungen viel Arbeitsbereitschaft verraten und für eine ganze Reihe von Forth-Freunden ganz bestimmt interessant sind. Letzteres ist auch die Meinung von Friederich Prinz, der mich bat, die "Zeitschrift" (man bekommt sie von Rafael Deliano kostenlos zugeschickt) zu rezensieren, obwohl ich betone, daß ich mich nicht als "embedded"-Fachmann betrachte. Hier also mein Versuch, den Inhalt wiederzugeben. Ich verwende weitestgehend die vom Autor stammenden Formulierungen.

1 Read.me

Es scheint ein Sonderheft der VD über "Embedded Systems" geplant gewesen zu sein, ein Projekt, aus dem nichts wurde. Rafael Deliano hat das "Sonderheft" hiermit nun selbst fertiggestellt und verteilt es (auf Anfrage) kostenlos.

2 Datum & Normalzeit

Der Jahr-2000-Fehler oder ein kurzer Abriß der generellen Unzulänglichkeiten eines jeden Kalenders.

4 Marszeit

Welche Uhren wären für eine Marskolonie angemessen? Besprechung des Buches "A Space Traveler's Guide to Mars" von I.M. Levitt.

6 UNIX-Zeit

32-Bit-Zähler, der jede Sekunde um 1 weiterzählt. Sehr effizient, aber für den Menschen ungeeignet. Aufbereitungsprobleme. Der Autor beschreibt sein Programm DATE! und DATE@.

8 Julianischer Tag

Das Umrechnen von Datumsangaben mit dem gregorianischen Kalender ist extrem unhandlich. Scaligers Kalender ist eine Kette aus Tagen, ohne Jahre und Tage, der sich ausgezeichnet für Rechnungen eignet. Obwohl er schon Jahrhunderte alt ist, findet man ihn immer noch in Chronologie, Astronomie und Raumfahrt. Der Autor gibt ein Programm an, das auf einem Algorithmus beruht, den er der Literatur entnommen hat.

9 Heinz Zemanek: Kalender und Chronologie

Eine Buchbesprechung.

12 Treiber für Touch-Memory

Der Autor bespricht Hard- und Software für das TM DS 1991 (Zweipunktsteuerung eines ICs). Listing in nanoForth auf einem Mitsubishi-6502.

16 Treiber für LCD-Display

Trotz der Vielfalt an Herstellern und Bauformen haben Text-LCD-Module eine Gemeinsamkeit: sie verwenden den HD44780 als LCD-Controller. Auch die Pinbelegung der Schnittstelle ist relativ einheitlich. Die hier dargestellte Hardware und Software wird deshalb zwar nicht für alle Typen sofort passen. Sie stellt aber einen guten Ausgangspunkt dar. Die nötigen Änderungen sind meist minimal. Der Autor bespricht Hardware (Timing, Pinbelegung, Verdrahtung, Reset-schaltung, Biasregelung) und Software (Forth).

20 HDLC-Rahmen in Software

High Level Data Link Control ist ein sehr verbreitetes Format zur Übertragung von Datenpaketen. kann man diese exotischen Bausteine vermeiden, indem man die Bits in Software verarbeitet. "Flag, Datenblock, CRC, Ruhe, Coder, Decoder", alles in Forth.

22 Intel-Hex und Motorola-S-Files

Auf einem Einplatinencomputer, auf dem FORTH installiert ist, kompiliert man Software ins RAM und testet sie dort. Dem Autor geht es in seinem Programm um die Übertragung des Binärfiles in einem der genannten Formate für einen Einplatinencomputer, auf dem FORTH installiert ist, auf das EPROM-Programmiergerät.

24 Gerd Klawitter: Zeitzeichensender

Eine Buchbesprechung

25 Konvertierung von nanoForth

nano-Forth ist ein JSR-threaded 16-Bit-FORTH, das konsequent auf die Verwendung in Einplatinencomputern mit 8-Bit-CPU's zugeschnitten ist. Der Autor stellt eine Liste der Befehle auf, die sich von FORTH-83 und figFORTH unterscheiden.

26 Lineare Assemblerprogramme

In FORTHs, die auf Einplatinencomputern laufen, sind meist Postfix-Assembler implementiert. Mit diesen kann man einfach und übersichtlich linearen Code erzeugen. Der Autor entwickelt als Beispiele einen CCD-Zeilensensor, einen ISDN-Signalgenerator und ein trickreiches Programmstück zur Messung kurzer Zeiten.

Meine Meinung: Der Zeitschriftenversuch "embedded" ist lesenswert. Herrn Delianos Energie ist beachtenswert. Seine Bemühungen sind lobenswert. Im übrigen denke ich, daß ich mit Rafael Deliano genauso gut zur üblichen Gesprächsform finden werde, wie mit anderen Forth-Freunden auch.



Prelude - Objekte und Methoden mit Fehlerprüfung

OO - Konzept 'nach Manfred Mahlow'

Prelude

von Klaus Schleisiek
SEND Signal-Elektronik und Netzdienste GmbH
Stubbenhuk 10
20459 Hamburg
kschleisiek@send.de,

In diesem System gibt es mehrere Speicherbereiche mit unterschiedlichen Bitbreiten:

"Hauptspeicher" mit 32-Bit RAM
"Pufferspeicher" mit 16-Bit RAM
"Nichtflüchtiger Speicher" mit 16-Bit E2PROM

Dieses System zu programmieren und dabei ständig gewärtig zu sein, in welchem Speicherbereich man nun gerade arbeitet, um danach aus einem Zoo von Operatoren (@, **RAM**!, **BAT**!, **NV**!, **EROM**!) den jeweils "richtigen" zu benutzen,

Manfred Mahlow hat auf der Jahrestagung der Forth-Gesellschaft 1997 sein "Prelude-Konzept" vorgestellt als einen Mechanismus, mit dem äußerst effizient "early-binding" OBJECTs und METHODen realisiert werden können.

Ich war sofort Feuer und Flamme für seine Erfindung, und ich hatte das Gefühl, daß dies endlich eine forthige Objektorientierung sei, mit der sich eine der Hauptschwächen von Forth kurieren läßt: Zu viele Operatoren (Operator-Overloading). Manfreds Mechanismus fügt dem Forth-Interpreter nur ganz wenig hinzu (den PRELUDE-Mechanismus), ohne in die Syntax und Semantik von Forth einzugreifen, und für den Anwender gibt es drei neue Worte: CLASS, OBJECT und METHOD.

Als ich 1997 auf der Rückreise von der Jahrestagung sein Konzept durchdacht habe, fühlte ich mich an die Zeit erinnert, in der ich den CREATE ... DOES>-Mechanismus verstanden hatte: Ich brauchte 3 Stunden intensiven Nachdenkens, um zu verstehen, warum und wie der PRELUDE-Mechanismus funktioniert. Das ist sicherlich ein Schwierigkeit, die die weitere Verbreitung behindern wird, aber man muß es nur als Forth-Systementwickler verstehen. Die Anwendung der neuen Worte (CLASS, OBJECT, METHOD) ist ziemlich einfach.

Manfreds Realisierung hatte überhaupt keine Fehlerüberwachung. Wenn man auf ein Objekt nicht eine passende Methode anwandte, sondern irgendeine andere Forth-Instruktion, so merkt man das als Anwender erstmal garnicht. Sowa sehe ich als eine muntere Quelle hochwertiger (schwer zu findender) Fehler an, und ich begann, über eine Fehlerüberprüfung nachzudenken. Nach einigen Telefonaten und Faxen mit Manfred und der Erfindung eines Tricks (nämlich die Phrase ' **NOOP PRELUDE !** vor der Definition von OBJECT) entstand eine Fehlerüberprüfung, die brauchbar erscheint.

Wenn man in ein Anwendungsvokabular kompiliert, und wenn dann ein Objekt nicht von einer passenden Methode gefolgt ist, gibt's was auf die Finger.

Nach weiteren 6 Monaten war dann auch der Crosscompiler für unser TMS320C32-System so erweitert, daß Objekte crosscompiliert werden konnten.

war ein Daueralltraum. Es war eine ständig munter sprudelnde Quelle wirklich hochqualitativer Bugs und eine Dauerherausforderung, sinnvolle Namen erfinden zu müssen.

Sobald der Crosscompiler endlich in der Lage war, mit Objekten und Methoden umzugehen, haben wir die CLASSES NV und BAT definiert, in denen auf einer logischen Ebene mit @ und ! immer 32-Bit Daten manipuliert werden. Und dann hatten wir endlich unser erste Bootprom mit dem objektbasierten System, und wir mußten nur noch den Anwendungscode anpassen.

Das brachte uns die nächste Überraschung: Im Anwendungscode haben wir öfter Adressarithmetik betrieben, um auf ein Feld innerhalb eines Verwaltungsrecords zuzugreifen. Und da hat dann der frisch erfundene Fehlerüberwachungsmechanismus, auf den ich so stolz war, fröhlich zugeschlagen, obwohl der Code eigentlich in Ordnung war.

Nach einigem Experimentieren und Nachdenken habe ich dann ein weiteres Wort zum "Object word-set" hinzugefügt, das ich [] genannt habe. Dieser "jetzt bitte keine Fehlerüberprüfung"-Operator schaltet ganz einfach bis zur nächsten Methode die Fehlerüberprüfung aus. Danach war es dann einfacher, die Applikation auf Objekte umzustellen, als ich vorher gedacht hatte.

In der Rückschau bedauere ich es, daß wir nicht eher die Energie aufgebracht hatten, den Mechanismus ans Laufen zu kriegen. Es macht die Arbeit mit unterschiedlichen Speicherbereichen ERHEBLICH einfacher.

Im folgenden Abschnitt ist der Code unserer Implementation. Das Prelude-Konzept ist tief verwurzelt im Vokabular-Mechanismus und deshalb hier erst einmal unser bewährter Vocabulary word-set, den ich bei seiner Funktionalität für erträglich komplex halte:

```
\ the vocabulary "infrastructure" 24/04/98
5 CONSTANT #vocstack \ maximum number of vocabulary
\ laries in the search order

CREATE CONTEXT ( -- addr ) \ vocabulary "stack"
#vocstack ALLOT 0 , \ allocate vocabulary
\ "stack"
CONTEXT #vocstack 0 FILL \ and initialise it
```




```

: VOCABULARY ( <name> -- )
  VARIABLE
  DOES> ( -- ) CONTEXT ! ;

VOCABULARY ROOT      \ eventually it will contain
\ FORTH at the least...

VARIABLE CURRENT

: DEFINITIONS ( ps: ==> ) CONTEXT @ CURRENT ! ;

: ALSO ( -- )
  CONTEXT [ #vocstack 1- ] LITERAL + @
  ABORT" vocabulary stack full"
  CONTEXT DUP CELL+ 4 CMOVE> ;

: ONLY ( -- )
  CONTEXT [ #vocstack CELL+ ] LITERAL 0 FILL
  ROOT ALSO ;

: PREVIOUS ( -- ) CONTEXT CELL+ CONTEXT
  #vocstack CMOVE ;

```

Hier nun unsere Implementation von Manfreds Prelude-Konzept mit Fehlerüberwachung:

```

\ The PRELUDE mechanism 24/04/98

VARIABLE PRELUDE          \ M. Mahlow's Variable

: HEADER ( <name> -- )
  PRELUDE @ ?DUP IF , THEN \ compile PRELUDE
  \ cfa when present
  build-dictionary-header
  PRELUDE @ IF set-prelude-bit THEN
  \ ein zusätzliches Markierungsbit
  PRELUDE OFF ;

: CREATE ( <name> -- )
  HEADER 1 CELLS ALLOT DOES> ( -- addr ) ;

VARIABLE CHECK_METHOD CHECK_METHOD ON
  \ turns error checking on or off

: CLASS ( <name> -- ) VOCABULARY IMMEDIATE ;

CLASS a-class \ we need a first class definition...

: CLASS-CONTEXT? ( -- f )
  CONTEXT @ 1- @ ['] a-class @ = ;

: DOPRELUDE ( cfa -- ) \ execute prelude code
  \ when prelude bit is set
  test-prelude-bit \ this is VERY
  \ implementation dependent...
  IF get-prelude-code EXECUTE
  CHECK_METHOD ON EXIT
  THEN DROP
  STATE @ 0=EXIT
  CHECK_METHOD @ 0=EXIT
  CONTEXT @ CURRENT @ = ?EXIT
  CLASS-CONTEXT? ABORT" not a method" ;

```

DOPRELUDE wird als Teil der INTERPRET-Loop hinter FIND ausgeführt. Hier einige Erläuterungen zum Aussortieren der Fehlerbedingung:

- Wenn das System interpretiert (nicht-compiliert), dann weiß ich, was ich tue und will nicht von einer blöden Fehlerüberwachung genervt werden.
- Genauso, wenn die Variable CHECK_METHOD ausgeschaltet ist, will ich nichts von "Fehlern" hören.

- Wenn ich neue Methoden in ein Class-Vokabular compiliere, würde mir die Fehlerüberwachung nur in die Quere kommen.
- In jedem anderen Fall kann man erstmal davon ausgehen, daß es Blödsinn ist, nach einem Objekt (das einen Class-Kontext geöffnet hat) KEINE Methode anzuwenden. Ansonsten muß eben der Operator [] zuhulfe genommen werden.

Die Objekte und Methoden selbst sind bei uns genauso implementiert, wie sie von Manfred Mahlow 1997 beschrieben wurden.

\ Objects and Methods 24/04/98

```

: \V ( -- )          \ removes class-context
  \ from search order
  CONTEXT CELL+ @ CONTEXT ! ;

: (v>p ( -- ) CONTEXT @ 1- PRELUDE ! \V ;

: METHOD ( -- )
  STATE @ ABORT" METHOD can not be compiled"
  ['] \V PRELUDE !
; IMMEDIATE

' NOOP PRELUDE !
: OBJECT ( -- )
  (v>p STATE @ 0=EXIT
  PRELUDE @ , PRELUDE OFF COMPILE (v>p
; IMMEDIATE

' NOOP PRELUDE ! vor der Definition von OBJECT ist ein
Trick, ohne den der Fehlerüberwachungsmechanismus es verhindern
würde, daß man in einem Class-Vokabular überhaupt ein definierendes
Wort für Objekte definieren könnte. Dadurch, daß PRELUDE auf die CFA
von NOOP gesetzt wird, wird OBJECT von DOPRELUDE als "Methode"
erkannt und nicht weiter fehlerüberprüft. Dieser Prelude-Code von
OBJECT tut aber - garnichts, und ändert demnach auch nicht den
Kontext, wie dies Methoden normalerweise tun (wegen \V).

' NOOP PRELUDE !
: [] ( -- ) \ disables method-error-checking
  \ until a method is compiled
  CLASS-CONTEXT? 0= ABORT" no class context"
  CHECK_METHOD OFF ; IMMEDIATE

: INHERIT ( -- ) \ definition class inherits
  \ context class words
  CONTEXT @ @ CURRENT @ BEGIN DUP @ WHILE @
  REPEAT ! \V ;

```

Also, angenommen wir haben eine Class DISK-ACCESS und wir wollen eine Class MSDOS definieren, die auf den Methoden von DISK-ACCESS aufbaut. Das können wir nun mit folgender Phrase erreichen:

```
MSDOS DEFINITIONS DISK-ACCESS INHERIT
```

Zum Schluß nun die Definition der Klassen NV und BAT, mit denen wir unsere Speicherbereichsprobleme in den Griff bekommen haben:



Prelude - Objekte und Methoden mit Fehlerprüfung

```
\ Class NV "Non-Volatile" in E2PROM memory 24/04/98
: EROM! ( 32b erom.addr -- )
  some complex code to store 32-bits into E2PROM ;

"shadow"-ram-address CONSTANT MEM_NV
a-quantity CONSTANT #NVMAX

CLASS NV NV DEFINITIONS FORTH

VARIABLE PTR PTR OFF \ NV-area allocation pointer

METHOD : ALLOT ( n -- )
  NV PTR +! PTR @ #NVMAX U> ABORT" E2PROM full" ;

: CREATE ( ds: -- )( ib: name )
  NV OBJECT PTR @ CONSTANT ;

: VARIABLE ( ds: -- )( ib: name )
  NV CREATE 1 ALLOT ;

METHOD : ADDR ( nv.addr -- addr ) MEM_NV + ;

METHOD : @ ( rel.addr -- n ) MEM_NV + @ ;

METHOD : ! ( n nv.addr -- )
  DUP #NVMAX U< 0= ABORT" NV address out of range"
  2DUP NV ADDR 2DUP @ =
  IF 2DROP 2DROP EXIT THEN \ value at this
    \ location unchanged
    ! \ store in "shadow" ram-area
    EROM! ; \ store in E2PROM

METHOD : ON ( nv.addr -- ) True SWAP NV ! ;

METHOD : OFF ( nv.addr -- ) False SWAP NV ! ;

METHOD : >MOVE ( addr nv.addr count -- )
  \ move string into nv-area
  BOUNDS ?DO DUP @ I NV ! CELL+ LOOP DROP ;

METHOD : MOVE> ( nv.addr addr count -- )
  \ move string from nv-area to memory
  ROT NV ADDR -ROT CMOVE ;

FORTH DEFINITIONS

\ Class BAT "BAttery-backed" 24/04/98

"battery"-ram-address CONSTANT RAM_BAT
end-of-battery-area CONSTANT RAM_BUF

CLASS BAT BAT DEFINITIONS FORTH

VARIABLE PTR RAM_BAT 2/ PTR !
  \ BAT-area allocation pointer

METHOD : ADDR ( bat.addr -- addr ) 2* ;

METHOD : ALLOT ( n -- )
  PTR @ OVER + BAT ADDR RAM_BUF
  > ABORT" BAT area full"
  PTR +! ;

: CREATE ( <name> -- b.addr )
  BAT OBJECT PTR @ CONSTANT ;

: VARIABLE ( <name> -- b.addr )
  BAT CREATE 1 ALLOT ;

METHOD : @ ( bat.addr -- 32b )
  BAT ADDR DUP CELL+ @ SWAP @ JOIN ;

METHOD : ! ( 32b bat.addr -- )
  BAT ADDR >R SPLIT R@ ! R> CELL+ ! ;
```

```
METHOD : OFF ( b.addr -- ) False SWAP BAT ! ;
METHOD : ON ( b.addr -- ) True SWAP BAT ! ;

METHOD : >MOVE ( addr b.addr quan -- )
  BOUNDS ?DO COUNT I BAT ! LOOP DROP ;

METHOD : MOVE> ( b.addr addr quan -- )
  BOUNDS ?DO DUP BAT @ I ! 1+ LOOP DROP ;

FORTH DEFINITIONS
```

Liebe Mitglieder und Freunde der FORTH-Gesellschaft,

die Teilnehmer der FORTH-Tagung 1998 wissen es ja bereits, ab 1. Juli 1998 wird die FG von Ute Woitzel in Rostock verwaltet.

Das FORTH-Büro hat also dann die neue Adresse:

FORTH-Gesellschaft e.V.
Postfach 161204
D-18025 Rostock
Telefon: 0381-4007872

Für die gute Zusammenarbeit in all diesen Jahren und Ihr Vertrauen bedanke ich mich ganz herzlich und wünsche der FORTH-Gesellschaft viel Erfolg mit FORTH und den neuen Projekten!

Ein ganzer Abschied kann es aber doch nicht sein, Heinz und ich werden deshalb die FORTH-Tagung 1999 im Süden von Bayern organisieren und hoffen auf eine besonders rege Beteiligung.

Wir würden uns wirklich sehr freuen, viele bekannte - und auch neue - Gesichter wieder zu sehen.

Also planen Sie für Ende April 1999 die FORTH-Tagung ein!

Herzliche Grüße aus München

Ulrike Schnitter



Objekte in HolonForth

Aufsatz und Vortrag des Verfassers anlässlich der EuroForml 97

von Wolf Wejgaard
Forth Engineering
6045 Meggen, Schweiz
wejgaard@acm.org

Übersetzung ins Deutsche von Friederich Prinz

Objekte sind wünschenswerte Programmelemente. HolonForth, ein interaktives, plattformübergreifendes Entwicklungssystem, offeriert Mechanismen zur Definition von Objekten in "eingebetteten Systemen". Holon befreit von der Komplexität einer Klassenhierarchie und implementiert 'early binding'. Methoden sind globale Worte, verknüpft mit privaten Messages in einem Objekt-Typ. Diese Methoden können 'vererbt' werden. Werden Methoden ohne eine Message angewendet, dann wird automatisch eine 'voreingestellte' Message ausgewertet. Der Compiler überprüft die Gültigkeit von Messages. Zusätzlich bietet HolonForth einen 'delayed' Objekt-Mechanismus zur Definition von Arrays in algebraischer Notation, z.B. MATRIX(J K).

Objekte

Objekte sind Datenstrukturen mit privaten Zugriffsfunktionen (Methoden), die durch das Senden einer Botschaft (Message) an das Objekt AUSGEWÄHLT (aktiviert) werden. Objekte machen Programme lesbarer und einfacher handhabbar. Objekte werden durch Definitionsworte erzeugt, die Klassen genannt werden. Forth unterstützt einen Mechanismus zur Definition von Definitionsworten. Konventionelle objektorientierte Systeme verstecken Methoden im Inneren von Klassen, weshalb Methoden nur innerhalb der jeweiligen Klasse verwendet werden können. Die Weiterverwendung einer Methode in einer anderen Klasse ist nicht vorgesehen. Um die Weiterverwendung von einmal definierten Methoden zu ermöglichen, wurde ein Vererbungsmechanismus erfunden. Klassen werden von anderen Klassen abgeleitet. Die abgeleitete Klasse nutzt die bereits existierenden Methoden und Datenstrukturen und fügt eigene Elemente hinzu. Gelegentlich redefiniert eine abgeleitete Klasse eine übernommene Methode, um den Methodennamen in neuem Kontext zu verwenden. Dies ermöglicht das 'Überladen' von Funktionen und Polymorphismus.

Gewöhnlich enthält eine objektorientierte Programmierumgebung eine Vielzahl von Klassen, die in eine Klassen-Hierarchie eingebunden sind. Jedes Element hat seinen festen Platz innerhalb dieser Hierarchie. Die Klassen-Hierarchie bringt Ordnung in die Programme. Viele Programmierer und - teilweise auch - ihre Vorgesetzten, lieben diese Ordnung und bevorzugen darum objektorientierte Systeme. Die Hierarchie mag für Programmbibliotheken nützlich sein. Aber es gibt bessere Wege, um ein Programm zu ordnen.

Auch Forth-Systeme nutzen Objekte. Eine Anzahl entsprechender, objektorientierter Implementierungen steht zur Verfügung [1]. Die Syntax der Definition einer Klasse entspricht typischer Weise der Darstellung in Bild 1. Die Datenelemente und die Methoden werden innerhalb der Klassendefinition festgelegt. Der Compiler erzeugt eine private Liste (Tabelle oder Vokabular) mit den Namen der Variablen und Methoden.

HolonForth

```
Class Point super Object
M: move ... ;M
M: set ... ;M
M: get ... ;M
EndClass
```

Typische Syntax einer Klassen-Definition

Diese Definition lässt sich mit HolonForth nicht kompilieren, weil die Definition eines Wortes während der nicht abgeschlossenen Definition eines 'übergeordneten Wortes nicht möglich ist.

Bild 1

HolonForth ist ein interaktives, plattformübergreifendes Entwicklungssystem auf der Basis von Forth [2]. Der Hauptunterschied zu einem konventionellen Forth-System ist das 'Verwalten' der Quelltexte in einer Datenbank; jedes Programmwort ist ein Datensatz. Die Orientierung 'am Wort' eröffnet die Möglichkeit, neue Leistungsmerkmale in der Programmierung mit Forth zu realisieren.

HolonForth ordnet die Worte in einer Struktur aus Modulen und Gruppen, und macht ein Programm lesbar wie ein Buch. HolonForth lässt Sie genauso direkt an einem Targetprogramm arbeiten, wie Sie mit dem Text in Ihrem Texteditor arbeiten. Jede Änderung an einem Targetwort hat einen sofortigen Effekt im Programm des Targets. HolonForth lädt einzelne, geänderte Worte und ersetzt deren Code im laufenden Targetprogramm. HolonForth erzeugt stets nur den Targetcode, der aktuell benötigt wird. Wenn Sie ein Wort laden (kompilieren und zum Target senden), werden alle benötigten 'Subdefinitionen', die von diesem Wort genutzt werden, auf Veränderungen geprüft und - gegebenenfalls - erneut übersetzt und geladen.

Object-Typen

```
: GetPoint ...;
: PutPoint ...;
: MovePoint...;
Message: get
Message: put
Message: move
```

```
Objecttype: Point
Data: integer, integer,
Methods: getPoint Put-Point MovePoint
Messages: get put move
```

Die Definition einer Klasse in HolonForth

Bild 2

HolonForth macht ausführlichen Gebrauch von Objekten; jede Systemvariable ist ein Objekt. HolonForth implementiert allerdings keine Klassen-Struktur. Es gibt keine Vererbung von Klassen. Trotzdem können alle Methoden wieder-, bzw.



Objekte in HolonForth

weiterverwendet werden. Bild 2 zeigt die Definition von Klassen in HolonForth. Das Verhalten eines Objektes wird zur Gänze an *einem* Platz definiert. Sie müssen sich nicht an einer Kette aus Vererbungsinstanzen entlang hangeln, wenn Sie Details des Objektes sehen wollen.

Die Objekt-Syntax von HolonForth basiert auf der Arbeit von Terry Rayburn [3]. Terry begann mit dem CREATE-DOES> Konstrukt und ließ den DOES> Part eine Sammlung von Methoden umfassen. Ich habe diese Technik auf den Meta-Compiler Mechanismus von HolonForth erweitert. Die folgenden Objekt-Typ Compilerworte werden vom Host verwendet.

Objecttype:

Jeder Objekt-Typ ist autonom, er ist nicht von einem anderen Typ abgeleitet (erbt). Darum nenne ich ihn nicht "Klasse". Ich möchte die Notation hierarchischer Klassenstrukturen vermeiden.

Data:

Das Wort Data: ist äquivalent zu CREATE. Der Datenbereich des Objektes wird durch Worte wie integer, und byte, aufgebaut. Diese Worte ersetzen die archaischen ",," und "c,,". HolonForth nutzt keine Namen für Instanzvariablen. Es ist selten ein Problem, direkt an der Datenstruktur im Speicher zu arbeiten, zumindest im Bereich der „eingebetteten Systeme“, wo diese Strukturen meist sehr einfach sind.

Methods:

Die Methoden sind vordefinierte, globale Worte. Sie sind einfach im Quelltext wiederzufinden und ebenso einfach auszutesten. Und sie können problemlos in anderen Objekt-Typen wiederverwendet werden.

Messages:

Messages sind ebenso wiederverwendbare, globale Worte. Zu den Basis-Objekttypen gehören die Messages 'get' und 'put'. Holon trennt Methoden und Messages von einander. In vielen objektorientierten Systemen dient der Name der Methode gleichzeitig als Name der zugehörigen Message an ein Objekt. Wenn wir die Namen von Methoden und Messages von einander trennen, gewinnen wir neue Freiheitsgrade. Methoden sind durch die Nachricht an ein Objekt gebunden, die diese Methode aufruft. Wenn ein und dieselbe Message verschiedenen Methoden zugeordnet werden kann, dann können wir Operatoren "überladen" und Polymorphismen realisieren.

Implementierung

HolonForth ist ein Cross-Compiler. Der Objekt-Typ Mechanismus arbeitet im Host und erzeugt Code im Target. Wenn ein Objekt erzeugt wird, wird der zugehörige Datenbereich im Target aufgebaut, wo auch die Methoden des Objekts definiert sind. Objekte werden zur Compile-Zeit erzeugt, und Messages werden zur Compile-Zeit an diese Objekte gebunden (early

binding). Das bedeutet, daß Objekte auch in eingebetteten Systemen ohne Einschränkung zur Verfügung stehen.

Die Definition eines Objekt-Typs (Bild 2) erzeugt eine Struktur mit folgendem Inhalt im Host:

1. Der Code, der den Datenbereich im Objekt aufbaut. Dieser Code besteht gewöhnlich aus Host-Worten. Er kann aber ebenso gut Target-Worte aufrufen. Während der Programmentwicklung kommuniziert HolonForth mit einem 'lebenden' Target. Target-Worte können auch dann ausgeführt werden, wenn das Programm bereits geladen ist und arbeitet. Dies ermöglicht den Aufbau von High-Level-Forth Konstrukten in der plattformübergreifenden Programmierung.

2. Ein Array von Methoden. Das Array enthält die Execution-Token der Methoden.

3. Ein Array mit den zugehörigen Messages. Eine Message wird durch eine eindeutige Nummer repräsentiert. In einem konventionellen System werden Sie vermutlich einen Hash-Code verwenden. In HolonForth besitzt jedes Wort eine einzigartige, nur zu diesem Wort gehörige Nummer: den Index des Wortes in die Datenbank.

Mechanismus

In HolonForth sind Messages an Objekte gebunden (Prefix Notation, z.B. move ThePoint...). Die Message ist ein 'immediate' Wort, das seine zugehörige Nummer in der Systemvariablen 'Msg#' ablegt. Das Objekt holt diesen Inhalt aus der Variablen und überprüft diese Angabe mit dem Inhalt seiner Messagetabelle. Wenn das Objekt die Message in seiner Tabelle nicht finden kann, dann wird eine Fehlermeldung erzeugt. Anderenfalls reagiert das Objekt auf die mit dem Tabelleneintrag korrespondierende Methode.

Wenn das Objekt im interpretierenden Modus aufgerufen wird, legt es die Adresse seines Datenbereichs auf den Stapel und führt die durch die Message genannte Methode aus. Im kompilierenden Modus kompiliert das Objekt das Execution Token einer Code-Routine, welche die Adresse des Datenbereiches zur Laufzeit übergibt, gefolgt von dem Execution Token der Methode. Der Aufruf eines Objektes kompiliert deshalb grundsätzlich zwei 'Zellen'.

Der Mechanismus bietet die Implementierung einer 'voreingestellten' Methode an. Wenn ein Objekt ohne Message aufgerufen wird, enthält die Systemvariable 'Msg#' eine NULL. Dann nutzt das Objekt die erste Methode seiner Liste. Die erste Methode eines Integers ist GetInteger, woraufhin das Objekt seinen Integerwert auf den Stapel gibt, so wie wir es eigentlich haben wollen.

Indizierte Arrays

Arrays waren ein Problem in Forth. Wie lassen sich am einfachsten Array-Indizes zusätzlich zu Operatoren und Parametern übergeben? Dies ist einer der Fälle, in denen ich die algebraische Notation ARRAY(I J) bevorzuge. Glücklicherweise



se kann der Objekt-Mechanismus in HolonForth sehr einfach so erweitert werden, daß diese Notation möglich wird. Diese Erweiterung ist das 'delayed' Objekt. Ein 'delayed' Objekt interpretiert nicht und kompiliert nicht - es legt statt dessen die Nummer der Message und die Code-Adresse des Objektes auf einen Objekt-Stapel. Die Indizes werden anschließend durch das Programm auf den Daten-Stapel gelegt. Die schließende Klammer ") " am Ende der Definition holt die Message und die Objektadresse vom Objekt-Stapel und startet den Code des Objekts.

Die Indizes können selbst Objekte sein. Sie können definieren: ARRAY (X Y), wobei X und Y Integer-Objekte sind. HolonForth 'vernestet' Objekte innerhalb 'delayed' Objekte. Die schließende Klammer ist unabhängig vom Objekt-Typ. Das Objekt kennt seinen eigenen Typ und reagiert entsprechend. Unterschiedliche Typen schließender Parameter, wie z.B. ") " , ")) " , oder "))) " sind auch bei unterschiedlichen Dimensionen von Arrays nicht notwendig.

Zusammenfassung:

HolonForth bietet Objekte, die auch für die Programmierung eingebetteter Systeme verwendbar sind. Es gibt keine Klassen-Hierarchie. Der Objekt-Mechanismus ist einfach und zuverlässig. HolonForth separiert Methoden von Messages und ermöglicht 'Überladungen' und Polymorphismen ohne die Vererbung von Klassen.

Referenzen

- [1] Andrew McKewan, Object-Oriented Programming in ANS Forth, Forth Dimensions, XVIII, 6 (März 1997), S. 14-29
- [2] Wolf Wejgaard, Holon - A NEW WAY of Forth, Proc. EuroFORTH, 1992, S. 13-18
- [3] Terry Rayburn, METHODS>Object-Priented extensions redux, FORML Proceedings, 1987, S. 343-355

Nachtrag zum Binding:

Das einfache Objektprinzip von Holon ist nicht auf early Binding beschränkt. Holon verwendet bisher nur early Binding (Binden zur Kompilierzeit), da dies in unseren Applikationen genügt. Es wäre aber kein Problem, Holon auch um late Binding zu ergänzen. Hierzu müssen nur die Objekttyp-Tabellen (Meldungen und Methoden) mit in das Zielsystem gelegt werden. Ausserdem brauchen wir ein Hilfswort im Zielsystem, das jeweils die gegebene Methodenreferenz zur Laufzeit über die aktuelle Objekttyp-Tabelle auflöst.

11. Mai 1998 / Wolf Wejgaard

OOP in Forth ? Muß man das wirklich haben ?

Konzepte zur Realisierung von OOP in FORTH wurden auf den Tagungen der Forth-Gesellschaft und in der VD schon mehrfach thematisiert. Für mich, als Anwender aus dem Hobbybereich, hat sich dabei immer die Frage gestellt: "Wer braucht das eigentlich

wozu" ? Für mich ist Forth bereits seit 1969 'objektfähig'. Jede CONSTANT ist ein Objekt, das Code und zugehöriges Datum miteinander vereint. Mehr 'Objektivität' braucht kein Mensch, meinte ich.

Vor einigen Jahren habe ich - nur so zum Spaß - einen Parser "in ZF" geschrieben, mit dessen Hilfe Michael Major anschließend Objekte definiert hat. Diese Objekte konnten weit über die begrenzten Fähigkeiten von CONSTANT hinaus zur Ableitung weiterer Objekte genutzt werden. 'Vererbung' war nie ein Problem... Wir fanden das recht 'witzig', aber im Grunde wenig nützlich, weil einfach viel zu weit von Forth entfernt.

Uns war natürlich klar, daß 'man' nicht an intensiver Arbeit mit Objekten vorbei kommen kann, wenn man mit einem System umgehen muß, daß 'für Windows' arbeitet. Darüber hinaus sahen wir keinen echten Bedarf für 'Objekte in Forth'. Diese Haltung ist mir seither in vielen Gesprächen immer wieder begegnet.

Die beiden vorhergegangenen Aufsätze von *K. Schleisiek* und *W. Wejgaard*, und der folgende Aufsatz von *E. Woitzel*, machen allerdings deutlich, daß Objekte in Forth aus unterschiedlichsten 'Beweggründen' durchaus Sinn machen. Das 'Überladen' von Operatoren und Funktionen, Polymorphismus und nicht zuletzt die Implementierung von 'Sichtbarkeitsregeln' - die ich wahrlich schätzen gelernt habe - können in komplexeren Anwendungen und Umgebungen das Leben des Programmierers erheblich vereinfachen.

Die drei unterschiedlichen Ansätze zur Realisierung von 'OOP' wurden anlässlich eines 'Workshops' auf der diesjährigen Tagung der FG lebhaft diskutiert. Klaus Schleisiek konnte die Teilnehmer mit seiner eigenen Begeisterung für das PRELUDE-Konzept anstecken, das sich vorhandener, klassischer Forth-Konzepte bedient. Wolf Wejgaards Konzept in HOLON erfreut nicht nur mich wegen seiner Einfachheit und Klarheit. Und Egmont Woitzels Konzept in ComForth ist voller technischer 'Finessen', so wie es ein 'Forth für Windows' sein muß. Diese drei 'Leckerbissen' forthigen Schaffens wären für sich allein die Tagung am Niederrhein wert gewesen.

Nun schaffen - aus welchen Gründen auch immer - stets viel zu wenige Mitglieder der FG den Weg zur jeweiligen Jahrestagung. Die Anregung der Diskussion können wir den Daheimgebliebenen natürlich nicht vermitteln. Ausgeschlossen bleiben von den Inhalten der Diskussionen sollen sie aber auch nicht. Darum habe ich die Autoren gebeten, ihre Vorträge für die VD aufzuarbeiten und ihre Arbeit der gesamten FG bekannt zu machen, was alle drei selbstverständlich gerne getan haben.

fep



Objektorientierte Programmierung in comFORTH 4

Teil - 1 -

*Dr.-Ing. Egmont Woitzel
FORTECH Software Entwicklungsbüro
Joachim-Jungius-Straße 9 18059 Rostock
EWOi@fortech.de*

Entwirft man eine objektorientierte Spracherweiterung für Forth, hat man eine ganze Reihe von Entscheidungen zu treffen. Sollen Klassen verwendet werden, oder nur Objekte? Falls Klassen verwendet werden, sollen sie gleichzeitig auch Objekte sein? Soll eine Klasse von einer oder mehreren anderen erben können? Sollen beim Erben die Implementierung von Methoden oder nur die Interfaces geerbt werden? Wie wird die Sichtbarkeit von Methoden und Instanzvariablen gesteuert? Kann man durch alle Objekte einer Klasse gemeinsam verwendete Variablen definieren? Werden Methoden erst zur Laufzeit an den auszuführenden Code gebunden oder zur Übersetzungszeit? Verwendet man die gewöhnlichen Wortlisten für die Bindung zur Laufzeit oder ein Tabellenverfahren? Soll man die Konstruktion und Destruktion von Objekten mit spezialisierten Methoden unterstützen? Soll eine automatische Speicherbereinigung nicht mehr benötigte Objekte vernichten? Soll das Aufrufverhalten eines Objekts fest oder definierbar sein? Sollen Objekte auf einem separaten Stapel liegen?

Das durch comFORTH 4 verwendete Objektmodell orientiert sich sehr stark an C++. Einer der wichtigsten Gründe dafür ist die angestrebte (teil)automatische Übersetzung von comFORTH-Klassen in COM-Objekte. Ein weiteres Argument für diese Wahl besteht darin, daß viele "mehrsprachige" Programmierer mit dem C++-Objektmodell vertraut sind und so konzeptionell nicht umdenken müssen. Zusätzlich wurde das Ziel verfolgt, die objektorientierten Erweiterungen möglichst "stoßfrei" in ein Standard-Forth zu integrieren. Dazu gehört die Verwendung von Wortlisten für die Kapselung von Klassendefinition genauso wie der Verzicht auf einen separaten Objektstapel.

Innerhalb des comFORTH 4 Objektmodells wird wie in C++ zwischen virtuellen und nichtvirtuellen Methoden unterschieden. Es wird echte Mehrfachvererbung unterstützt, allerdings nicht das Konzept des virtuellen Erbens. Die Sichtbarkeit von Definitionen wird durch die Kategorien PRIVATE, PROTECTED und PUBLIC gesteuert. Eine weitere Einschränkung gegenüber C++ ist, daß die Sichtbarkeit geerbter Definitionen nicht mehr beeinflußt werden kann. Ebenso wie in C++ sind Klassen zur Laufzeit zwar als Datenstruktur vorhanden, aber selbst keine Objekte. Daher gibt es keine Metaklassen, so daß eine Virtualisierung von Klassendefinitionen nicht unterstützt wird. Die Konstruktion und Destruktion von Objekten wird vollständig in die Hand des Programmieres gelegt.

Klassen

Klassen werden im allgemeinen in der Form

```
CLASS:  classname  
      ...  
;CLASS
```

definiert. Unter dem Namen classname wird eine Datenstruktur für die Klasse angelegt. Das Verhalten einer Klasse entspricht weitgehend dem einer Wortliste. Im Gegensatz zu gewöhnlichen Wortlisten sind Klassen aber automatisch IMMEDIATE-Worte, also auch während der Compilation aktiv. Zusätzlich wird durch den Aufruf eines Klassennamens nicht die Suchordnung beeinflußt. Die Klasse aktiviert ihre Wortliste nur transient, Definitionen der Klasse werden also nur für einen Moment, nämlich unmittelbar nach dem Nennen des Klassennamens sichtbar. Wird zu diesem Zeitpunkt DEFINITIONS aufgerufen, wird die Klassenwortliste als Definitionswortliste eingetragen und überschreibt die erste Wortliste in der Suchordnung.

Zwischen dem Namen der neuen Klasse und dem abschließenden ;CLASS müssen alle Objektdaten, Vererbungsbeziehungen und virtuellen Methoden definiert werden. Dazu aktiviert CLASS: eine spezielle Wortliste, die einige Worte aus den Wortlisten FORTH und COMFORTH überlagert. Alle Definitionen zwischen CLASS: und ;CLASS gelangen in die Wortliste der Klasse.

Sichtbarkeit

Die zu Klassen gehörenden Wortlisten bestehen zur Steuerung der Sichtbarkeit der in ihr enthaltenen Definitionen genau genommen aus drei einzelnen Wortlisten. In welche der drei Wortlisten eine neue Definition aufgenommen werden soll, wird durch die drei Worte PUBLIC, PROTECTED und PRIVATE gesteuert. Die zuletzt aktivierte interne Wortliste wird in der Klasse gespeichert und überdauert auch vorübergehende Wechsel per DEFINITIONS. Werden in einer Klasse Definitionen gesucht, so wird immer zuerst in PUBLIC, dann in PROTECTED und erst zum Schluß in PRIVATE gesucht. In PROTECTED und PRIVATE wird aber nicht in jedem Fall gesucht. PRIVATE wird nur dann durchsucht, wenn die betreffende Klassenwortliste auch die aktuelle Definitionswortliste ist. Für ein Durchsuchen von PROTECTED genügt es dagegen, daß die zur Definitionswortliste gehörende Klasse von der zu durchsuchenden geerbt hat - falls die Definitionswortliste überhaupt zu einer Klasse gehört.

Das klingt zunächst etwas kompliziert, führt aber zu dem einfachen Verhalten, daß PUBLIC-Definitionen alle sehen, PROTECTED-Definitionen im Inneren der Klasse und ihrer Erben und PRIVATE-Definitionen nur im Inneren der Klasse selbst. Innen und Außen wird dabei durch DEFINITIONS ge-



steuert. Wird eine Klasse durch DEFINITIONS zur Definitionswortliste, ist man in der Klasse, sonst draußen. Da auch WORDS diesen Regeln folgt, werden immer nur die in der konkreten Situation benutzbaren Definitionen angezeigt.

Instanzenvariablen

Die einfachste Form, einem Objekt zugeordnete Daten zu definieren, besteht in der Verwendung von VARIABLE zwischen CLASS: und ;CLASS. Der wesentliche Unterschied zu einem mit dem gewöhnlichen VARIABLE definierten Wort besteht darin, daß eine Instanzvariable mehrfach existieren kann, nämlich in jedem Objekt der Klasse genau einmal. Die Adresse der Variablen läßt sich daher auch nur im Zusammenhang mit einem Objekt bestimmen. Die durch eine Instanzvariable gelieferte Adresse liegt im Forth-Datenraum. Daher können alle auf gewöhnliche Variablen anwendbare Operationen auch auf Instanzvariablen angewendet werden.

Obwohl Instanzvariablen natürlich als PUBLIC definiert werden können, ist es meistens sinnvoll, ihre Sichtbarkeit wenigstens auf PROTECTED zu beschränken. Die als PUBLIC definierten Methoden können dann viel einfacher den Zustand des Objekts kontrollieren.

Einfache Methoden

Diejenigen Funktionen, die unbeschränkten Zugang zum Datenspeicher eines Objekts haben und sich auf genau ein Objekt beziehen, werden als Methoden bezeichnet. Man sagt dann auch: "die Methode wird auf dieses Objekt angewendet". Methoden erhalten als obersten Parameter die Adresse des betreffenden Objekts übergeben und werden analog zu Doppelpunktdefinitionen in der Form

```
M: methodname ( xxx -- yyy )
... ;M
```

definiert. Die Definition muß natürlich innerhalb der zugehörigen Klasse erfolgen, aber nicht zwischen CLASS: und ;CLASS, da hier nicht der Klassen-Compiler verwendet werden soll. Da innerhalb von Methoden häufig direkt oder indirekt auf das Objekt Bezug genommen werden muß, auf das sie angewendet wurde, wird innerhalb der Methode automatisch eine lokale Variable angelegt, die mit der Objektadresse initialisiert wird. Benutzt man innerhalb einer Methode eine Instanzvariable, bezieht sich diese immer auf dieses Objekt, solange man nicht explizit ein anderes benutzt. Ebenso beziehen sich Aufrufe anderer Methoden immer implizit auf dieses Objekt. Die Objektadresse wird automatisch auf den Datenstapel gelegt.

In einigen Situationen muß man auch direkt auf das implizite Objekt zugreifen, beispielsweise um seine Adresse als Ergebnis zurückzugeben. Dafür ist das Wort THIS vorgesehen. THIS

darf nur innerhalb von Methoden aufgerufen werden und liefert zur Laufzeit die Adresse des impliziten Objekts. Zusätzlich wird zur Übersetzungszeit die Wortliste der Klasse transient aktiviert. Dadurch wird erreicht, daß die Wirkung von THIS varname innerhalb einer Methode identisch zum direkten Aufruf von varname ist. Gleiches gilt für Methoden. Diese explizite Aufrufform braucht man manchmal, wenn ein anderes Objekt einer Methode der eigenen Klasse als Parameter mitgegeben werden soll.

Objekte

Bevor endlich ein erstes Beispiel betrachtet werden kann, ist nur noch zu klären, wie ein solcherart spezifiziertes Objekt nun mit Leben erfüllt werden kann. Dazu werden in comFORTH 4 dieselben Worte verwendet, mit denen auch Variablen von Atom-, Record- und Array-Typen angelegt werden. Diese können je nach gewünschter Lebensdauer im Wörterbuch, als lokale Variable oder dynamisch angelegt werden. Der dynamischen Erzeugung widmen wir uns aber erst ein wenig später. Für die Definition von Objekten im Wörterbuch wird das Definitionswort OBJECT in der Form

```
typename OBJECT objectname
```

verwendet, wobei hier als typename der Name einer Klasse eingesetzt werden muß. Solcherart definierte Objekte hinterlassen nach einem Aufruf ihre eigene Adresse und führen zu einer transienten Aktivierung des eigenen Typs. Da die Typaktivierung schon zur Übersetzungszeit erfolgen soll, werden Objekte automatisch als IMMEDIATE deklariert. Zur Laufzeit bleibt als Effekt nur die Lieferung der Objektadresse auf dem Datenstapel übrig.

Analog kann man auch lokale Variablen vom Klassentyp innerhalb von Forth-Code verwenden. Solche Variablen leben nach ihrer Deklaration nur bis zum nächsten EXIT. Sie werden in comFORTH 4 mit OBJ definiert:

```
: name ( xxx -- yyy )
... typename OBJ objectname ... ;
```

Bis auf den Umstand, daß objectname nur bis zur Beendigung der Definition von name verwendet werden kann, reagieren derart definierte Objekte genau wie die mit OBJECT definierten.

Äpfel zählen zum Ersten

Wer Brodie's Bücher gelesen hat, wird sich vermutlich an sein Beispiel zum Zählen von Früchten erinnern. Die wichtigste Idee des Programms bestand darin, vor dem Benutzer die Art und Weise zu verbergen, in der die Adresse für die eigentliche Zählzelle ermittelt wurde. Dadurch konnten die in einem realen Projekt unweigerlich notwendigen späteren Än-



Objektorientierte Programmierung

derungen mit geringstmöglichem Aufwand in das Programm eingebaut werden. Auch wenn die Eleganz der Brodie'schen Lösung kaum erreicht werden kann, ist die Verlockung einer objektorientierten Alternativimplementierung doch ziemlich groß.

Das Ziel der ersten Iteration soll sein, einfach Äpfel zu zählen. Der Programmzustand wird ausschließlich durch die Anzahl der bereits gesammelten Äpfel bestimmt. Um diese Variable herum müssen wir also unsere Datenkapsel bauen. Dazu definieren wir zunächst eine Klasse CAppleCounter mit einer privaten Instanzvariablen #Apples:

```
CLASS: CAppleCounter ( apple counter class )
  VARIABLE #Apples
;CLASS
```

Innerhalb dieser Klasse benötigen wir drei öffentliche Methoden zur Initialisierung, zum Hinzufügen von Äpfeln und zur Anzeige der aktuellen Anzahl.

```
ALSO CAppleCounter DEFINITIONS
PUBLIC
```

```
M: Prepare ( -- )( set number of apples to
zero )
  #Apples OFF ;M
```

```
M: Add ( n -- )( add a given number of app-
les )
  #Apples +! ;M
```

```
M: ? ( -- )( disp. the current number of app-
les )
  #Apples @ . ;M
```

```
PREVIOUS DEFINITIONS
```

Wie man sieht, kann man innerhalb der Methoden #Apples wie eine gewöhnliche Variable verwenden. Die tatsächlich gelieferte Adresse zeigt immer in das Objekt, auf das die Methode angewendet wird. Die Adresse dieses Objekts wird per Konvention nicht im Stapelkommentar der Methoden aufgeführt, da sie beim Eintritt in die Methode automatisch von dort entfernt wird.

Das eigentliche Programm besteht jetzt nur noch aus einem konkreten Objekt dieser Klasse und seiner Initialisierung:

```
CAppleCounter OBJECT Apples ( -- addr )
  ( apple counter )
  Apples Prepare \ initialize it
```

Die Benutzung erklärt sich selbst:

```
3 Apples Add 8 ok
2 Apples Add 8 ok
Apples ? 8 5 ok
```

Ein gravierender Unterschied einer Lösung mit gewöhnlichen

Variablen besteht in der Kapselung der Variablen #Apples. Sie ist außerhalb der Klasse CAppleCounter nicht mehr sichtbar, wie eine Kontrolle mit WORDS ergibt (das I zeigt im WORDS von comFORTH 4 an, daß ein Wort als IMMEDIATE markiert wurde).

```
CAppleCounter WORDS 8 I ? I Add I Prepare ok
```

Aber auch die Methoden sind nur für einen Moment sichtbar. So funktioniert offensichtlich die folgende Sequenz nicht korrekt:

```
Apples DUP . ? 8 143056 142880 ok
```

143056 ist offensichtlich die Adresse des Objekts Apples. Aber ? liefert nicht mehr die korrekte Anzahl der Äpfel. Das liegt aber einfach daran, daß nicht mehr die Methode ? der Klasse CAppleCounter, sondern das normale ? aus der Wortliste FORTH gefunden wurde. Damit liest man zwar eine Speicherzelle im Objekt Apples aus, aber nicht die der Instanzvariablen #Apples zugewiesene. Möchte man auch nach einer Adreßduplikation noch eine Methode aufrufen, muß man wohl oder übel den zugehörigen Klassennamen benennen. Sonst müßte der Forth-Interpreter ja hellseherische Fähigkeiten besitzen. Anstelle der obigen Zeile muß man also folgendes eintippen:

```
Apples DUP . CAppleCounter ? 8 143056 5 ok
```

Typisierte Zeiger

Wie das obige Beispiel zeigt, lebt das Objektsystem von comFORTH 4 von der transienten Aktivierung der zu einer Objektadresse gehörigen Wortliste. Solange man nur mit benannten Objekten arbeitet, ist dieser Zusammenhang immer gegeben. Faktorisiert man jedoch seinen Code, so endet dies meist sehr schnell. Gut faktorisierter Code bezieht sich nur selten auf benannte Variablen, sondern bekommt im Normalfall nur deren Adressen als Parameter übergeben. Man ist daher relativ oft dazu gezwungen, einer Adresse einen Typ zuzuordnen. comFORTH 4 unterstützt das durch spezielle Pseudovariablen für Zeiger.

Zeiger sind eigentlich nichts anderes als per VALUE definierte Pseudokonstanten. Genau wie diesen kann auch Zeigern ein Wert mit Hilfe von TO zugewiesen werden. Im Unterschied zu Pseudokonstanten sind Zeiger aber einem Typ zugeordnet. Werden Zeiger aufgerufen, legen sie ihren aktuellen Inhalt auf den Stapel und aktivieren ihren Typ transient. Dies funktioniert wie bei Objekten schon zur Übersetzungszeit. Falls möglich, wird bei der Zuweisung einer Adresse an einen Zeiger überprüft, ob die Zuweisung zulässig ist. Das funktioniert aber nur bei Klassentypen und nur zur Laufzeit. Die Zuweisung einer Null ist immer zulässig.

Soll ein Zeiger im Wörterbuch angelegt werden, verwendet



man dafür das Definitionswort POINTER in der Form
value typename POINTER pointername

Analog ist eine Definition als lokale Variable mit PTR möglich:

```
: name ( xxx -- yyy )
  ... value typename PTR pointername ... ;
```

Mit diesem Wissen können wir uns an die nächste Iteration des Äpfelzählens wagen.

Äpfel zählen zum Zweiten

Wie das Leben so spielt, war weder uns noch unserem Auftraggeber klar, daß rote und grüne Äpfel nicht in einen Topf geworfen werden dürfen. Brodie löste das Problem ganz einfach durch eine Zeigervariable. Genau das können wir auch. Wir definieren einfach für jede Farbe eine eigene Instanz der Klasse CAppleCounter. Das bisherige Objekt Apples wird ersetzt durch einen Zeiger auf ein Objekt der Klasse CAppleCounter. Abgerundet wird das Programm noch durch zwei Wörtchen, die das der gewünschten Farbe zugeordnete Objekt dem Zeiger zuweisen. An der Klasse CAppleCounter selbst ist keine Modifikation nötig.

```
CAppleCounter OBJECT RedApples ( -- addr )
  ( apple counter )
  RedApples Prepare \ initialize it

CAppleCounter OBJECT GreenApples ( -- addr )
  ( apple counter )
  GreenApples Prepare \ initialize it

RedApples POINTER Apples ( -- addr )
  ( current apple counter )

: Red ( -- )( activate red apples )
  RedApples TO Apples ;

: Green ( -- )( activate green apples )
  GreenApples TO Apples ;
```

Die Benutzung ist wiederum selbsterklärend:

```
3 Apples Add 8 ok
7 Green Apples Add 8 ok
Apples ? 8 7 ok
Red Apples ? 8 3 ok
```

Allerdings wird die Freude über diese schnell durch die vorsichtige Ankündigung unseres Kunden getrübt, daß er manchmal auch die Gesamtanzahl aller Äpfel wissen will. Und gelbe Äpfel wurden auch schon gesichtet.

Mehr über Instanzvariablen

Es existieren noch mehr Möglichkeiten zum Einbau von Instanzvariablen in Objekte als nur VARIABLE. Neben diesen

können auch Pseudovariablen, Zeiger und Objekte als Instanzvariablen vereinbart werden. Das geschieht mit den Worten VALUE, POINTER bzw. OBJECT zwischen CLASS: und ;CLASS. Beim Einsatz von VALUE und POINTER ist zu beachten, daß im Gegensatz zum normalen Gebrauch kein Initialisierungswert angegeben wird, da ja erst bei der Erzeugung eines Objekts tatsächlich Speicher reserviert wird.

Klassendefinitionen

Bislang haben wir im Inneren von Klassen nur Methoden definiert. Es spricht aber nichts dagegen, innerhalb einer Klasse auch ganz gewöhnliche Forth-Worte zu definieren. Sie gehorchen denselben Sichtbarkeitsregeln, denen auch Methoden und alle anderen Definitionen innerhalb der Klasse unterliegen. Dadurch kann man zur Faktorisierung von Funktionen ganz gewöhnliche Doppelpunktdefinitionen benutzen. Solche Definitionen dürfen im Unterschied zu Methoden nicht implizit auf Instanzvariablen und andere Methoden zugreifen, obwohl sie für sie sichtbar sind. Das liegt daran, daß sie bei ihrem Aufruf nicht automatisch eine Objektadresse übergeben bekommen. In C++ werden solche Funktionen wegen der Art ihrer Deklaration als statische Funktionen bezeichnet, die meisten anderen objektorientierten Sprachen bezeichnen solche Funktionen als Klassenmethoden.

Genauso kann man innerhalb einer Klasse auch gewöhnliche Variablen definieren, über die die Objekte einer Klasse miteinander verbunden werden können. Genau das können wir dazu verwenden, alle Äpfelzähler in einer Liste zusammenzufassen.

Äpfel zählen zum Dritten

Unser Auftraggeber macht Ernst. Wir sollen die Gesamtanzahl aller Äpfel zählen. Die drohenden gelben Äpfel warnen uns zusätzlich davor, einfach die roten und die grünen Äpfel zusammenzuzählen. Eine allgemeine Lösung muß her. Wir beschließen also, alle Äpfelzähler-Objekte in einer Liste zusammenzufassen. Dazu wird in jedem Objekt zusätzlich zu dem Zähler ein Zeiger auf einen Nachfolger vereinbart, also auf den Schwanz der Liste.

```
CLASS: CAppleCounter ( apple counter class )
  VARIABLE #Apples
  CAppleCounter POINTER Tail
;CLASS
```

Der Anker für die Liste wird innerhalb der Klasse definiert. Ebenso eine einfache Doppelpunktdefinition mit dem Namen Total?, die diese Liste entlang läuft und die Werte aller Zähler zusammenaddiert. Dabei kann man auch den Gebrauch von lokalen typisierten Zeigern betrachten. Hier wird nur der Name des globalen Zeigers auf die Zählerliste angegeben, da dieser sowohl die Adresse des ersten Zählers liefert als auch die zugehörige Klasse aktiviert. Dem lokalen Zeiger counter wird damit dieselbe Klasse zugeordnet, die auch dem globa-



Objektorientierte Programmierung

len Zeiger CounterList zugeordnet ist, nämlich CAppleCounter. Die Liste wird in der Methode Prepare aufgebaut, die bislang nur zur Initialisierung des Zählers gebraucht wurde. In Prepare läßt sich auch der Gebrauch von THIS für den Selbsteintrag in die Zählerliste beobachten.

ALSO CAppleCounter DEFINITIONS

```

0 CAppleCounter POINTER CounterList ( --
addr )
      ( anchor )

PUBLIC

: Total? ( -- )
  ( display the total count of apples )
  CounterList PTR counter \ get the list head
  0 \ start with zero
  BEGIN \ walk through the counter List
    counter
  WHILE \ there is a current counter
    counter #Apples @ +
    counter Tail TO counter
  REPEAT
  \ that's all
  . ;

M: Prepare ( -- )
  ( set number of apples to zero )
  ( and add the object to the counter
list )
  #Apples OFF
  CounterList TO Tail
  THIS TO CounterList ;M

```

PREVIOUS DEFINITIONS

Die übrigen Methoden und das Anwendungsprogramm sind von diesen Änderungen nicht betroffen, da sich das bisherige Interface nicht verändert hat. Ein Schönheitsfehler bleibt übrig, das Wort Total? ist trotz öffentlicher Definition nur nach der Aktivierung der Klasse CAppleCounter sichtbar. Dem kann man am einfachsten durch eine Redefinition außerhalb der Klasse abhelfen.

```

: Total? ( -- )( syntactic sugar )
  CAppleCounter Total? ;

```

Ein schneller Test zeigt, daß das Verfahren funktioniert:

```

4 Red Apples Add 8 ok
3 Green Apples Add 8 ok
Red Apples ? 8 4 ok
Green Apples ? 8 3 ok
Total? 8 7 ok

```

Ende Teil - 1 -

Die Teile - 2 - und - 3 - finden Sie in den nächsten Ausgaben der VD.

Programmierwettbewerb:

Für ein eigenes, erstes Beispiel ist es zu spät - die VD muß 'raus', wenn Sie pünktlich bei Ihnen auf dem Schreibtisch liegen soll... Trotzdem reizt es mich, mit dem WIN32FOR auszuprobieren, wie sich dort das ÄPFEL&BIRNEN-ZÄHL&SORTIER-Programm 'am pffiffigsten' schreiben läßt. Auch mit dem HOLON würde ich das gerne testen - und werde es wohl auch versuchen.

Dabei weiß ich - das ist der Vorteil des Editors -, daß hierzu von Egmont Woitzel noch 'Einiges' kommt. Ob andere Umgebungen als comFORTH 4 und andere Programmierer als Brodie und Woitzel sich beim Obstzählen schwerer tun, würde sicher nicht nur ich sehr gerne wissen.

Also her mit den kleinen Beispielen ! In welcher Umgebung (mit welchem Forth) geht es einfacher, schöner, ellegant, kürzer oder auch 'nur' forthiger ? (Sie bemerken, daß ich bewußt nur subjektive Kriterien ansetze ?) Läßt sich unter dem Verzicht auf OOP mit dem ZF oder dem F-PC eine sehenswerte Lösung schreiben ? Wie sieht eine PRELUDE-Lösung aus ? WIN32FOR und HOLON habe ich bereits genannt - wer 'traut sich' ? Und die Nutzer von bigFORTH können sicher auch Lösungen beisteuern...

Zu gewinnen gibt es - selbstverständlich - nichts, außer dem Spaß, den gesammelten Erfahrungen und der olympischen Ehre ;-)

fep

```

\-----
\ comFORTH 4
\ Forth Sample ObjectApples1 -- an object oriented
\ implementation of Leo Brodies Apple Counter
\ Iteration 2
\
\ Revision: 0.01
\ Last Update: 1998-May-16
\ Author: E. Woitzel
\-----
\ Synopsis:
\-----
(
Purpose:
demonstrates basic object oriented techniques
Dependencies:
Core
Interface:
)

```

ONLY FORTH ALSO COMFORTH DEFINITIONS

```

CR.( Include ObjectApples Sample [2]...)

\=== Class AppleCounter
\--- class declaration

CLASS: CAppleCounter ( apple counter class )
VARIABLE #Apples
CAppleCounter POINTER Next
;CLASS

\--- class implementation

ALSO CAppleCounter DEFINITIONS
0 CAppleCounter POINTER CounterList

PUBLIC

```

```

: Total? (--)( diplay the total count of apples )
CounterList PTR CurrentCounter \ get the list head
0 \ start with zero apples
BEGIN \ walk through the counter List
  CurrentCounter
  WHILE \ there is a current counter
    CurrentCounter #Apples @ +
    CurrentCounter Next TO CurrentCounter
  REPEAT
  \ that's all
.;

```

```

M: Prepare (--)( set number of apples to zero )
( and add the object to the counter list )
#Apples OFF
CounterList TO Next

```

```
THIS TO CounterList ;M
```

```

M: Add ( n -- )( add a given number of apples )
#Apples +! ;M

```

```

M: ? ( -- )( display the current number of apples )
#Apples @ . ;M

```

PREVIOUS DEFINITIONS

```

\ === Application
\ --- static objects

```

```

CAppleCounter OBJECT RedApples (-- addr)( apple counter )
RedApples Prepare \ initialize it

```

```

CAppleCounter OBJECT GreenApples (-- addr)( apple counter )
GreenApples Prepare \ initialize it

```

```
RedApples POINTER Apples (-- addr)( current apple counter )
```

```
\ --- color selection
```

```

: Red (--)( activate red apples )
RedApples TO Apples ;

```

```

: Green (--)( activate green apples )
GreenApples TO Apples ;

```

```
\ --- statistics
```

```

: Total? (--)( syntactic sugar )
CAppleCounter Total? ;

```

```
\ --- usage sample
```

```

3 Apples Add
7 Green Apples Add
Apples ?
Red Apples ?
Total?

```

```

.( done. )
CR

```

Aus dem Netz gefischt...



Die umfangreichen Untersuchungen der österreichischen Wissenschaftler am mumifizierten 'Ötzi' haben einwandfrei ergeben, daß er, als er noch unter den Lebenden weilte, kein C-Programmierer gewesen sein kann. Die Pathologen konnten Spuren des Hirns nachweisen. Dafür erhärtet sich der Verdacht, daß es sich bei der Leiche um einen ehemaligen FORTHER handeln muß. Einschließlich der Sandalen für das Hochgebirge waren alle seine Werkzeuge selbst gefertigt.

This is actually a pretty nice little example. If you comment out the guts of the 'Calculate' definition, and compile it, then type: 'Start: CalcWindow1', it will put up a window with four little edit boxes and one calculate button. What could be easier.

Tom Zimmer

CRRepublic wrote:

I've pulled most of what works for readability. The edit controles work when accessed outside the window. I.E. after the window is started Typing:

```
GetText: CalcWindow1.Edit_1 NUMBER? drop drop TO VMIN>
```

```

Puts whatever value is in window Edit_1 into the value VMIN>
But Calculate does not work where it is and it can't be moved past On_Init: because
['] Calculate SetFunc: Button_1 is in On_Init:

```

```

>:Object CalcWindow1 <Super Window
> EditControl Edit_1 \ an edit window
> EditControl Edit_2 \ an edit window
> EditControl Edit_3 \ an edit window
> EditControl Edit_4 \ an edit window
> EditControl Edit_5 \ an edit window
> EditControl Edit_6 \ an edit window
> ButtonControl Button_1 \ a button
>

```

```

>: Calculate
> GetText: Edit_1 NUMBER? drop drop TO VMIN>
> GetText: Edit_2 NUMBER? drop drop To VMAX>
> GetText: Edit_3 NUMBER? drop drop To GMIN>
> GetText: Edit_4 NUMBER? drop drop To GMAX>
> GetText: Edit_5 >float F-> GRADEMIN>
> GetText: Edit_6 >float F-> GRADEMAX>
> ;
>

```

```

>:M StartSize: (-- w h) \ the screen origin
> \ of our window
> 250 290
> ;M
>

```

```

>:M StartPos: (-- x y) \ the width and height
> \ of our window
> 200 100
> ;M
>

```

```

>:M On_Init: (-- ) \ things to do at the start
> \ of window creation
> On_Init: super \ do anything superclass needs
> self Start: Edit_1
> 80 24 30 20 Move: Edit_1
> s" 20" SetText: Edit_1
> self Start: Edit_2
> 140 24 26 20 Move: Edit_2
> s" 60" SetText: Edit_2
> self Start: Edit_3
> 80 46 30 20 Move: Edit_3
> s" 20" SetText: Edit_3
> self Start: Edit_4
> 140 46 30 20 Move: Edit_4
> s" 10" SetText: Edit_4
> self Start: Edit_5
> 80 68 30 20 Move: Edit_5
> s" 15" SetText: Edit_5
> self Start: Edit_6
> 140 68 30 20 Move: Edit_6
> s" -15" SetText: Edit_6
>

```

```

> self Start: Button_1
> 100 250 70 25 Move: Button_1
> s" Calculate" SetText: Button_1
> ['] Calculate SetFunc: Button_1
> GetStyle: Button_1
> BS_DEFPUSHBUTTON OR
> SetStyle: Button_1
>

```

```
;M
```

```
>:Object
```

Verbesserung von FDUMP und F. aus Zen Floating Point

von Fred Behringer
Planegger Str. 24, 81241 München

Die Information über den F-Stack, die FDUMP aus meinem Artikel "Zen Floating Point" in VD 1/98 liefert, kann nicht ganz befriedigen. "Leerstellen" werden nicht berücksichtigt. Im vorliegenden Artikel ist eine verbesserte Fassung zu finden. Eng zusammen damit hängt eine "unintelligente" Eigenschaft bei meinem Wort F. aus VD 1/98. Auch das wird hier ausgebessert.

Stichworte: Turbo-Forth-16, 80486, Gleitkommateil. Maschinennahe Direktprogrammierung. DOS.

Ting prima: Der Artikel von C.H. Ting [2] aus der FD 1/97 mit dem Titel "Zen Floating Point" ist wirklich ausgezeichnet. Er hat in England (Alan Wenham [3]) und in Deutschland (mich selbst [1] und zumindest Friederich Prinz) dazu angeregt, über festgefahrene Programmierparadigmen nachzudenken. Nein, keine verbotenen GOTOs und keinen verpönten Spaghetticode. Nein, nur die Tatsache, daß der wertvolle Gleitkommateil des 80486 jetzt schon jahrelang brach liegt, nur weil man meint, an den Maschinencode nicht ranzukommen. Ob dies oder das in Turbo-Pascal geht, wie schwierig es ist, in C++ die Dinge wirklich nach eigenem Gusto in den Griff zu bekommen oder ob selbst der Assembler (welcher eigentlich?) nicht immer das tut, was man haben will - das alles ist irrelevant, wenn man bedenkt, daß man ja auch direktprogrammieren kann. Keiner wirft sein Fahrrad weg, nur weil er sich ein Auto gekauft hat.

Direkte Maschinensprache unter Forth: Es spricht wirklich nichts dagegen, auf die Möglichkeiten der direkten Maschinenprogrammierung zurückzugreifen. Forth macht es einem leicht. Forth als großes Organisationstalent. Und in genau derselben Zeit, in der ich zur Konstruktion einer CODE-Definition im Assembler-Glossar nachschlagen muß (wer weiß denn schon alles auswendig?), kann ich mir auch die 80486-Datenblätter ansehen und die Maschinenbefehle als Hexzahlen so in die CODE-Definition plazieren, wie es mir gefällt. Die betreffende CODE-Definition steht mir ab dann immer und ewig genauso leicht zur Verfügung wie andere Forth-Worte auch.

Forth-Inlinecode: Von Turbo-Pascal her kennt man den Begriff des "Inlinecodes". Hexzahlen. Kein Assembler. Ganz primitiv. Im frühen BASIC beim VC 20 und C16 konnte man mit SYS in den Maschinenraum springen. Bei den OC-CAM-Compilern von INMOS ist es der Begriff des GUY-Konstrukts. Nicht ganz so primitiv. Aber immerhin: Zugelassen werden nur ganz bestimmte Maschinenbefehle. An andere ist auf diese Weise nicht heranzukommen. Was für ein Leben, in dem ich mich mit dem abfinden muß, was mir ein Compiler-Entwickler vorschreibt! Und kein Quelltext! Keine Mög-

lichkeit, selbst etwas zurechtzubiegen. - Paradigma hin, Paradigma her, was ich hier propagiere, ist "Inlinecode" in Forth. Natürlich nur gelegentlich. Natürlich nur, wo's angebracht ist. Über CODE-Definitionen beliebig einbaubar. Beliebiger. Ohne künstlich errichtete Schranken. Die gibt es in Forth ohnehin nicht.

Nur Notlösung: Natürlich ist das Operieren in reiner Maschinensprache keine Methode, die um ihrer selbst willen empfohlen werden kann. Aber bevor man verzagt ... Um beim Fahrradbeispiel zu bleiben: Wie ist es denn am autofreien Sonntag? Da entdeckt man, daß man manches, das sonst nur mit dem Auto erreichbar zu sein scheint, auch mit dem Fahrrad angehen kann. Übel ist nur der dran, der gleich mit dem Auto zur Welt gekommen ist und gar kein Fahrrad kennt.

Über Ting hinaus: Manche Dinge in Tings Artikel gehen nicht. FDUMP gehört dazu. Herrn Ting ist das inzwischen bewußt (Mitteilung von Henry Vinerts). Das schmälert nicht seine Leistungen. Alan Wenham von der FIG UK ist das auch aufgefallen. Ich habe das und anderes in meinem Artikel [1] in der VD 1/98 berücksichtigt und FDUMP und andere Worte neu gefaßt und erweitert.

Ärger mit dem F-Stack: Der Gleitkomma-Stack ist nicht unbedingt das, was man vom Forth-Datenstack her kennt und erwartet. Der Gleitkomma-Stack kann Leerstellen enthalten. Die können beliebig verteilt sein. Wenn unten keine Leerstelle ist, wird oben nichts akzeptiert: Der F-Stack steckt fest, der F-Stack ist voll. Beim F-Stack wird nicht nur einfach eins draufgesetzt, es wird der bisherige Inhalt (nach außen hin) durchgeschoben, ohne Möglichkeit, unten etwas rausfallen zu lassen. Wenn er voll ist, ist er voll und "klemmt".

Verbesserung an FDUMP: Leerstellen im F-Stack sind nicht etwa mit Nullen oder irgendeiner Sonder-NaN (Not a Number) belegt. Ihre jeweilige Belegung, die von meinem bisherigen FDUMP (VD 1/98) auch ausgegeben wird, ist das, was von dem Zustand her übrigbleibt, da die nunmehrige Leerstelle noch keine Leerstelle war. Das verwirrt bei der vom bisherigen FDUMP erzeugten Bildschirmausgabe (meiner jetzigen bescheidenen Meinung, "IMHO", nach). Leerstellen werden nach Konstruktion des F-Stacks nicht über ihre "Belegung", sondern anders (man studiere dazu das Programm) gekennzeichnet. Ebenfalls abrufbar. Im jetzigen ("verbesserten") FDUMP (siehe Programm) werden diese eigentlichen Kennzeichnungen verwendet, um Leerstellen in der Bildschirmausgabe wirklich als "Leerstelle" auszugeben.

Verbesserung an F.: In FDUMP ist F. enthalten. Die eigentliche Verbesserung an FDUMP habe ich an F. vorgenommen. F., die zerstörungsfreie Anzeige einer Gleitkommazahl, benötigt für Zwischenrechnungen (zur Umwandlung F --> D) zwei Plätze auf dem F-Stack. Es war demzufolge beim bisherigen

F. nicht möglich, einen "vollen" F-Stack hintereinanderweg über F. anzuzeigen. Beim jetzt vorzuschlagenden F. ist das dagegen sehr wohl möglich. Geben Sie achtmal hintereinander FLD1 ein und probieren Sie achtmal hintereinander F. aus. (Das neunte Mal F. bringt dann, wie es sein soll, die Meldung "leer"!)

Ich brauche für das verbesserte F. den Puffer FBUFFER und die zwischenspeichernden Worte FSAVE und FRSTOR. Die Verbesserung von FDUMP (zerstörungsfreie Anzeige des gesamten F-Stacks) ist nun im Grunde genommen im verbesserten F. enthalten: Einfach achtmal F. anwenden. Ganz so einfach ist es nicht: Um nicht anderswie umständlich programmieren zu müssen, mußte ich dafür sorgen, daß auch der Inhalt von FBUFFER nochmal zwischengespeichert wird. as wird über CMOVE nach FBUFFER2 erledigt.

Verbesserung an FLOAT?: Bei FLOAT bestehen ähnliche Probleme. Es werden zwischendurch zwei F-Stack-Stellen "verbraten". Mit dem jetzigen FLOAT kann man den F-Stack nicht "vollschreiben". Ich behalte mir eine programmäßige Ausarbeitung für demnächst vor. Man versuche es zur Übung selbst!

Verbessertes Programm als Einschub: Der unten folgende Programmteil, der alle Verbesserungen gegenüber dem Programm aus VD 1/98 enthält, ist an die Stelle des bisherigen Programmteils zwischen das Ende des Wortes JUSTIFY und den Anfang des Wortes FB<F@ zu setzen.

[1] Behringer, F.: Zen Floating Point, Vierte Dimension 14, Heft 1, 1998.

[2] Ting, C.H.: Zen Floating Point, Forth Dimensions, Jan/ Feb 1997.

[3] Wenham, A.: Meandering Forth, Forthwrite 94, Jan 1998.

Und hier das Programm:

```

UNTIL
THEN ;          \ Ende des Wortes JUSTIFY

\ -----Neue Fassung -----

HEX            \ FSAVE und FRSTOR jetzt
              \ vorgezogen

CODE FSAVE ( -- )
  DD C, 36 C,   \ FSAVE FBUFFER
  FBUFFER ,
  9B C,        \ FWAIT: Coprozessor
              \ fertig ?
NEXT END-CODE

CODE FRSTOR ( -- )
  DD C, 26 C,   \ FRSTOR FBUFFER
  FBUFFER ,
  9B C,        \ FWAIT: Coprozessor
              \ fertig ?
NEXT END-CODE

CREATE FBUFFER2 80 ALLOT \ Für zerstörungs-
\ freie Anzeige in FDUMP

DECIMAL

```

```

: F. ( F: f --          \ Ausgebessertes Wort F.
  FEXAM 18176 AND      \ 4700h and
  DUP 16640 AND 16640 = \ 4100h; C3 and C0 = 1 ?
  IF ." leer"         \ DROP FDROP EXIT THEN
  DUP 1280 =          \ 500h;
  IF ." +unendlich"   \ DROP FDROP EXIT THEN
  DUP 1792 =          \ 700h;
  IF ." -unendlich"   \ DROP FDROP EXIT THEN
  DUP 256 =           \ 100h;
  IF ." +NaN"         \ DROP FDROP EXIT THEN
  DUP 768 =           \ 300h;
  IF ." -NaN"         \ DROP FDROP EXIT THEN
  DUP 16384 =         \ 4000h;
  IF ." +0.0"         \ DROP FDROP EXIT THEN
  DUP 16896 =         \ 4200h;
  IF ." -0.0"         \ DROP FDROP EXIT THEN
                    512 OR 1536 =
  IF
    FSAVE FRSTOR      \ Ausgebessert,
                      \ FStack aufbewahren
    6 FFREE 7 FFREE   \ Ausgebessert,
                      \ Platz schaffen
    10. D>F FSWAP
    JUSTIFY
    FSIGN @
    IF ." -" ELSE SPACE THEN
    FDIGITS @ 0 DO FOVER F* LOOP
    F>D <# FDIGITS @ 0 DO # LOOP ASCII
                . HOLD #S #> TYPE
    FEXPONENT @ ?DUP
    IF ." E" 1 .R THEN
    FDROP FDROP FRSTOR \ Ausgebessert,
                      \ FStack wiederherstellen
    FDROP EXIT        \ TOS nach erfolgter
                      \ Ausgabe wegnehmen
  THEN
  ." denormalisiert?" FDROP ;

: FLOAT ( d --; F: -- f ) \ 0.001234, 1234.5678
                          \ als Eingabe
  10. D>F                \ Basis
  D>F
  DPL @ 0
  ?DO FOVER F/ LOOP     \ Auch z.B. 12345.
                          \ zulässig.
  FSWAP FDROP ;

: FDUMP ( -- )           \ Neues Wort FDUMP
  FSAVE                  \ FStack --> FBUFFER,
                          \ FStack dann leer
  FBUFFER FBUFFER2 80 CMOVE
                          \ FStack-Abbild sichern
  FRSTOR                 \ FBUFFER --> FStack
  8 0 DO CR F. LOOP
                          \ FStack registerweise auslesen
  FBUFFER2 FBUFFER 80 CMOVE
                          \ Stack-Abbild nach FBUFFER
                          \ zurückholen
  FRSTOR ;               \ und wieder auf FStack legen

\ ----- Ende der neuen Fassung -----
\ :FB<F@ ( -- )         \ Weiter wie in VD 1/98

```



E-MAIL gefällig ?

Die Redaktion hat mehrere Anrufe, Anschreiben und Hinweise erreicht, daß die Mitglieder der FG nur sehr schwer bis gar nicht via E-Mail erreichbar sind. Dies liegt daran, daß die E-Mail Adressen bisher nicht veröffentlicht wurden, daß einige Mitglieder mehrere E-Mail Adressen haben und von anderen nur veraltete, ungültige Adressen bekannt sind.

Wir wollen der nächsten VD eine aktualisierte E-MAIL Adressliste beilegen. Bitte senden Sie hierzu eine kurze Mail an F.PRINZ@MHB.GUN.DE mit der Adresse, die Sie für sich bevorzugen.

Gehaltvolles

zusammengestellt und übertragen
von Fred Behringer

Forth Dimensions der Forth Interest Group, USA

November/Dezember 1997

7 Forth in Control: Analog/Digital Converters Ken Merk

Wenn Sie angekoppelte Gerätschaften über den Parallelport steuern, haben Sie sicher manchmal den Wunsch, die Geräte zu überwachen und nachzusehen, ob sie richtig funktionieren. Die hierzu benötigten Rückgabesignale können digitaler Art sein, so daß sie der Computer direkt lesen kann, sie können aber auch in analoger Form vorliegen, so daß sie erst in digitale Signale gewandelt werden müssen, bevor sie der Computer verstehen kann. Dieser Artikel liefert eine allgemeine Übersicht darüber, wie Forth in die Steuerung eingebracht werden kann, und geht in diesem Zusammenhang auch auf Hardwarefragen ein.

11 MicroFont: 4x5 Rob Chapman

Eingebettete Steuervorrichtungen (oder müssen wir im Deutschen unbedingt "embedded systems" sagen?) und ähnliche Kleinsysteme kommen nicht selten mit ganz kleinen Flüssigkristall- oder ähnlichen -Anzeigen daher. Beim Autor war es eins von den Ausmaßen 128x64. Solche Anzeigevorrichtungen haben zum Teil nur eine beschränkte Anzahl von Zeichen im Zeichensatz oder sind sonstwie für die beabsichtigte Anwendung nicht optimal eingerichtet. In diesem Artikel wird ein 4x5-Zeichensatz vorgestellt, der alle darstellbaren ASCII-Zeichen enthält und gut lesbar ist. Und der Forth-Code kann ganz leicht verändert werden, um die dargebotenen Zeichen zu erweitern oder sonstwie zu bearbeiten.

22 Misty Beach Forth: An Implementation in Java Mark Roulo

Dieser Artikel liefert eine Übersicht über Misty Beach Forth, eine Forth-Implementation, die unter Java läuft. Auf den ersten Blick scheinen sich Forth und Java leicht aneinanderzufügen. Die virtuelle Java-Maschine hat jedoch einige Eigenheiten, die das Aneinanderfügen nicht ganz so einfach gestalten. Außerdem hat der Autor, ganz im Sinne der wohldefinierten Java-Semantik, ein paar zusätzliche Beschränkungen eingebaut, die die Misty-Beach-Forth-Implementation unnötig erschweren.

26 Factors Influencing the Use of Forth for Large Projects

Dr. Everett F. Carter, Jr.

Der Autor beschäftigt sich mit den Faktoren, die bei großen Projekten für oder gegen die Wahl von Forth sprechen. Häufig wird die Wahl einer bestimmten Sprache von Überlegungen der Geschäftsleitung, des Kunden oder des Betriebswesens diktiert und ist eine geschäftspolitische und keine technisch bedingte Entscheidung. Von Projekt zu Projekt gesehen, kann man an solchen Vorgaben meist wenig ändern. Auf längere Sicht lassen sich solche Entscheidungen aber durch Aufklärung beeinflussen und durch die Einsicht, daß es keinen technischen Grund gibt, die eine oder andere Sprache vorzuziehen. Der vorliegende Artikel beschäftigt sich hauptsächlich mit den technischen Randbedingungen, die bei großen Systemen wichtig werden.

Gehaltvolles

zusammengestellt und übertragen
von Fred Behringer

FORTHWRITE der FIGUK, Großbritannien

Nr. 95, März 1998

1 Editorial Chris Jakeman

Chris entdeckt, daß viel mehr Mitglieder viel mehr aktiv sind, als er sich das vor der Übernahme der Redaktionsgeschäfte vorgestellt hat. Schon wieder sind mehrere Forth-Systeme im Entstehen begriffen.

2 Announcing the FIG UK Awards

Zwei jährlich zu vergebende Preise werden ausgelobt, einer für den besten Beitrag zu Forth allgemein und einer für den besten Forth-Artikel (in der Forthwrite).

3 Forth News Chris Jakeman

Neues über Jack Brien und die Homepage der FIG UK - Forth im Space-Shuttle - ShBoom-Chip von Charles Moore, Stack und Register, von Genroco für Satellitenübertragung verwendet - iTV-Mikroprozessor für die Internet-Idiotenbox für unter 50 Dollar - Alan Wenhams F83-Gleitkomma-Bibliothek im Internet auf taygeta - FP auch in Aztec von Thomas Wortington - Forth, auf Java aufgesetzt: Internet-Adressen - PilotFORTH 0.6.0 - Win32Forth 3 .

6 Deferred Language Translation Richard Borrell

Wollen Sie ein Forthsystem, das für den wahlweisen Gebrauch in drei Sprachen konzipiert ist? Richard untersucht Schwierigkeiten beim Debuggen und den Mangel an Lesbarkeit und macht den Vorschlag, das Wort "." zu deferieren. Damit kann man dann, je nach Wahl, Meldungen in dieser oder jener Sprache ausgeben lassen. Hierzu ist anzumerken, daß in Turbo-Forth eben dieses Problem der Mehrsprachigkeit auf Metacompilationsebene gelöst wird. Deutsche Version von und bei mir (der Rezensent).

11 Welcome Disk - Bons Mots Chris Jakeman

MANY (von Chris Jakeman) und PLACE (von Ben Hoyt). Beide (wahrscheinlich anders definiert) befinden sich auch in Turbo-Forth (der Rezensent).

12 File Encryption Mike Greenwood

Mikes Firma heißt "Bagnol Greenwood Partners" und arbeitet mit Tom Zimmers FF-Forth, "solange, bis sie genügend Pennies zusammenhaben, um zu F-PC oder Win32Forth aufsteigen zu können". Es geht um E-Mail-Verschlüsselung und -Entschlüsselung. Ein Pseudozufallszahlengenerator verändert jeden einzelnen Buchstaben. Bei gleichem Anfangswert ("Seed") und bekanntem Algorithmus ist die resultierende "Zufallszahl" natürlich immer und immer wieder dieselbe und der verschlüsselte Text kann so rekonstruiert werden.

18 A Puzzle (again) Chris Jakeman

Keiner springt auf dieses Rätsel ein.

19 The Lighter Side Chris Jakeman

Belustigendes aus dem Internet: "... and God sent him Forth", Genesis 3:23 KJV (King James' Version) usw.

21 Square Roots Once More Fred Behringer

Hm, ... Mein Einstand als Mitglied bei FIG UK: Größte ganze Zahl kleiner/gleich der Wurzel einer gegebenen nichtnegativen Ganzzahl. Binärdarstellung nach dem Positionsprinzip, sukzessive Approximation "von unten" (Turbo-Forth, 32-Bit-Zahlen).

32 Library Surplus Sale Sylvia Hainsworth

Ein Haufen schöne Sachen zum Ausverkaufspreis

24 Forth at Work Chris Jakeman

Raus aus den Universitäten, rein in die Wirtschaft! Interview mit einer Firma, die ihre Wurzeln in der University of Birmingham hat und auf Forth baut.

28 From the NET - Minimal Word Sets Ray Allwright

Ray sucht im Netz und wird fündig: Wie sieht "der" minimale Wortschatz aus, der (allein auf Highlevel-Ebene) ein ANS-Forth-System hochzuziehen gestattet? Er zieht Überlegungen von Bernd Paysan an. Prima! Solch vernünftige Antworten hätte ich mir auf meine diesbezüglichen Leserbrieffragen in der VD gewünscht (der Rezensent). (Ich habe die Eindeutigkeitsfrage zurückgestellt. Ich wollte nicht "den" minimalen Wortschatz, ich wollte die minimale Zahl von Worten usw. Klar, daß da erst ein ganzes Gebäude von wohldefinierten Begriffen errichtet werden muß, die eindeutig festlegen, was mit der Frage eigentlich gemeint ist.)

33 Letters Chris Jakeman

Ein Brief aus Japan über "Fuzzy-Forth".

Gehaltvolles

zusammengestellt und übertragen
von Fred Behringer

VIJGEBLAADJE der HCC Forth- gebruikersgroep, Nederlande

Nr. 9, April 1998

Treffen am 11.4.98 in Utrecht: Hans Bezemer sprach über 4th, ein von ihm entwickeltes Forth-System für den PC.

Nieuws op 8051 gebied Willem Ouwerkerk

Der Microcontroller 8051 wird schon seit Jahren überall und immer wieder eingesetzt. Es gibt eine ganze Reihe neuerer Typen dieser Familie. Der Autor bespricht Fabrikate von Atmel, Dallas, Philips, Intel, Siemens, EPIC und UMC.

Het Forth-BBS (2) Leendert van den Heuvel

Diesmal geht es um einen automatisierten Zugang zu dem Bulletin Board System (BBS) unserer holländischen Forth-Freunde.

Een Grootst Gemene Deler Leendert van den Heuvel

Größter gemeinsamer Teiler nur mit Schieben und Abziehen. "In Maschinencode programmiert, ist das eine recht schnelle Methode", sagt der Autor. Verwenden tut er aber "nur" Colon-Definitionen, 7 Stück an der Zahl.

Forth-gg produkten

Frühere Vijgeblaad-Hefte: 2,50 Gulden pro Heft. Noch vorrätig sind die Nummern 36, 37, 41, 42, 43, 45, 46. Versandkosten 12,50 Gulden.

Bestellungen bei Willem Ouwerkerk, Boulevard Heuvelink 126, 6828 KW Arnhem. Tel: 026 443 1305 .

Neues aus Rußland, Frankreich, England und Amerika

zusammengetragen von Fred Behringer

In Rußland scheint es keine organisierte Forth-Gruppierung zu geben. Um so erfreulicher ist es, wenn man auf eine unschuldige E-Mail-Anfrage eine klare und ausführliche zügige Antwort bekommt. Michael L. Gassanenko gehört zu den versprengt wirkenden Forth-Aktivisten Rußlands. Er hat kürzlich auch in der Forth Dimensions veröffentlicht und mir wertvolle Hinweise auf weitere Forth-Adressen in Rußland, die ich leider noch nicht nachprüfen konnte (es gibt solche, die antworten, und solche, die nicht) in ausgezeichnetem Englisch gegeben. Seine Adresse: **Michael L. Gassanenko** <mlg@ias.spb.su>

Von **Marc Petremann** <MP7@wanadoo.fr> aus **Frankreich**, dem Schöpfer von Turbo-Forth, kommt der (wirklich nur ganz ganz kurze) Hinweis darauf, daß seit Mai 1998 das Update 04 von **Win32Forth 3.5** auf <ftp://www.taygeta.com/pub/Forth/Compilers/native/windows/Win32For/35updt04.exe> liegt. Nicht immer liegt in der Kürze die Würze. Wahrscheinlich kommt da noch mehr. Über die Beweggründe und das weitere Drumherum war nichts zu erfahren. Wir in Deutschland verbinden eigentlich mit Win32For den Namen Tom Zimmer. Aber es kann ja jeder selbst dort nachsehen.

Zu den Forth-Freunden aus **England** haben wir jetzt eine sehr gute Beziehung. Ihre außerordentlich gut gestaltete Homepage findet sich unter <http://www.users.zetnet.co.uk/aborigine/forth.htm> Wenn man die Seite aufschlägt, springt einem sofort das Link zur Deutschen FORTH-Gesellschaft ins Auge. Das finde ich nett. Wir werden Entsprechendes tun.

Aus **Amerika** erreicht uns ein Aufruf zur Teilnahme an der diesjährigen FORML-Tagung vom 20. - 22. November 1998 im schönen Kalifornien. Es ist die zwanzigste Wiederkehr der Tagung und sie steht unter dem Motto "**Forth-Schnittstellen zur Welt**". Es wird die gesamte Mannschaft der amerikanischen Forth-Gewaltigen vertreten sein, Wil Baden, Everett F. Carter, Jr., John D. Hall, Glen Haydon, William Ragsdale, Elizabeth D. Rather, C.H. Ting und natürlich auch Charles H. Moore. Tagungsleiter ist **Marlin Ouerson** <editor@forth.org> . Unterstützt wird er von Robert Reiling. Vortragsanmeldungen werden möglichst schnell erbeten, spätestens jedoch bis 1.10.98 .



Twentieth Anniversary of the FORML Conference

"Forth Interfaces to the World"

20. - 22. November 1998; Pacific Grove, Kalifornien

Das vorgesehene Themenspektrum umspannt u.a. "fremde" eingebettete Systeme (Windows CE, Inferno, pSOS, Vrtx), "Rapid Application Development", Forth auf neuen 32-Bit Embedded Chips, Forth in der Welt von Windows, Koexistenz mit C und Forth im Internet. Unter den Konferenzteilnehmern werden u.a. sein: Wil Baden, Everett F. Carter jr, John D. Hall, Glen Haydon, Charles H. Moore, William Ragsdale, Elizabeth D. Rather und C.H. Ting.

Informationen / Anmeldungen laufen über:

Konferenzvorsitz: M. Ouerson EDITOR@FORTH.ORG
Konferenzleitung: R. Reiling AMI@BEST.COM
Organisationsbüro: OFFICE@FORTH.ORG



14. euroFORTH Konferenz

18. - 21. September 1998 auf Schloß Dagstuhl
(bei Saarbrücken)

Weitere Informationen lassen sich finden unter:

DEC.BOURNEMOUTH.AC.UK/FORTH/EURO/EF98.HTML

Liebe Mitglieder der Forth-Gesellschaft

Die Veränderungen in bezug auf die VD sind seit Januar d.J. mehrfach, sowohl in der VD als auch in den öffentlichen Forth-Foren diskutiert worden. Aufklärungsbedarf besteht zu diesem Thema sicherlich kaum noch. Zusammenfassend sollen an dieser Stelle nur noch einmal die wichtigsten Änderungen genannt werden:

- Die VD hat keinen 'externen', freiberuflichen Editor mehr. Der Editor, besser: die Editoren in der Moerser Forthgruppe, bleiben Mitglieder der Forth-Gesellschaft. Die Arbeit der Editoren wird ehrenamtlich geleistet. Der Verein erstattet bei der Erstellung der VD angefallene Kosten; auf Nachweis. Hier fallen im Wesentlichen Kosten für Büromaterial an. Die Vervielfältigung der VD, über einen Copyshop in Mittweida, wird von Thomas Beierlein organisiert. Einschließlich der Transportkosten der fertigen VD bis zum Forthbüro beträgt der Gesamtaufwand pro Ausgabe zur Zeit ca. 600,- DM.

Die Direktoren sehen ihren Auftrag, ausgesprochen von der Mitgliederversammlung 1997, langfristig 4 Ausgaben der VD zu sichern und die VD finanziell auf eine solide Basis zu stellen, als erfüllt an.

- Die erheblichen Kosteneinsparungen bei der VD haben die finanzielle Situation des Vereins deutlich entspannt. Erstmals bleibt finanzieller Spielraum zur Realisierung von Projekten über die VD hinaus.
- Aufgrund der entspannten Finanzlage des Vereins konnten die Direktoren der Mitgliederversammlung eine deutliche Senkung der Mitgliederbeiträge vorschlagen. Die Versammlung wollte dem Vorschlag, alle Beiträge zu senken, nicht folgen. Insbesondere die anwesenden, fördernden Mitglieder haben sich gegen eine Senkung ihrer Beiträge ausgesprochen.

Vor allem die Halbierung des ermäßigten Beitrages für erwerbslose Mitglieder, für Studenten und für Rentner, hat in der Versammlung einhellige Zustimmung gefunden. Dieser Schritt trägt der aktuellen Situation in der Gesellschaft Rechnung. Der neue, ermäßigte Mitgliedsbeitrag deckt nicht die Kosten (einschließlich Versand zum Mitglied) der VD für das einzelne Mitglied. Hier war sich die Versammlung darin einig, die fehlende Kostendeckung mit den eigenen Beiträgen in einem Akt der Solidarität zu finanzieren.

Die neuen Mitgliederbeiträge gelten ab dem 1. Januar 1999:

Ordentliche Mitglieder 80,- DM/Jahr
Ermäßigter Beitragssatz 32,- DM/Jahr
Fördernde Mitglieder 176,- DM/Jahr

Die Anzahl der Mitglieder mit ermäßigtem Beitragssatz beträgt ca. 10 % der Gesamtanzahl der Mitglieder der Forth-Gesellschaft.

- Das Forthbüro wird ab dem 1. Juni 1998 nicht mehr von Ulrike Schnitter in Unterschleißheim betreut werden. Ulrike Schnitter hat das Direktorium gebeten, sie aus ihrem Amt zu entlassen. Nach mehr als 10 Jahren Arbeit für den Verein ist Frau Schnitter der festen Überzeugung, daß es Zeit ist, das Forthbüro in 'jüngere' Hände zu geben. Das Direktorium dankt Ulrike Schnitter für ihre Arbeit, so wie die Mitgliederversammlung ihr gedankt hat.

Ab dem 1. Juni 1998 leitet das Forthbüro **Ute Woitzel** in Rostock. Das Büro wird dann zu erreichen sein unter:

Forth-Gesellschaft e.V.

Postfach 16 12 04
D-18025 Rostock
Tel.: 0381-4007872
E-Mail: SECRETARY@ADMIN.FORTH-EV.DE

Neue Vorhaben/Aufgaben der Direktoren für das laufende Jahr sind die Repräsentation der FG im Internet und die Begleitung des Projektes Forth(buch). Über das Projekt Forth(buch) werden Sie an anderer Stelle in dieser Ausgabe der VD detaillierte Informationen finden. Wir möchten Sie, liebe Mitglieder, an dieser Stelle dazu aufrufen, sich in dieses Projekt entsprechend Ihrer ganz persönlichen Fähigkeiten, Neigungen, Erfahrungen und Ressourcen einzubringen. Unser Vereinsziel ist die Förderung und Verbreitung der Programmiersprache Forth. Das Projekt ist dazu geeignet, diesem Ziel ein gutes Stück näher zu kommen.

Bericht des Direktoriums

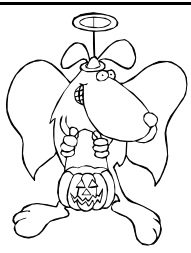
Die Repräsentation der Forth-Gesellschaft im Internet soll unter dem Domain-Namen FORTHEV.DE geschehen. Hierzu muß ein Provider gefunden werden, der uns ausreichend Platz auf einem Server bereit stellt. Der Server muß mit akzeptablen Übertragungsraten adressierbar sein. Das Gesamtkonzept beinhaltet die Forderung nach langfristiger Sicherheit für den Verein, vor allem nach sicherer Finanzierbarkeit. FORTHEV.DE soll unter anderem Links zu den Homepages (sofern vorhanden) der Mitglieder beinhalten, einen Link zum Forthbüro, die VD im *.PDF Format und vieles mehr. FORHEV.DE soll eine ansprechende Adresse für alle werden, die sich im Internet über Forth und die Forth-Gesellschaft informieren möchten.

Es gibt noch vieles zu tun. Die Direktoren lassen nichts davon liegen. Bei allem persönlichen Einsatz ist das Direktorium aber nach wie vor auf die Mithilfe möglichst aller Mitglieder des Vereins angewiesen. Die Projekte Forth(buch) und Internet rufen nach Köpfen und Händen, die bereit sind, sich für ihren (!) Verein zu engagieren. Gleiches gilt für die VD. Alle bisherigen Anstrengungen werden vergeblich sein, wenn Sie unsere Zeitschrift nicht mit den notwendigen Beiträgen 'versorgen' !

Lassen Sie uns gemeinsam tun, was für den Verein zu tun ist.

In diesem Sinne grüßt Sie das Direktorium der Forth-Gesellschaft

Thomas Beierlein
Egmont Woitzel
Friederich Prinz



"Einen zum Abschluß" - hat *Claus Vogt* der Redaktion eingereicht. Die Übersetzung des amerikanischen Originals stammt von *Friederich Prinz*...

(True story)

Während einer jüngst stattgefundenen Computer Messe (COMDEX) hatte **Bill Gates** sich zu folgender **Äußerung** hinreißen lassen: "Wenn GM im gleichen Maße mit der technologischen Entwicklung Schritt gehalten hätte, wie es die Computer-Industrie getan hat, würden wir alle Autos zu 25,- Dollar/Stück fahren, die mit einer Gallone Benzin pro 1.000 Meilen auskommen."

General Motors hatte darauf zunächst geantwortet: "Ja, aber möchten Sie, daß Ihr Auto 2 Mal am Tag kaputt geht ?"

Im Anschluß veröffentlichte General Motors 10 Punkte über Autos á la Bill Gates:

1. Jedes Mal wenn die Farbmarkierungen auf der Straße nachgezogen werden, müßten Sie ein neues Auto kaufen.
2. Gelegentlich würde Ihr Auto ohne erkennbaren Grund auf der Autobahn absterben. Sie würden dies akzeptieren, erneut starten und weiterfahren.
3. Gelegentlich würde Ihr Auto, bei beliebigen Fahrmanövern einfach stehen bleiben. Sie müßten die Maschine re-installieren. Aus unerfindlichen Gründen würden Sie auch dies akzeptieren.
4. Sie könnten stets nur eine Person zur gleichen Zeit in Ihrem Auto haben, es sei denn, Sie hätten "Car95" oder "CarNT" gekauft. Allerdings müßten Sie dann auch zusätzliche Sitze kaufen.
5. Macintosh würde mit Sonnenenergie angetriebene Autos bauen, zuverlässig, 5 Mal so schnell, doppelt so einfach zu fahren. Leider führen diese Autos nur auf 5 Prozent aller Straßen.
6. Die Macintosh-Autobesitzer würden teure Microsoft-Erweiterungen für ihre Autos kaufen, die ihre Autos viel langsamer machen würden.
7. Öl, Benzin und sonstige Warnlichter würden ersetzt werden durch ein einziges "general car default"-Warnlicht.
8. Neue Sitze würden Jedermann zwingen, Jedermann's Abmessungen zu haben.
9. Der Airbag würde Sie vor dem Öffnen fragen: "Are you sure ?"
10. Wenn Sie in einen Unfall verwickelt wären, hätten Sie nicht die leiseste Ahnung, was passiert wäre.

Jerry Eads, American Bio Catalysts

Forth-Gruppen regional

Moers Friederich Prinz
Tel.: 02841-58398 (p) (Q)
(Bitte den Anrufbeantworter nutzen !)
(Besucher: Bitte anmelden !)
Treffen: (fast) jeden Samstag,
14:00 Uhr, MALZ, Donaustraße 1
47443 Moers

Mannheim Thomas Prinz
Tel.: 06271-2830 (p)
Ewald Rieger
Tel.: 06239-8632 (p)
Treffen: jeden 1. Mittwoch im Monat
Vereinslokal Segelverein Mannheim e.V.
Flugplatz Mannheim-Neuostheim

München Jens Wilke
Tel.: 089-89 76 890
Treffen: jeden 4. Mittwoch im
Monat, China Restaurant XIANG
Morungerstraße 8
München-Parsing

mP-Controller Verleih

Thomas Prinz
Tel.: 06271-2830 (p)

Gruppen Gründungen, Kontakte

Fachbezogen 8051 ... (Forth statt Basic, e-Forth)
Thomas Prinz
Tel.: 06271-2830 (p)

Forth-Hilfe für Ratsuchende

Forth allgemein Jörg Plewe
Tel.: 0208-49 70 68 (p)

Jörg Staben
Tel.: 02103-24 06 09 (p)

Karl Schroer
Tel.: 02845-2 89 51 (p)

Spezielle Fachgebiete

Arbeitsgruppe MARC4 Rafael Deliano
Tel./Fax: 089-841 83 17 (p)

FORTHchips Klaus Schleisiek-Kern
(FRP 1600, RTX, Novix) Tel.: 040-375 008 03 (g)

F-PC & TCOM, Asyst Arndt Klingelberg, Consultants
(Meßtechnik), embedded akg@aachen.forth-ev.de
Controller (H8/5xx// Tel.: ++32 +87 - 63 09 89
TDS2020,TDS8092 Fax: ++32 +87 - 63 09 88
Fuzzy

KI, Object Oriented Forth, Ulrich Hoffmann
Sicherheitskritische Tel.: 04351 -712 217 (p)
Systeme Fax: -712 216

Forth-Vertrieb volksFORTH / ultraFORTH
RTX / FG / Super8 / KK-FORTH
Ingenieurbüro Klaus Kohl
Tel.: 08233-3 05 24 (p)
Fax : 08233-99 71

Forth-Mailbox (KBBS) 0431-533 98 98 (8 N 1)
Sysop Holger Petersen
hp@kbbs.org
Tel.: 0431-533 98 96 (p) bis 22:00
Fax : 0431-533 98 97
Helsinkistraße 52
24109 Kiel



Möchten Sie gerne in Ihrer Umgebung eine lokale Forthgruppe gründen, oder einfach nur regelmäßige Treffen initiieren ? Oder können Sie sich vorstellen, ratsuchenden Forthern zu Forth (oder anderen Themen) Hilfestellung zu leisten ? Möchten Sie gerne Kontakte knüpfen, die über die VD und das jährliche Mitgliedertreffen hinausgehen ?

Schreiben Sie einfach der VD - oder rufen Sie an - oder schicken Sie uns eine E-Mail !



Hinweise zu den Angaben nach den Telefonnummern:

Q = Anrufbeantworter
p = privat, außerhalb typischer Arbeitszeiten
g = geschäftlich

Die Adressen des Büros der Forthgesellschaft und der VD finden Sie im Impressum des Heftes.



**Jahrestagung der
Forthgesellschaft e.V.
1998 in Moers
(Neukirchen-Vluyn)**