

Smalltalk in B2B-Anwendungen des Elektronischen Handels

Tutorium

des Fachgebiets *Programmiersprachen und Übersetzer*,
TU Darmstadt



mit Unterstützung der Firmen

Tomcat computer GmbH, Germering

und

CINCOM Systems GmbH & Co, OHG, Frankfurt



8. - 12. April 2002

Smalltalk in B2B-Anwendungen des Elektronischen Handels

Einleitung

Univ.-Prof. em. Dr.-Ing. Hans-Jürgen Hoffmann
TU Darmstadt

8. April 2002

(Web-Fassung - ohne verwendete Tageslicht-Projektionsfolien)



© bei H.-J. Hoffmann

Der **Heißluftballon**, das Erkennungszeichen der "Smalltalker"

Titelblatt des BYTE-Sonderhefts, August 1981, erste breit wahrgenommene Ankündigung von **Smalltalk-80**

(es gab damals auch ein weiteres Bild mit dem *Meer der Stürme* von BAL, Basic, Cobol, Fortran, Lisp, Pascal usw., darüber ruhig schwebend der Smalltalk-Ballon !)



Wer macht **small talk** ?

- *"Plauderei", "Geplänkel ", bei uns nicht ! -*

Wer kennt **Smalltalk** ?

- *Nein, dann wird es Zeit zum Kennenlernen ! -*

Wer kann **Smalltalk** ?

- *Smalltalk in zwei Tagen, das wäre zuviel versprochen;
Nacharbeit nötig ! -*

Wer mag **Smalltalk** ?

- *Nach dem Tutorium alle ! -*

Wer bevorzugt **Smalltalk** für
B2B-Anwendungen ?

- *Wenn es für beabsichtigte
Anwendung Sinn macht,
sollte es jeder tun ! -*

Wer schlägt **Ziel & Anspruch** für
**Smalltalk-B2B-
Anwendungen** vor ?



Smalltalk in B2B-Anwendungen des Elektronischen Handels

→ "Less is more"

→ Geschichtliche Entwicklung

Smalltalk in B2B-Anwendungen des Elektronischen Handels

>> Geschichtliche Entwicklung (I.) <<

- **PhD-Thesis von Alan Kay.** Univ. of Utah, Sept. 1969
The reactive engine
Vision des *Dynabook* (heute PC, Notebook, PDA, WAP u.ä.)
- **Smalltalk-71**, Xerox PARC (Palo Alto Research Center) 1970 und Learning Research Group 1971
Zustandsbehaftete *Objekte*, kommunizieren untereinander (*Botschaften-austausch*) mit Interpretation empfangener Botschaften in *Methoden*
- **Smalltalk-72**
Klassen (***Simula***) und Objekte als deren *Ausprägung*, flach (kein *erben*[†]); erste Implementierung (mit Speicherbereinigung, ***LISP***), GUI mit überlappenden Fenstern, und erste Anwendungen (u.a. KI & *Multimedia-Editor*)
— zeitgleich bei Xerox PARC Entwicklung von erster *Personal Workstation* (*Alto*) —

† *Vererben* ist eine falsche Übersetzung des engl. *to inherit*, auch inhaltlich unzutreffend !

Smalltalk in B2B-Anwendungen des Elektronischen Handels

>> Geschichtliche Entwicklung (Ia. - **Einschub** -) <<

- **Sechs "Hauptideen" von Smalltalk-72**

(siehe A. C.: Kay: The early history of Smalltalk; SIGPLAN Notices, 28 (1993), 3, 78)

1. Everything is an *object*
2. Objects communicate by sending and receiving *messages*
(in terms of objects)
3. Objects have their *own memory* (in terms of objects)
4. Every object is an *instance* of a *class* (which must be an object)
5. The class holds the shared *behavior* for its instances
(in the form of objects in a program list)
6. To *eval* a program list, control is passed to the first object and the remainder is treated as its message

Smalltalk in B2B-Anwendungen des Elektronischen Handels

>> Geschichtliche Entwicklung (II.) <<

- **Smalltalk-74**

- heute würde man von Smalltalk-72, Version *1.irgendetwas* reden —
- "*Resident object table*", Vorstufe für leistungsfähige Speicherbereinigung

- **Smalltalk-76**

(wesentliche Beiträge von *Dan Ingalls* und *Larry Tessler*)

- "Alles ist ein Objekt", auch *Klassen* sind Objekte, ausgeprägt aus (der einen) *Metaklasse Class*, baumförmige *Klassenhierarchie*, generische Identifikatoren *self* und *super*;
- Benutzungsumgebung zum interaktiven, "inkrementellen" Arbeiten mit dem *Smalltalk-"System"*;
- Systemkomponenten zum **Stöbern / Schmöckern** in Klasseigenschaften (und damit auch Objekteigenschaften) und der Klassenhierarchie (die sog. *Stöberer*, engl. *browser*), *Inspektoren* von Objektzuständen, und zur Testunterstützung (engl. die sog. *debugger*);
- *edit-compile-test-Zyklus* für Methoden

Smalltalk in B2B-Anwendungen des Elektronischen Handels

>> Geschichtliche Entwicklung (III.) <<

- **Smalltalk-78**

- wieder so etwas wie Smalltalk-76, Version 1. *irgendetwas* —

- **Smalltalk-80**

- Übersetzen von Methoden in **0-Adress-Code** (*bytecode* — Anleihe bei *Burroughs B5000, Pascal P* — ==> leichtes Portieren !) und damit (zunächst) *edit-compile-interpret-test-Zyklus* auf Methodenebene, *virtual machine (VM)* als Interpreter
 - Klassenhierarchie **auch** bei den Metaklassen, **individualisierte (Klassen-) Methoden** zum Ausprägen von Objekten — siehe spätere Folie — ;
 - **MVC-Paradigma** (heute sagt man dazu **Entwurfsmuster**);
 - Smalltalk-System **in** Smalltalk programmiert;
 - **BYTE-Sonderheft 1981**, breite Fachöffentlichkeit wird auf Smalltalk aufmerksam (insbesondere auch durch Bücher von *Adele Goldberg* et al.) — zeitgleich bei Xerox PARC Entwicklung von verbesserten *Personal Workstations (Dolphin und Dorado)* —

Smalltalk in B2B-Anwendungen des Elektronischen Handels

>> Geschichtliche Entwicklung (IV.) <<

- **Smalltalk-8x**

Verbesserung der Arbeitsweise durch Einführen des *edit-compile-just_in_time_compile (JIT)-test-Zyklus* auf Methodenebene

- **Wichtigste Einflüsse**

Sketchpad, Engelbart's NLS ("oNLine Systems"), Burrough's B5000, Conway's Coroutines, "Flex" von Evans/Cheadle, Joss, Lisp, Planner, Simula, Euler, Logo

- **Von der Smalltalk-Entwicklung ausgehende Einflüsse**

Wer nennt und zählt sie ? Viele bei heutiger Hardware und Software !

- **Fundgruben:**

Alan C. Kay: The early history of Smalltalk

(ACM SIGPLAN Notices, March 1993, 69 - 95)

Dave Thomas: Travels with Smalltalk

(*Celebrating 25 years of Smalltalk*, SIGS Publications, 1995)

Smalltalk in B2B-Anwendungen des Elektronischen Handels

>> Geschichtliche Entwicklung (V.) <<

- Wichtige **Smalltalk Produkte bzw. Anwendungen** ab etwa 1983 bis 1995:
 - > ***Smalltalk-80*** (mit verschiedenen verbesserten Versionen, 1983+) von ***ParcPlace Systems*** — oberes Preissegment für *Workstations* —
 - > ***Smalltalk/V*** von ***Digitalk*** — unteres Preissegment, PCs (nur mit Diskette !), MACs; ***parts*** in Komponententechnologie — **Popularisierung in USA**
 - > Prozeßtechnologie-Anwendungen durch ***Object Technology Inc. (OTI)***, Verbindung zu IBM; ***ENVY/Developer*** mit Komponententechnologie
 - > ***IBM Smalltalk*** und dann ***VisualAge for Smalltalk*** (***VisualAge for Java*** ist weitgehend in Smalltalk programmiert !)
 - > Firmenzusammenschluß von ***Parkplace Systems*** und ***Digitalk***, 1995; Umbenennung des Smalltalk-Systems in ***VisualWorks***
 - > Übernahme der Rechte an ***VisualWorks*** durch ***Cincom Systems, Inc.***
 - > ***VisualWorks*** in Deutschland z.B. vertrieben durch ***Cincom, Georg Heeg*** u.a.
 - > ***Self*** (D. Ungar; übrigens bei Sun bis um 1995 Parallelentwicklung zu Java !), ***Squeak***, ***GNU Smalltalk***, ***Smalltalk/X***, ***Smalltalk MT***, ***Dolphin Smalltalk*** u.a.

Smalltalk in B2B-Anwendungen des Elektronischen Handels

>> Geschichtliche Entwicklung (VI.) <<

- **Aktuelles** in den weiteren Referaten
- In Deutschland Implementierung:
 - für Atari und PCS-Systeme (*Georg Heeg*) um 1985;
 - für Workstations aller Art *Smalltalk-X* (*C. Gittinger*) in den 90ern;

(- der VM in LISP - auf Siemens BS2000 im Rahmen einer Seminar- und Studienarbeit an meinem Lehrstuhl durch *Joachim Richter & Georg Rainer Hofmann*, WS 1983/84 - nicht veröffentlicht -)
- In Deutschland frühe Smalltalk-Begebenheiten:

Fachgespräch *Smalltalk als Beispiel einer interaktiven Arbeitsumgebung* der *GI-Fachgruppe Interaktive Systeme* und *Tutorium über Smalltalk* (GChACM), April 1985, Darmstadt (siehe *Notizen zu Interaktiven Systemen, Heft 14*, und *H.-J. Hoffmann: Smalltalk verstehen und anwenden*; Hanser-Verlag, 1987)

Smalltalk in B2B-Anwendungen des Elektronischen Handels

→ "Less is more"

→ Geschichtliche Entwicklung

→ "Abzählbares" in Smalltalk

Über "Abzählbares" in Smalltalk (II.)

Klasse ***Magnitude*** mit (vielen) Unterklassen: (unvollständig !)

- *Object* > *Magnitude*
- *Magnitude* > *ArithmeticValue* > *Number* > *FixedPoint*
 - > *Fraction* ←
 - > *Integer* > *SmallInteger* ←
- > *Point*
- > *Character*
- > *Date* ←
- > *Time*

- Methoden von *Magnitude*:
Keine *Klassenmethoden*
Objektmethoden:

< <= = > >= dyadische Operationen
between: and: hash: max: min:

Operationen mit "Selektoren"

Über "Abzählbares" in Smalltalk (III.)

Klasse **Date**

Was ist :

- Date newDay: 8 month: #April year: 2002 April 8, 2002
- (Date newDay: 8 month: #April year: 2002) addDays: 4
April 12, 2002
- ((Date newDay: 8 month: #April year: 2002) addDays:4)
weekday
#Friday

Alles sind Objekte, empfangene Botschaften, Antworten/Resultate
(Fast) alle Syntax ist gezeigt!
(etwas mehr noch morgen und übermorgen, insbesondere weitere Klassen)

Über "Abzählbares" in Smalltalk (IV.)

Hierarchy Browser on Magnitude

File Edit View Class Protocol Method Help

ArithmeticValue
Number
FixedPoint

instance class
 shared variables

abs
negated
reciprocal

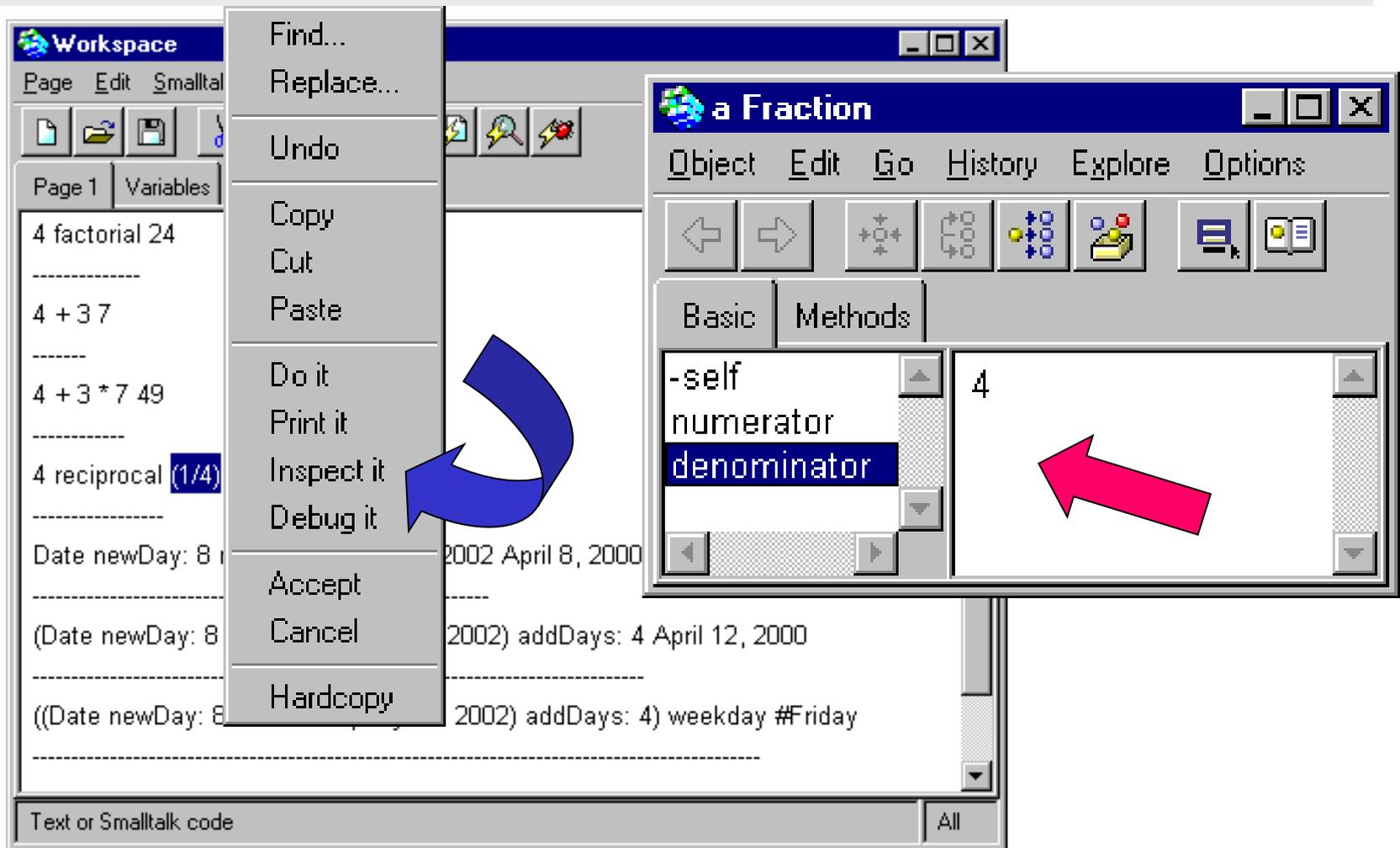
reciprocal

"Answer 1 divided by the receiver. Provide an error notification if the receiver is 0."

`^self class unity / self`

Ein Beispiel, reciprocal

Über "Abzählbares" in Smalltalk (V.)



Über "Abzählbares" in Smalltalk (VI.)

The screenshot shows the 'Hierarchy Browser on Magnitude' window. The left pane shows a class hierarchy with 'Integer' selected. The right pane shows the 'factorial' method. The main area displays the following Smalltalk code:

```
factorial
  "Answer the factorial of the receiver. Fail if the
  receiver is less than 0.

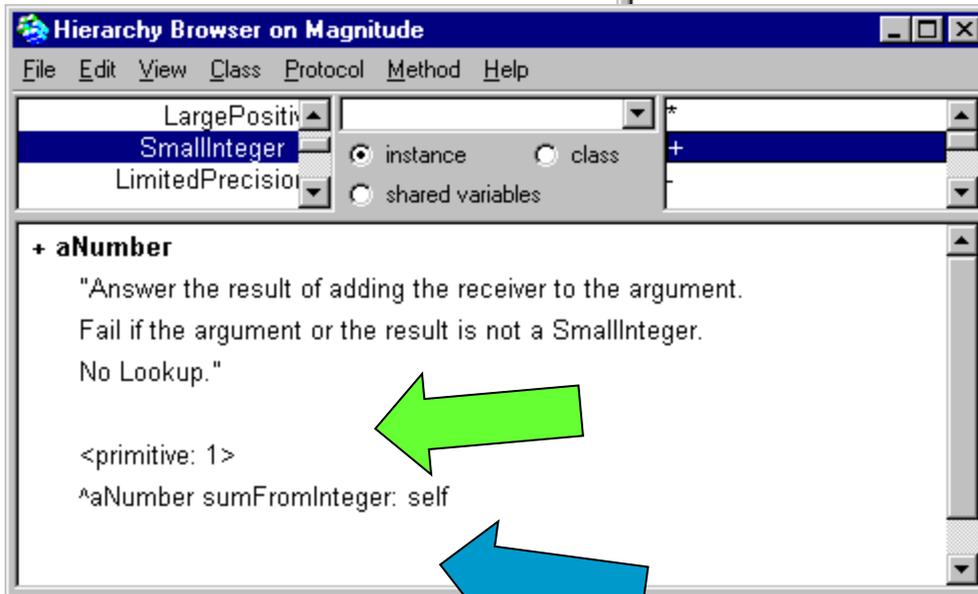
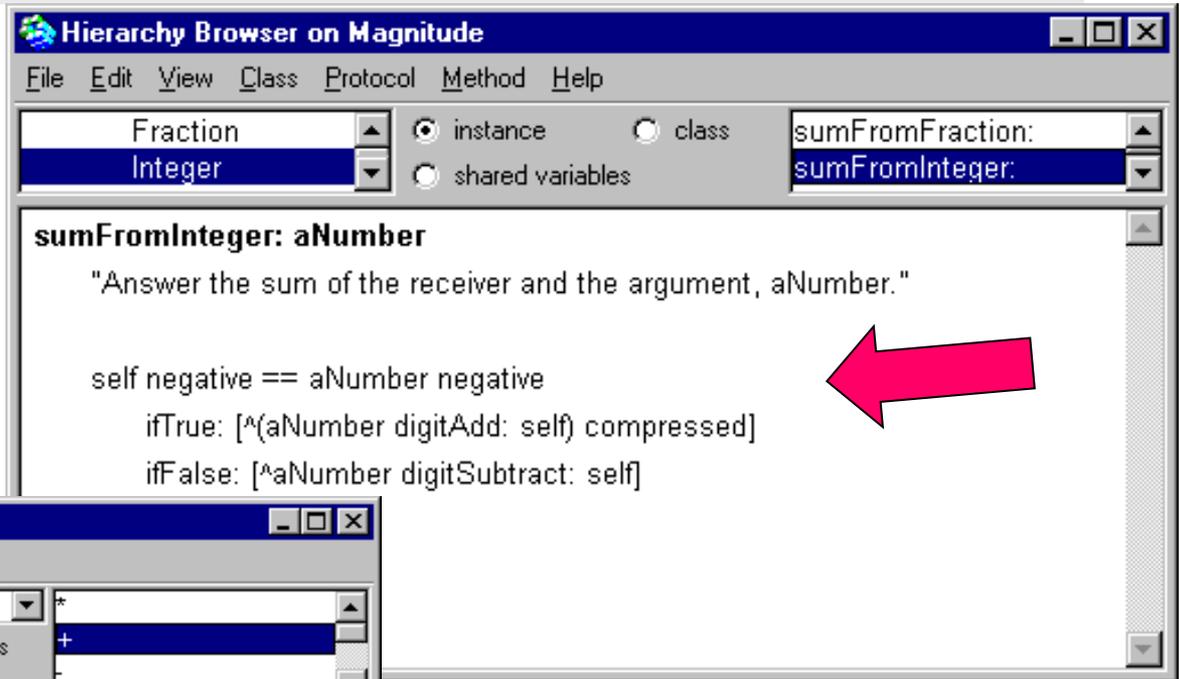
  For example, 6 factorial == 6*5*4*3*2*1."

  | tmp |
  self < 0
    ifTrue: [^self class
      raise: #domainErrorSignal
      receiver: self
      selector: #factorial
      errorString: (#errFactorialInvalid << #dialogs >> 'Factorial is invalid on negative numbers')].
  tmp := 1.
  2 to: self do: [:i | tmp := tmp * i].
  ^tmp
```

A grey arrow points to the line `2 to: self do: [:i | tmp := tmp * i].` in the code. A grey callout box with the text "Ein weiteres Beispiel, factorial" is overlaid on the right side of the code area.

Über "Abzählbares" in Smalltalk (VII.)

Ein drittes Beispiel,
Addition



1. Primitivoperation
2. Typkonversion
3. Ausprogrammiert

Über "Abzählbares" in Smalltalk (III.)

Noch eine "kleine" Aufgabe:

Können Sie das mit Ihrem bevorzugten Rechner und Ihrer bevorzugten Programmiersprache auch so schnell ?

Oder:

Was Sie schon immer wissen wollten ?

Was ist :

- 2 raisedTo: 256

115792089237316195423570985008687907853
269984665640564039457584007913129639936

Bitte Nachrechnen !

Über Ablaufkonstrukte und *"Boolesche Algebra"* in Smalltalk

Klasse ***Boolean***

Was ist :

- `5 < 4` ifTrue: ['falsch'] ifFalse: ['richtig'] **richtig'**
War das eine Überraschung ?

Erklärungen dazu (und noch mehr !)

morgen und übermorgen !

Sie wissen noch nicht alles !

Smalltalk in B2B-Anwendungen des Elektronischen Handels

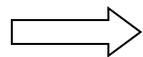
- "Less is more"**
- Geschichtliche Entwicklung**
- "Abzählbares" in Smalltalk**
- Klassensystem von Smalltalk**

Über das Klassensystem von Smalltalk (I.)

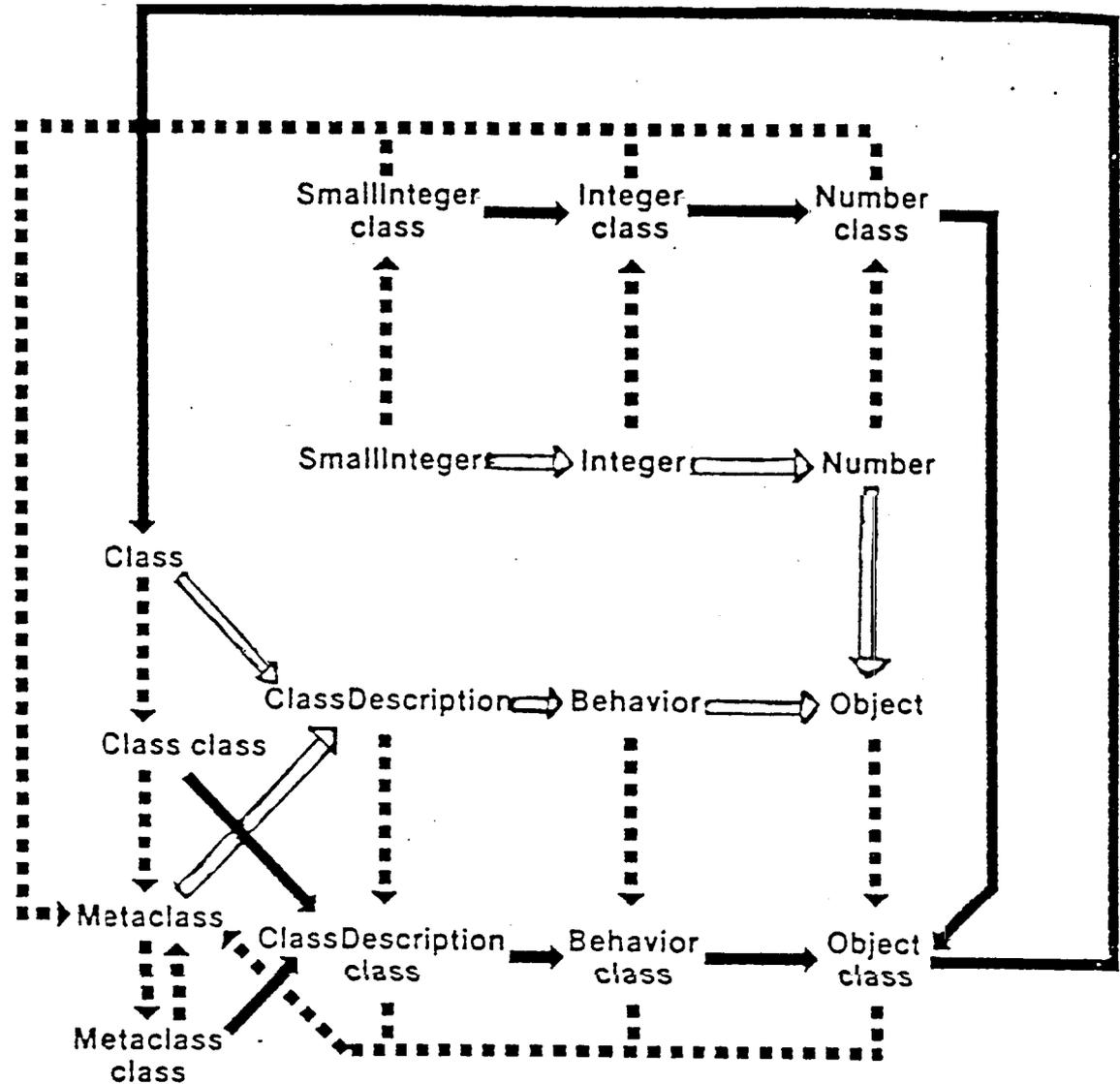
Keine Angst !

Alles sind **Klassen-
objekte** !

 Ausprägung

 Klassenhier.

 Metakl.hier.

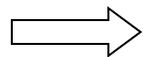


Über das Klassensystem von Smalltalk (II.)

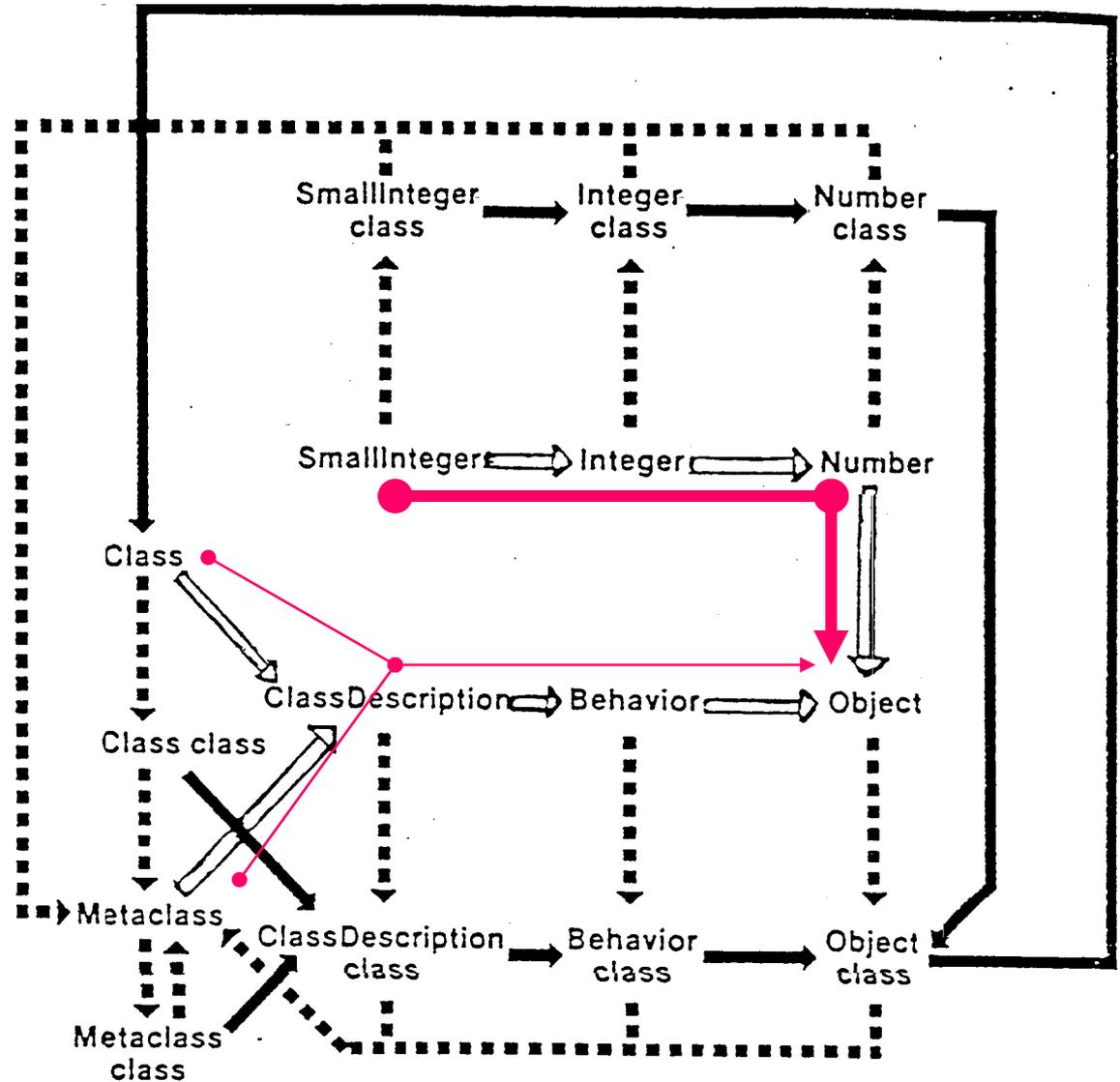
Was ist für den Programmierer von Interesse ?

Alles sind Klassenobjekte !

 Ausprägung

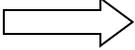
 Klassenhier.

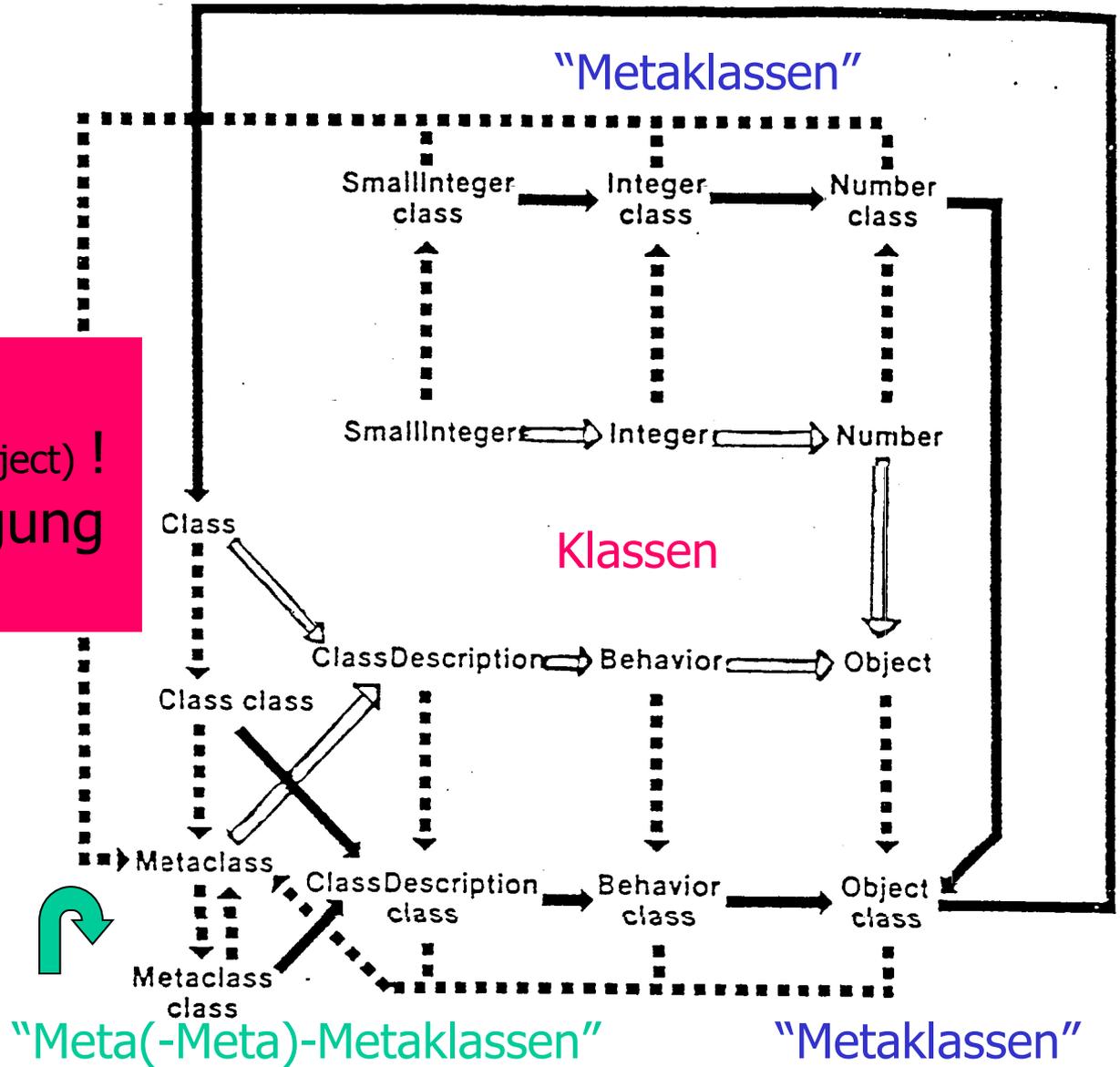
 Metakl.hier.



Über das Klassensystem von Smalltalk (III.)

Jede Klasse hat eine Oberklasse (Ausnahme Object) !
 ... Ist einzige Ausprägung ihrer *Metaklasse* !

- Alles sind **Klassen-objekte** !
-  Ausprägung
-  Klassenhier.
-  Metakl.hier.

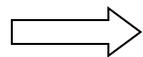


Über das Klassensystem von Smalltalk (IV.)

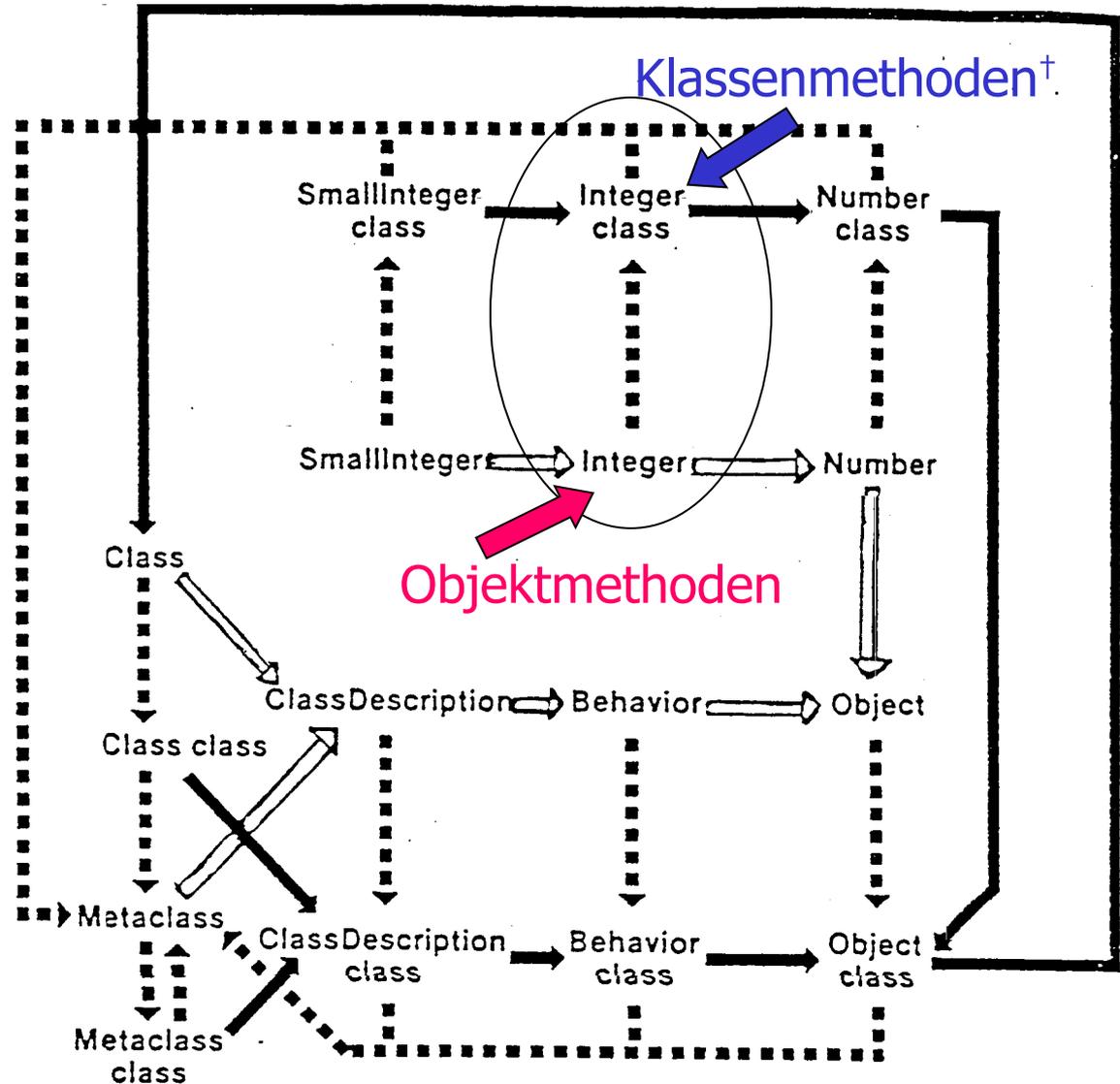
Alle Klassenobjekte verwalten Methoden für ihre Ausprägungen

Alles sind **Klassenobjekte** !

 Ausprägung

 Klassenhier.

 Metakl.hier.



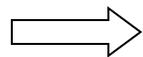
†Typischerweise zum Ausprägen von Objekten der jeweiligen Klasse

Über das Klassensystem von Smalltalk (V.)

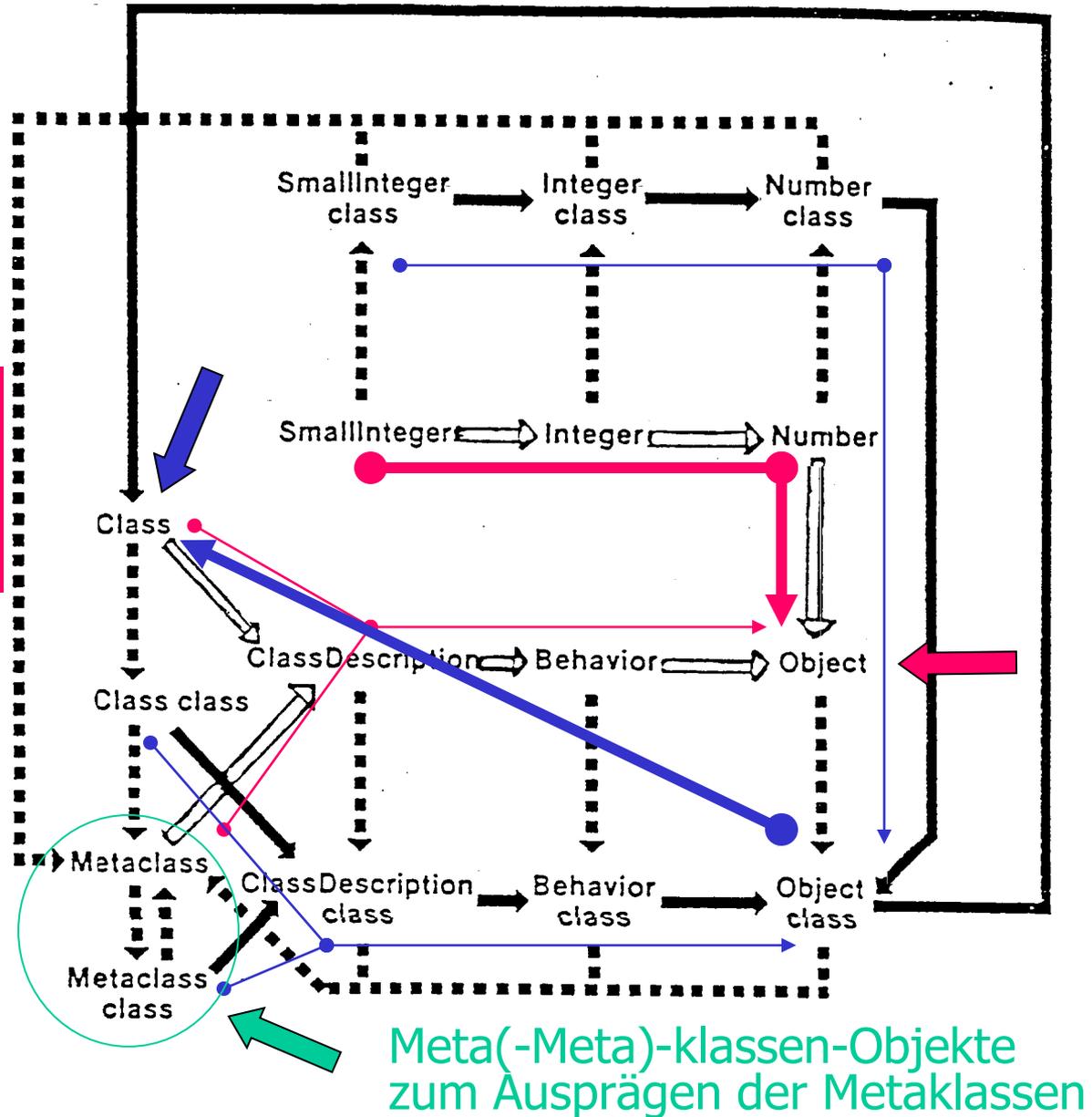
Hierarchie der
Klassen und Meta-
klassen verschieden

Alles sind **Klassen-**
objekte !

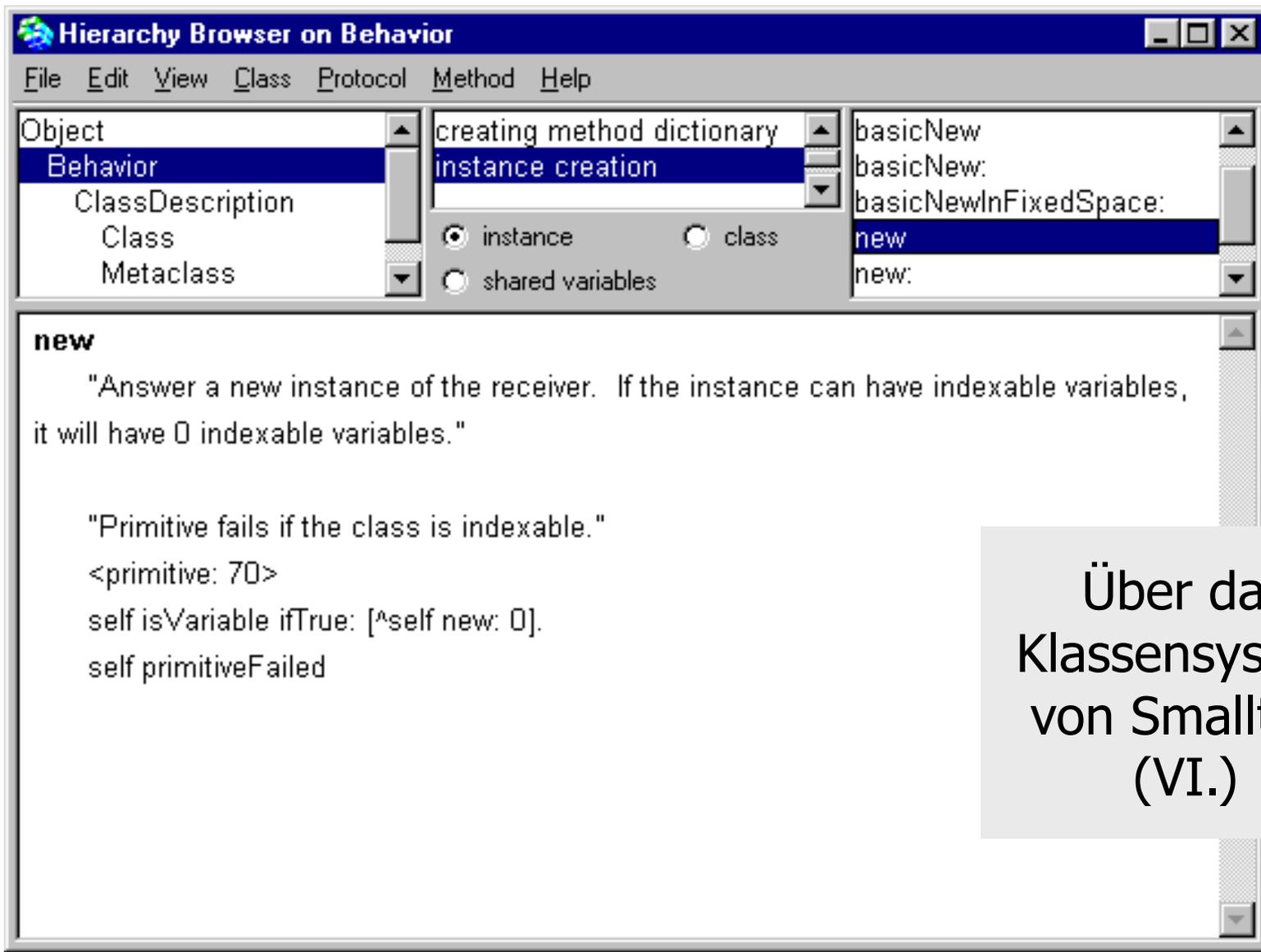
 Ausprägung

 Klassenhier.

 Metakl.hier.

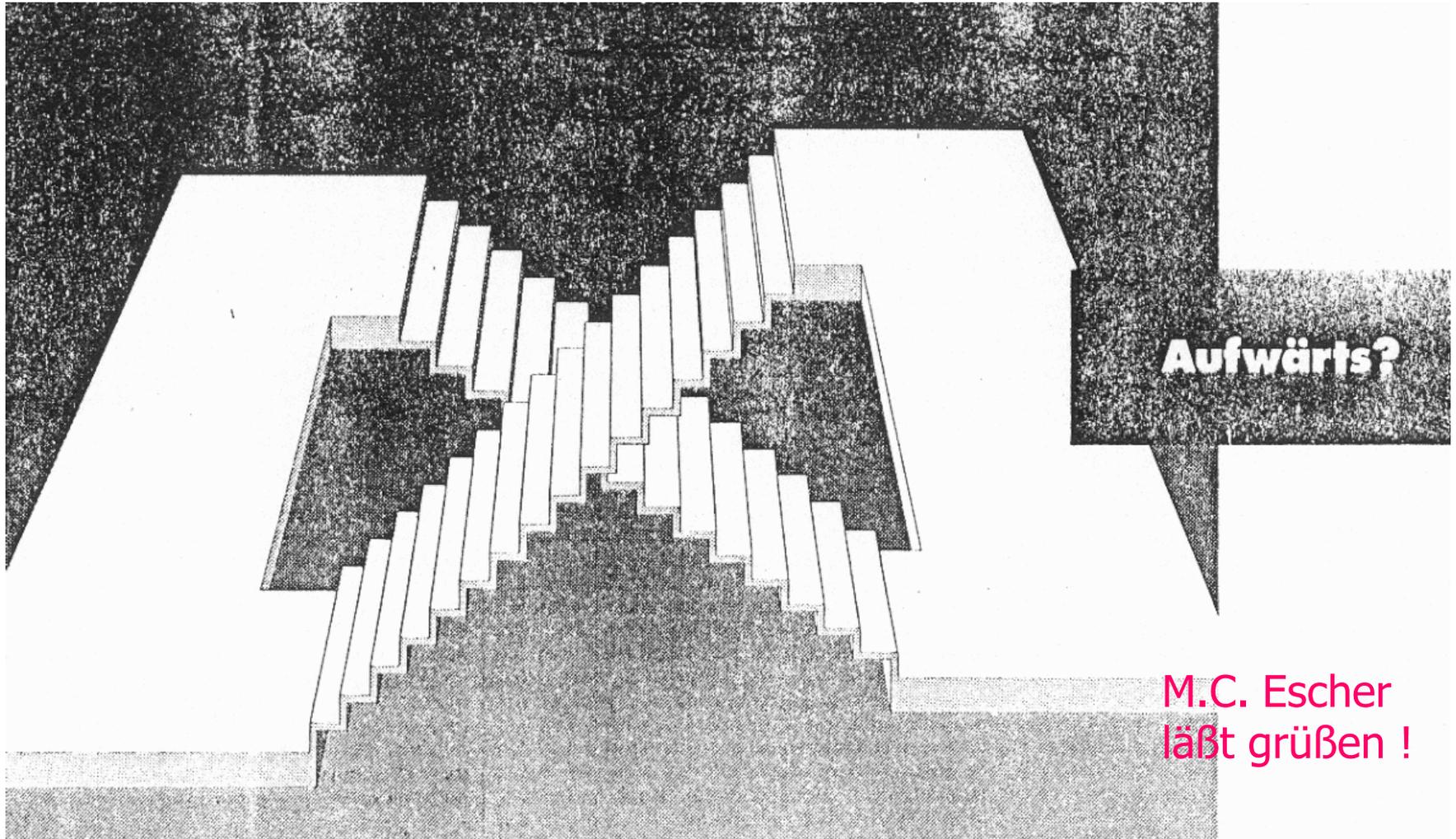


Meta(-Meta)-klassen-Objekte
zum Ausprägen der Metaklassen



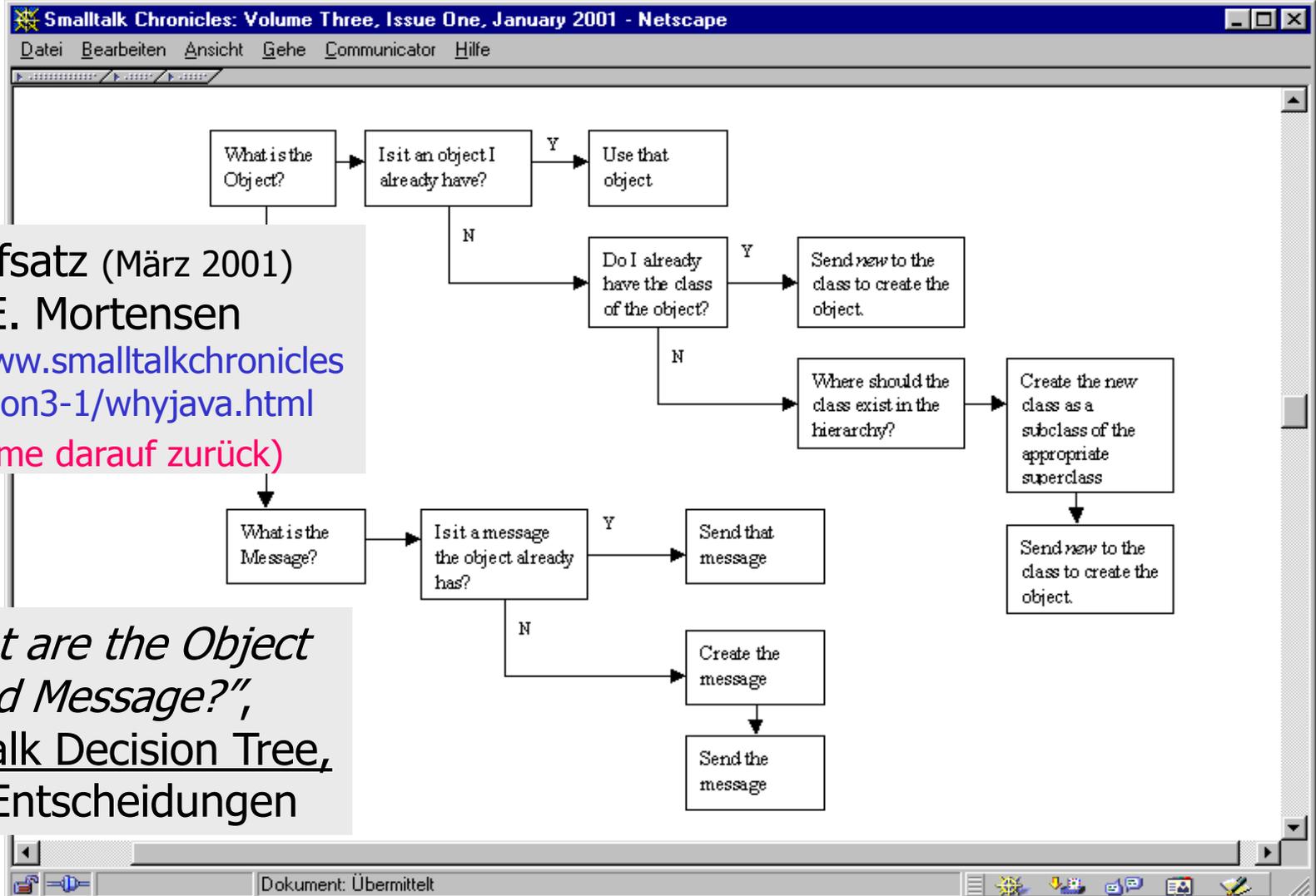
Über das
Klassensystem
von Smalltalk
(VI.)

Über das Klassensystem von Smalltalk (VII.)



M.C. Escher
läßt grüßen !

Zum Vorgehen beim Programmieren in Smalltalk (I.)



Aus Aufsatz (März 2001)
von S.E. Mortensen
<http://www.smalltalkchronicles.net/edition3-1/whyjava.html>
(ich komme darauf zurück)

"What are the Object and Message?",
Smalltalk Decision Tree,
drei Entscheidungen

Zum Vorgehen beim Programmieren in Smalltalk (II.)

Ein Zitat (P.W. Lount, consultant in USA, Entwerfer bekannter Computer-Spiele, peter@smalltalk.org):

"Why Smalltalk? Smalltalk uses a simplified version of English, nouns and verbs.

Objects are the nouns.

Verbs are messages sent from one object to the next, easy as 1, 2, 3.

It is no longer necessary to write cryptic programs. This means that almost anyone can learn to write Smalltalk programs."

Smalltalk ***in B2B-Anwendungen*** ***des Elektronischen Handels***

- ***"Less is more"***
- **Geschichtliche Entwicklung**
- ***"Abzählbares"* in Smalltalk**
- **Klassensystem von Smalltalk**
- ***"bytecode"* und *"virtual machine"***

Über "*bytecode*" und die VM ("*virtual machine*") von Smalltalk (I.)

- 0-Adressoperationen ("*bytecode*", #176 - #207 und #135 - #137 in Liste) auf Operandenkeller, z.B. für primitive arithmetische Operationen
- Spezielle *Push*- bzw. *Pop-Operationen* (#0 - #125) für bestimmte Objekte (ja, "*Objekte*" !)
- Operationen zur normalen Adressierung (2 bzw. 3 Byte, #128 - #134)
- Sprungoperationen (#144 - #175)
- Anstoß für "Botschaftenaustausch"
(in Smalltalk verpönt, es so zu sagen: Unterprogrammaufruf; #176 - #255)

Eigentlich ziemlich konventionell, wenn "eingebauter" Operandenkeller als selbstverständlich angesehen wird !

(beschrieben, wie im ursprünglichen Smalltalk-80 vorgesehen)

Kenner mögen dies mit dem Java-Bytecode vergleichen !

Über "*bytecode*" und die VM ("*virtual machine*") von Smalltalk (II.)

- Bedeutung der 0-Adressoperationen #176 - #207

The 32 special message selectors are listed below.

+	-	<	>
< =	> =	=	~ =
*	/	\	@
bitShift:	\\	bitAnd:	bitOr:
(at:)	(at:put:)	(size)	(next)
(nextPut:)	(atEnd)	==	class
blockCopy:	value	value:	(do:)
(new)	(new:)	(x)	(y)

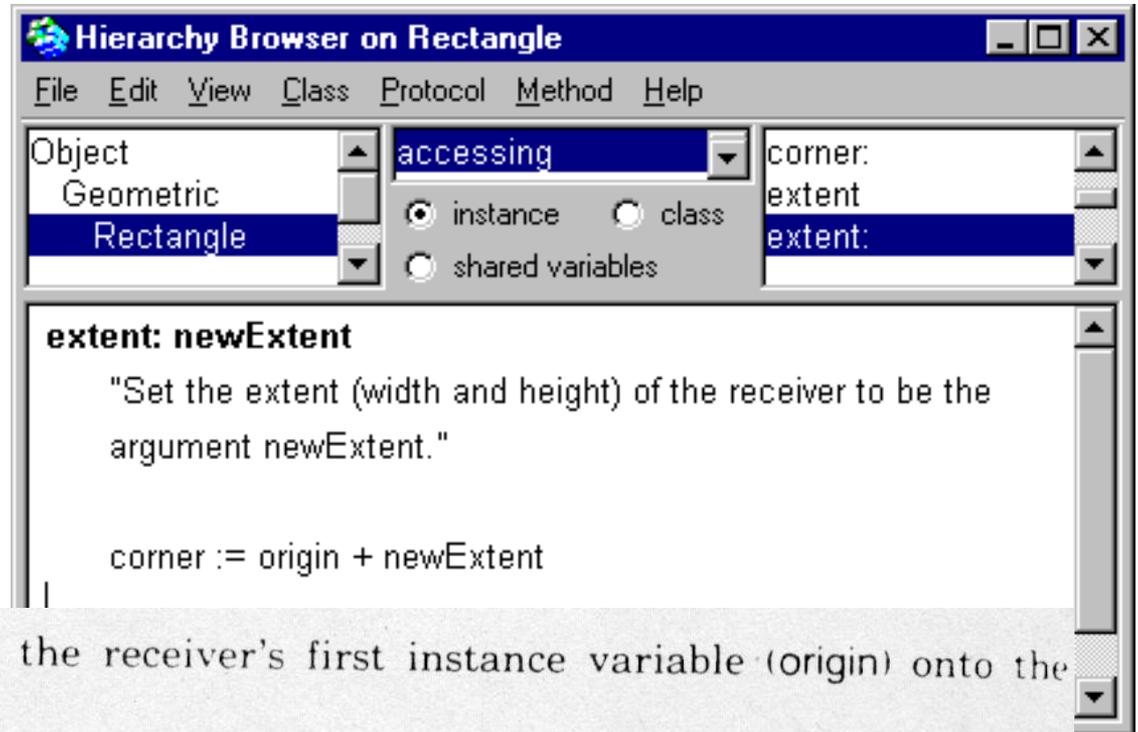
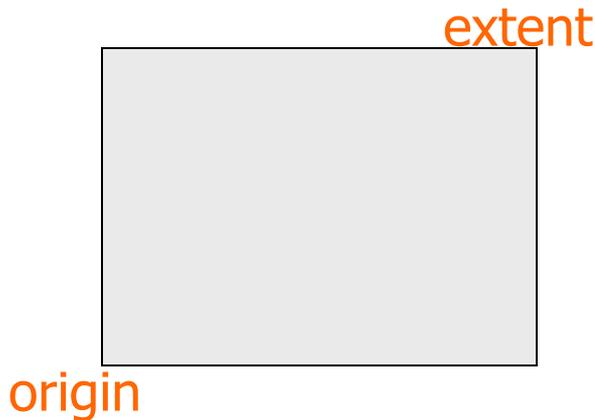
The selectors in parentheses may be replaced with other selectors by modifying the compiler and recompiling all methods in the system. The other selectors are built into the virtual machine.

Zur genauen Erklärung im Kurs, morgen und übermorgen !

(*x* bzw. *y* sind rechtwinklige Koordinaten eines Punkts bei Geometrieaufgaben !)

Über "bytecode" und die VM ("virtual machine") von Smalltalk (III.)

- Beispiel aus der Geometrie



```
0      push the value of the receiver's first instance variable (origin) onto the stack  
16     push the argument (newExtent) onto the stack  
176   send a binary message with the selector +  
97    pop the top object off of the stack and store it in the receiver's second instance variable (corner)  
120   return the receiver as the value of the message (extent:)
```

Auch bei Punkten,
Vektoraddition !

Länge 5 Byte !

Über "*bytecode*" und die VM ("*virtual machine*") von Smalltalk (IV.)

- (Der übrigens in gewissem Umfang strukturgebundene) **Methodeneditor des Smalltalk-Stöberers** übersetzt (!) den Methodentext in den Smalltalk-Bytecode
- Im "Zielcode" wird OO-Technologie **voll** eingesetzt, z.B. für Aufrufkeller, Klassen- und Objektrahmen, Codesegmente u.ä. (siehe spätere Folie)
- Die **VM** ist eine "Maschine" mit einem Halbenspeicher für Objekte (alles sind Objekte !), einem Aufrufkeller mit jeweils (u.a.) einem zu jeder Aufrufkontour zugeordneten Operandenkeller
- Der Objektzugriff (zu allen Objekten außer den ganzen Zahlen, siehe früher) erfolgt indirekt über eine "**Objekttafel**" (daher ist **Speicherbereinigung** und eventl. **Speicherkomprimierung** "leicht", auch während des Arbeitens der VM !)

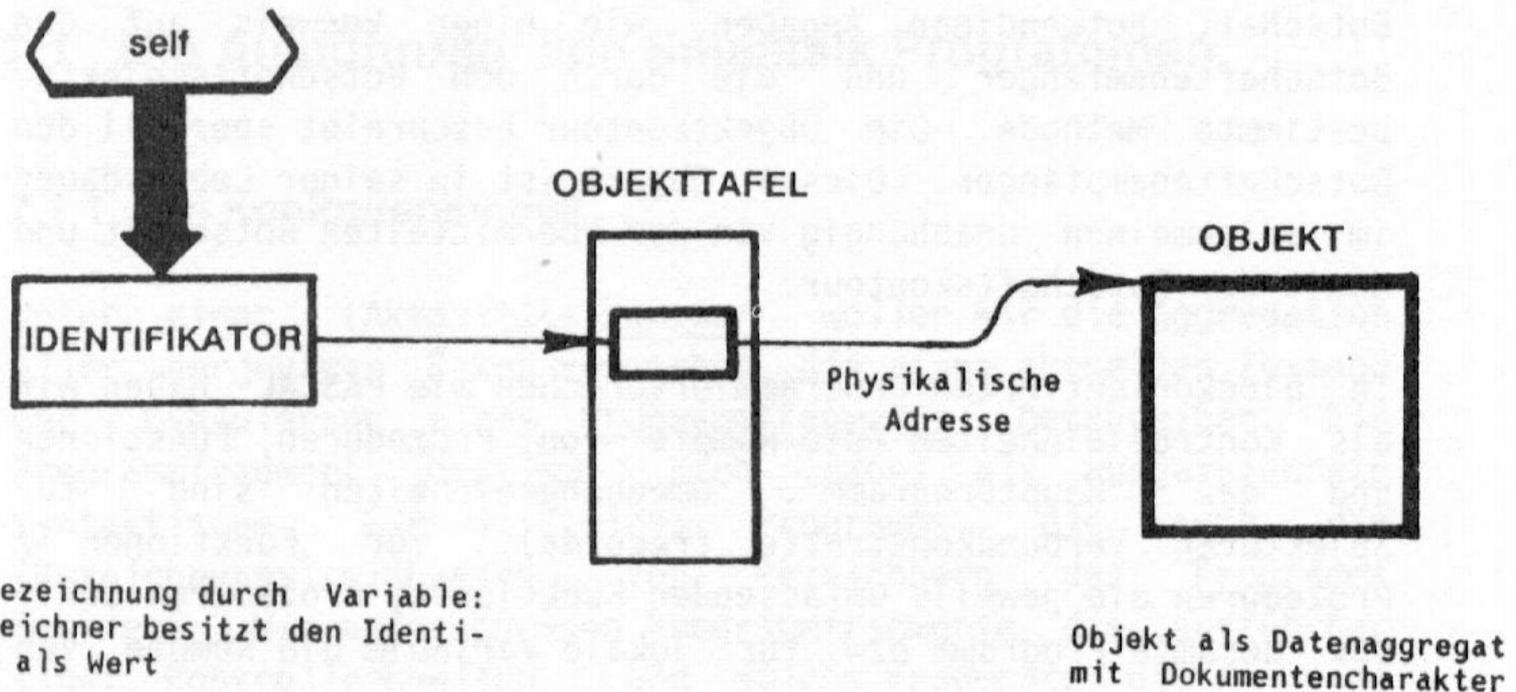
Über "*bytecode*" und die VM ("*virtual machine*") von Smalltalk (V.)

- Die VM kann auf einem realen Rechner als **Interpreter** realisiert sein (fast jeder kann so etwas programmieren !)
- Der VM kann ein "**Backend**" nachgestellt sein, das bei jedem Beginn eines Aufrufs des Smalltalk-Programms für alle Methoden als Codegenerator für eine reale Maschine wirkt
- Beim Aufruf einer Methode wird überprüft, ob diese nur in der Bytecode-Form vorliegt; wenn ja, wird durch das "Backend" für diese Methode Code generiert **und** solange aufbewahrt und wieder verwendet, bis der Smalltalk-Text der Methode geändert wird ("*just-in-time compilation*")

Skizzen auf den nächsten beiden Folien dienen zum Verständnis, sie spiegeln die Realität nicht voll wider !

Über "bytecode" und die VM ("virtual machine") von Smalltalk (VI.)

- Indirekter Zugriff auf Objekte (ziemlich stark vereinfacht):



- da Einträge in der Objekttafel (mindestens) 2 Byte umfassen, kommen keine ungeraden Adressen vor ==> interne Darstellung von *Smallinteger*!

Über "bytecode" und die VM ("virtual machine") von Smalltalk (VII.)

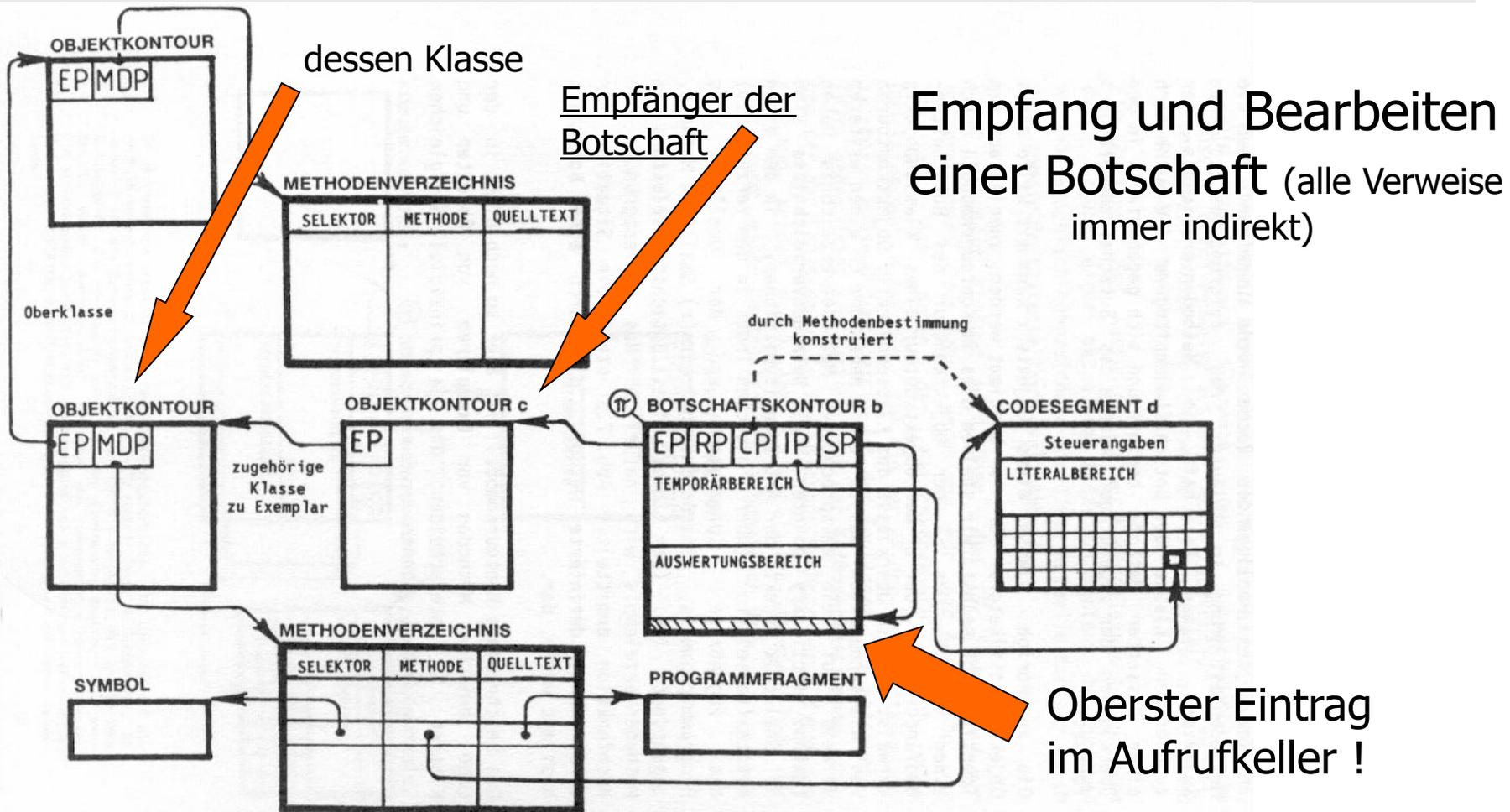


Abb. 7.3 Schema eines Botenschaftenaustauschs II

Kontouren für die Evaluierung einer Botschaft, die eine lokal definierte Methode durch ihren Selektor bestimmte.

Smalltalk in B2B-Anwendungen des Elektronischen Handels

- ***"Less is more"***
- **Geschichtliche Entwicklung**
- ***"Abzählbares"* in Smalltalk**
- **Klassensystem von Smalltalk**
- ***"bytecode"* und *"virtual machine"***
- **Smalltalk und Software-Engineering**

Smalltalk und Software-Engineering (I.)

- **Typisierung** (Vorbild *Pascal*)
 - Ist strenge Typisierung immer erstrebenswert? Modellbildung als Abbild der Realität kennt Variation über Zeit und / oder Betrachtungserfordernis
- **"Contracting"** (Vorbild *Eiffel*)
 - siehe M. Prasse: "*Design by contract*"- *Eine Implementation für Smalltalk*; 4. Fachkongreß STJA 98, 1998, 225 - 237
- **Entwurfsmuster**
 - siehe S.R. Alpert et al.: *The design patterns Smalltalk companion*; Addison Wesley, 1998
- **Anwendungsrahmen** (engl. *frameworks*; Vorbild *FP / FFP* -?-)
 - ein vielfach beobachteter Erfolg bei Smalltalk aufgrund Typlosigkeit
- **Komponenten** (Vorbild *Java* -?--; "*Software IC's*" von **B.J. Cox** -?-)
 - schon früh (/früher) in Smalltalk-Varianten verwendet, z.B. *parts*

Smalltalk und Software-Engineering II.

Was sind die Merkmale einer guten oder vollkommenen objektorientierten SW ?

Ein Kommentar zum industriellen Einsatz populärer Programmiersprachen

- H. Schröder, SQL Datenbanksysteme GmbH, Berlin

Softwaretechnik-Trends, 14 (1994) 3, 24 - 28. Daraus zwei Zitate:

- *"Es existiert da ein gewisser Herdentrieb: Was die anderen machen, mache ich auch."*
- *"Wenn man es genau nimmt - Stichwort 'vollkommen' - gibt es nur zwei Sprachen, die im Einsatz Relevanz besitzen: Eiffel und Smalltalk, denn beide sind 'echte' OO-Sprachen, mit denen sich also auch 'echte' OO-Programme erstellen lassen, und beide sind kommerziell verfügbar"*

Ist dem etwas hinzuzufügen ?

Ausbildung für und mit (oo &) Smalltalk (I.)

- Fachgespräch anlässlich der Net.OBJECTDays 2001
Lehre objektorientierter Konzepte, Erfahrungsberichte aus der Ausbildung
Tagungsband NET.OBJECTDays 2001, 327 ff.
- A. Goldberg: *What should we learn? / What should we teach?*
- Learning Works; STIA 1995 (auch OOPS Messenger 6 (1995) 4, 30 - 37; u.a.)
- Viele deutsche / englische Lehrbücher
- L. Schmitz: *Syntaxbasierte Programmierwerkzeuge*; Teubner, 1995

C++ ? Java ? C# ? Eiffel ? Oberon ? Beta ? Smalltalk ?

Wo finden sich Grundlagen, was wird überdauern ?

Welche Sprache / welches System regt zu übedem Lernen an ?

Einsicht in alle wesentlichen Aspekte ? Einheitlichkeit ?

"Less is more"

Ausbildung für und mit (oo &) Smalltalk (II.)

Studie von IDC, #9818, März 1995:

"Object Tools - Smalltalk market accelerates"

Wieviele Monate Entwicklungserfahrung müssen aufgewendet werden, um zu einem Expertenwissen in einer Programmiersprache zu gelangen ?

4GLs	17.1 Monate	
Smalltalk	18.2 "	(!)
C++	23.2 "	
COBOL	28.1 "	

Smalltalk ***in B2B-Anwendungen*** ***des Elektronischen Handels***

- ***"Less is more"***
- **Geschichtliche Entwicklung**
- ***"Abzählbares"*** in Smalltalk
- **Klassensystem von Smalltalk**
- ***"bytecode"*** und ***"virtual machine"***
- **Smalltalk und Software-Engineering**
- ***"Zeitgenossen"*** und ***"Nachahmer"***

Gibt es "Zeitgenossen" / "Nachahmer"?

- Was meinen Sie ? Ja, einige:

C++

vor 1981

B. Stroustrup: Classes - An abstract data type facility for the C-language; ACM SIGPLAN Notices 17(1982), 1, 42 - 51

Objective-C

um 1985

B. J. Cox: Object oriented programming - an evolutionary approach; Addison-Wesley, 1986

Eiffel

um 1985

B. Meyer: Object-oriented software construction; Prentice Hall Intl., 1988

Java

1992-95

J. Gosling et al. SUN Corp

C#

2000

NN (E. Gunnerson et al.) Microsoft Corp

- Und noch einige weniger Bekannte / weniger Bedeutungsvolle

Gibt es "*Zeitgenossen*" / "*Nachahmer*"?

C++ (I.)

- Parallelentwicklung mit (m.E.) geringerer "Erfindungshöhe"
- Hybrider Sprachansatz,
"Jedes C-Programm ist auch ein C++-Programm"
- Sauberes Vorgehensmodell bei der Programmierung fehlt
- Großer (überladener) Vorrat an möglichen Programmkonstrukten
- Mehrfaches Erben möglich
- (fehleranfällige) Speicherungsverwaltung
- Unsichere Referenzierung in Datenstrukturen (*Pointer*)
- Unsichere Typisierung
- Mitläufer der UNIX-Verbreitung (!)

Ein Buchumschlag mit
einem Heißluftballon
hat es auch der C++-
Gemeinde angetan
(2001) !



J. Sutherland: *The Bottom Line*; Object Magazine 2(1995) 5, 32-35

So what is the bottom line on **C++ versus OO COBOL versus Smalltalk for MIS applications development**? The table below shows my ratings. **One is good, three is bad**, and two is ugly. Remember these numbers are for MIS applications, not for systems level programming.

		C++	OO COBOL	Smalltalk
Flexibility	Dynamic Binding	2	2	1
	Dynamic Classes	3	1	1
	Multiple inheritance	2	2	3
	Support Roles	3	3	2
Ease of use	Class libraries	3	3	1
	Learning Curve	2	1	1
	Speed of development	3	2	1
	Portability	3	3	1
Support	Tools	1	3	1
	Multiple vendors	1	3	2
Performance	Performance	1 (experts)	3	2
	Garbage collection	3	1	1
Minimizing Risk	No memory leaks	3	1	1
	No overwriting memory	3	1	1

Gibt es "Zeitgenossen" / "Nachahmer"?

C++ (II.)

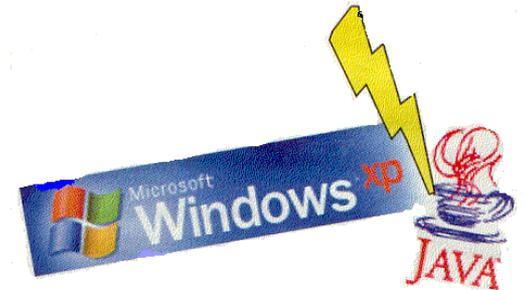
Ein Zitat (Bjarne Stroustrup: *Getting from the Past to the Future*; C++ Report, 11(1999) 10, 23 pp):

"The main problem for the C++ community today is to use Standard C++ in the way it was intended rather than as a glorified C or a poor man's Smalltalk."

Gibt es "Zeitgenossen" / "Nachahmer"?

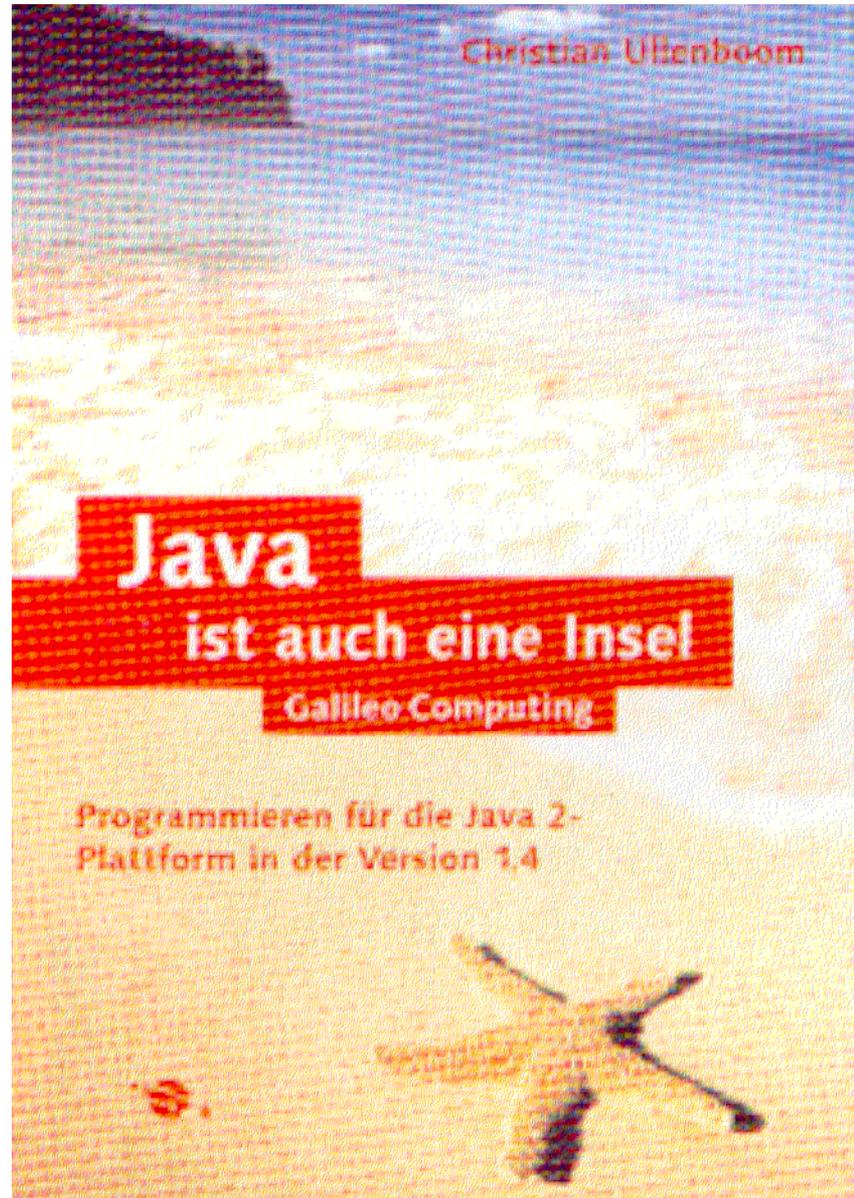
Java (I.)

- (Aus Vorläufern heraus) überlegte Neuentwicklung
- Reiner Ansatz zur Objektorientierung; VM-Lösung
- Vermeiden bekanntgewordener Probleme von C++ (u.a. keine Referenzierung mit Pointern, geschicktere Form beim mehrfachen Erben, strenge Typisierung)
- Gesicherte Speicherungsverwaltung einschl. -bereinigung
- (Schädliche) Konkurrenzstreitigkeiten und daraus Unverträglichkeiten
- Mitläufer der Verbreitung des WWW (!)
- Versprechungen (z.B. hinsichtlich Sicherheit) nicht voll eingehalten



Ein anderer Buchumschlag
mit Seestern statt eines
Heißluftballons; die "Insel
der Seligkeit" unerreichbar
(2001/2002 angekündigt;
"Bescheidenheit ist eine Zier ...") !

Wohl auf Kurzlebigkeit
angelegt == =>



Java - Two steps forward, one step back

Java is a wierd situation. It started as a powerful force from Sun's marketing department. It was the first language I have ever seen break the popularity barrier without the usual decade of acadamia and research use. For that matter, Java got popular before it was even used by anyone. Guess that is what happens when product managers are allowed to pick implementation languages from the back of a magazine.

Java fixes a lot of the frustrations of C++, but doesn't go far enough in my opinion. They didn't take enough from Smalltalk, even though they took quite a bit from it.

Good Garbage Collection
Compiling to bytecode (cross platform binaries)
Standard class libraries (from the start)

Bad Low level types (non-objects)
No first class objects, reflection mechanism is a joke.
No operator overloading.

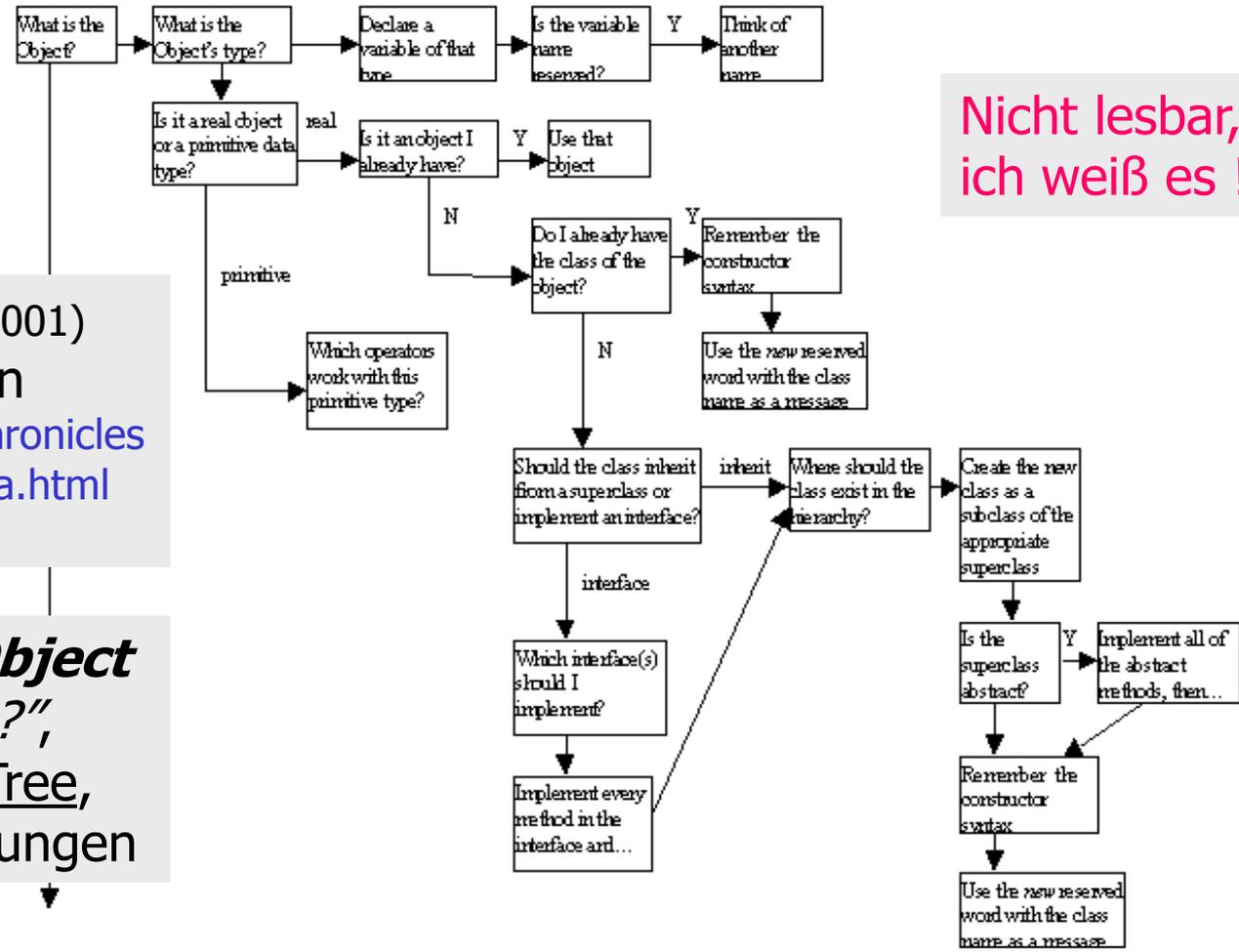
"More matter, with less art" - Gertrude, Hamlet

Copyright © 1997-2002 Ryan Davis & Zen Spider Software. All Rights Reserved.

Generated: 2002-02-16

<http://www.zenspider.com/Languages/Java/index.html>

Zum Vorgehen beim Programmieren in Java (I.)



Nicht lesbar, ich weiß es !

Aus Aufsatz (März 2001) von S.E. Mortensen <http://www.smalltalkchronicles.net/edition3-1/whyjava.html> (vgl. bei Smalltalk !)

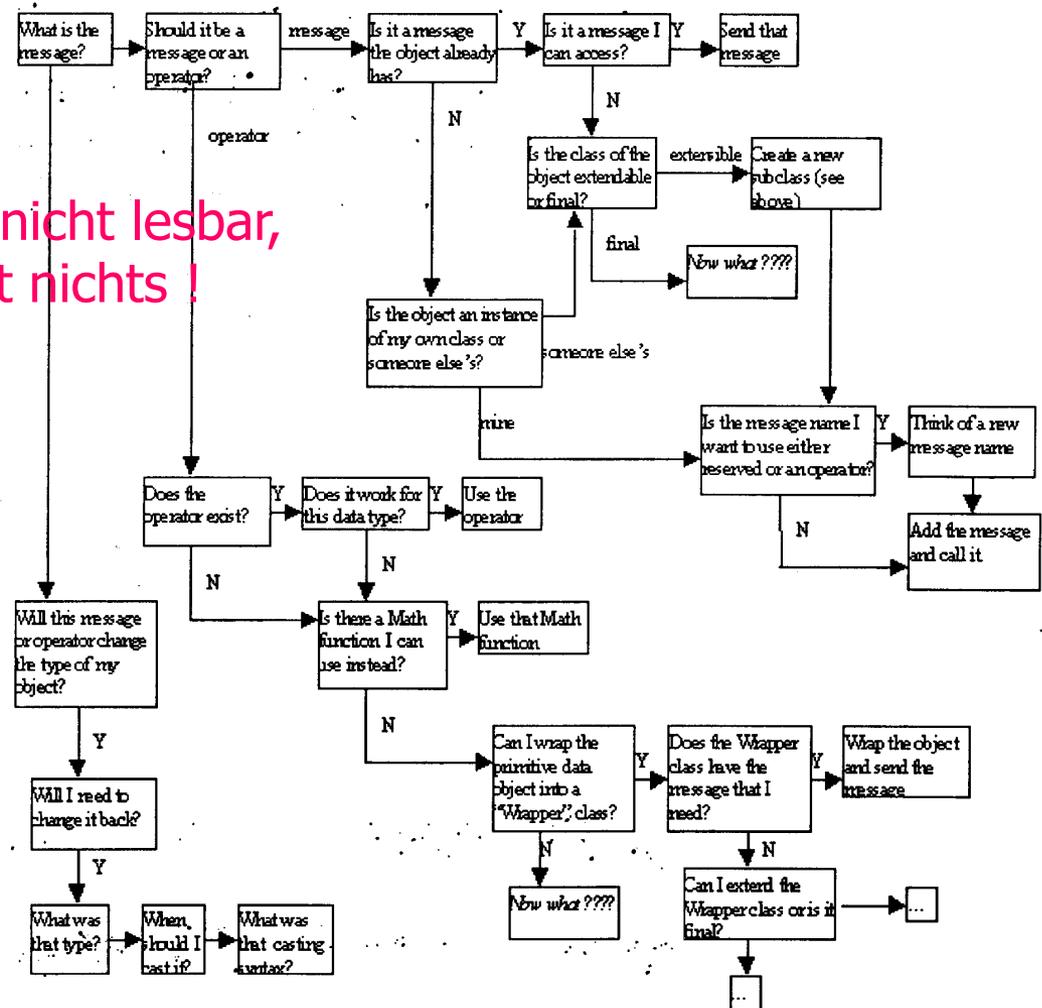
*"What are the **Object and Message?**", Java Decision Tree, sieben Entscheidungen*

Zum Vorgehen beim Programmieren in Java (II.)

Aus Aufsatz (März 2001)
von S.E. Mortensen
<http://www.smalltalkchronicles.net/edition3-1/whyjava.html>
(vgl. bei Smalltalk !)

*"What are the Object
and **Message**?"*,
Java Decision Tree,
>14 Entscheidungen

Auch nicht lesbar,
macht nichts !



Gibt es "*Zeitgenossen*" / "*Nachahmer*"? Java (II.)

Ein Zitat (L. Nicolle: *The Bulletin Interview 'James Gosling'*;
The Computer Bulletin, March 1998, 16 - 17):

"... 'Making technology invisible is extremely important. The people using computers have something else they're trying to do. ...'"

Gibt es "Zeitgenossen" / "Nachahmer"?

Java (III.)

Wer jetzt immer noch an Java glaubt, dem empfehle ich:

Giovanni Giorgi:

Smalltalk Tutorial for Java programmers,

http://objectsroot.com/squeak/squeak_tutorial.html

oder

http://daitanmarks.sourceforge.net/or/squeak/squeak_tutorial.html

(von einem wohl ganz jungen Italiener mit viel Charme und Engagement geschrieben !)

Gibt es "Zeitgenossen" / "Nachahmer"?

Java (IV.)

- **Smalltalk --> Java**

(scheint wirtschaftlich nicht uninteressant zu sein !)

Produkte von

- Gebit GmbH
- ObjectART Software GmbH (auch *Dolphin Smalltalk* !)
- weitere ?

- **Java --> Smalltalk**

(warum eigentlich nicht auch ?)

- Produkte ?

- **Smalltalk + Java / Java + Smalltalk**

- except AG (mit *Smalltalk X*, gemeinsame VM)

- Und jetzt noch (im Vorgriff): **.NET bzw. C# <--> Smalltalk**

- Produkte ? (*Smallscript* von QKS)

Gibt es "Zeitgenossen" / "Nachahmer"?

C#

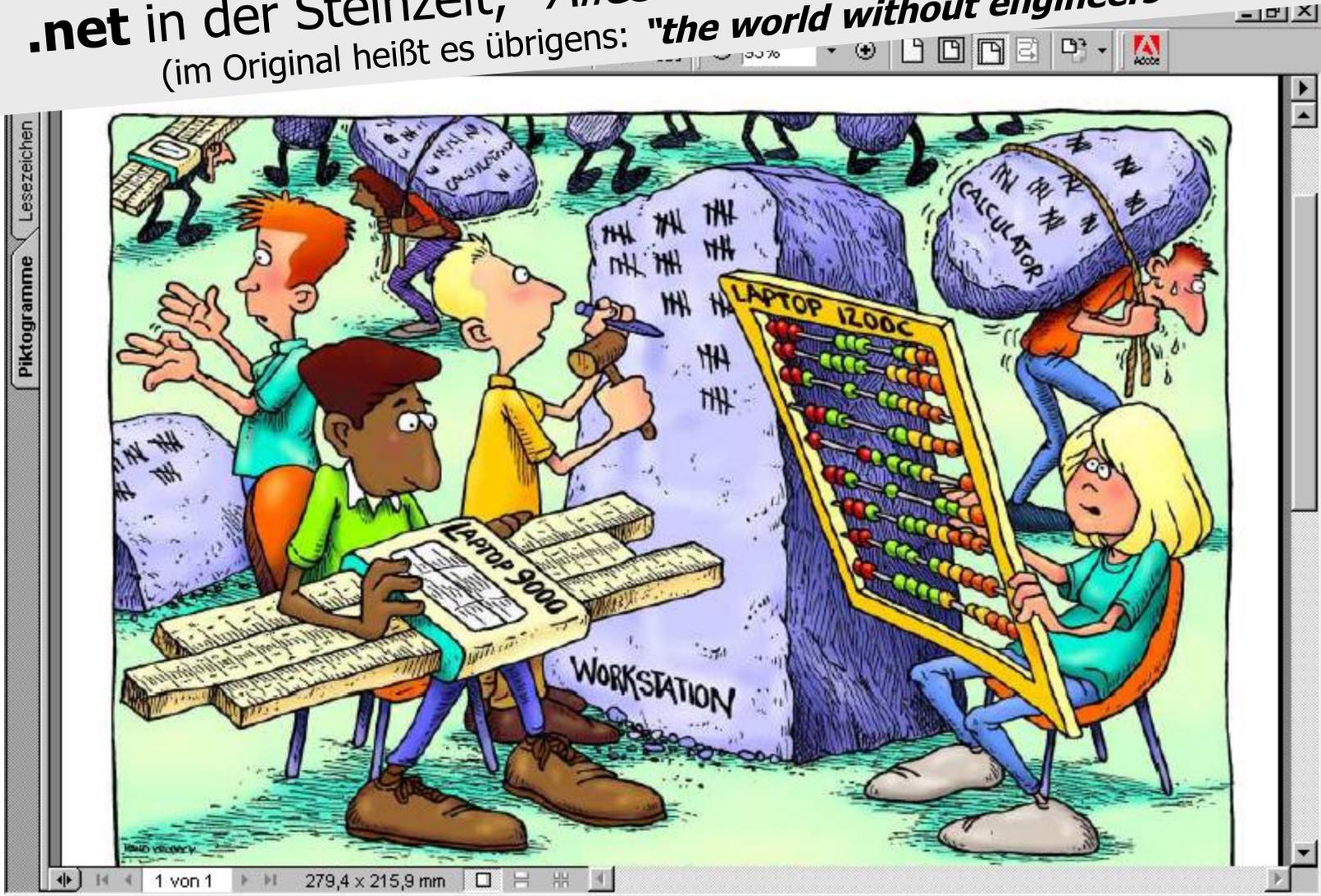
- (Aus rechtlichen und Konkurrenzgründen) Neuaufguß von C++ und Java; proprietär
- (Wesentlicher) Bestandteil der ".net"-Software-Architektur der Microsoft Corp.
- Vorgestellt als das "Gelbe vom Ei" in der Entwicklung von Programmiersprachen (ich habe schon mehrere solche hochgegriffenen Ankündigungen miterlebt und bin daher skeptisch)
- Ist es eine "eierlegende Wolfsmilchsau" / "Totgeburt" ?
- M.E. wird dabei eine weitere der auch bei anderen Programmiersprachen zu findenden, fragwürdigen Qualität im Stil von PL/I, Algol 68, Ada, C herauskommen

Hat die Doping-Kontrolle
für den Siegerpreis schon
stattgefunden
(2001/02 angekündigt;
m.E. "Vorschußlorbeeren") !

(Ich bin "weder verwandt noch
verschwägert" mit dem Verlag
Galileo Computing und möchte auch
keine Schleichwerbung machen !)



.net in der Steinzeit, "Alles schon einmal dagewesen!"
(im Original heißt es übrigens: "the world without engineers")



Die Nachahmer"als "Konkurrenten"

Open Source Directory - Nennungen (Stand April 2002, Auswahl)

http://dmoz.org/Computers/Programming/Languages/Open_Source/

	Java	1939
	C++	636
<u>Ada</u>	387	
<u>Lisp</u>	318	
<u>Basic</u>	260	
	Smalltalk	187 (einschl. Squeak +69 = 256)
<u>C</u>	180	
<u>Pascal</u>	71	
<u>Oberon</u>	65	
<u>Eiffel</u>	46	
	C#	?
<u>Beta</u>	(nicht erwähnt !)	
<u>Prolog</u> & <u>Simula</u>	(vergessen !)	

Smalltalk ***in B2B-Anwendungen*** ***des Elektronischen Handels***

- ***"Less is more"***
- **Geschichtliche Entwicklung**
- ***"Abzählbares"* in Smalltalk**
- **Klassensystem von Smalltalk**
- ***"bytecode"* und *"virtual machine"***
- **Smalltalk und Software-Engineering**
- ***"Zeitgenossen"* und *"Nachahmer"***
- **Ein (hoffentlich) guter Schluß**

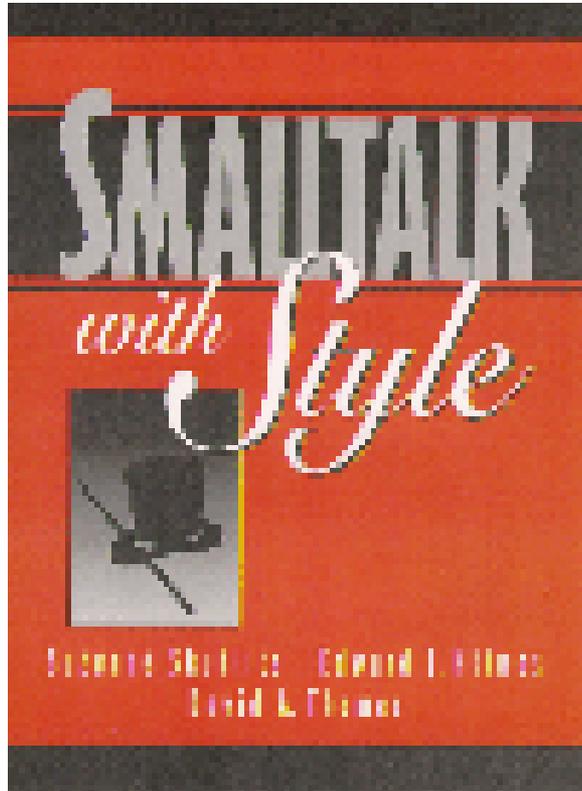
Literatur / weitere Information zu Smalltalk (I.)

- Die "Goldberg-Bücher" (Klassiker): alle Addison Wesley, 1983
 - Smalltalk-80: The Interactive Programming Environment*
 - Smalltalk-80: The Language and its Implementation* † <== gesucht !
 - Smalltalk-80: Bits of History, Words of Advice*
 - Smalltalk-80: The Language* (1989)
- Viele gute (und schlechte) Lehrbücher in Deutsch & Englisch
- "Smalltalk Solutions"-Konferenzserie in USA (April)
- "Camp Smalltalk" in USA (Juni) und in Europa (August)
- "NET.OBJECTDays"-Konferenzserie in Deutschland (Sept./Okt.)
- "OOP"-Serie in Deutschland (Januar)
- <http://www.gsug.org/> ==> German Smalltalk User Group
- <http://www.esug.org/> ==> European Smalltalk User Group
- <http://www.whysmalltalk.com/>
- <http://st-www.cs.uiuc.edu/> ==> Smalltalk Archive †2. Auflage, 1989

<== (bieten *Smalltalk CD* an !)

Literatur / weitere Information zu Smalltalk (II.)

Smalltalk with Style



Suzanne Skublics, OTI

Edward Klimas, Linea Engineering

David Thomas, OTI

Prentice-Hall, 127 pp, 1996

ISBN 0-13-165549-3

3 of 3 people found the following **review** helpful:

A definitive "**Must Have**", April 28, 2000

Reviewer: GO SMALLTALK!

This is one of two or three "must have" books that every person interested in or practicing Smalltalk needs to read and keep on a shelf near by. This book will teach you everything you need to know about writing clear and concise Smalltalk code. The author, Ed Klimas, is one of the most well known Smalltalk guru's around.

Ich schlieÙe mich dem an !

Literatur / weitere Information zu Smalltalk (III.)

Und noch ein **"Must"**:

Adele Goldberg: Why Smalltalk?,
Comm. ACM 38 (1995) 10, 105 - 107

Ein Zitat (Seite 107):

"Many long-time programmers - business experts enmeshed in Cobol or Basic or Pascal - give testimony to being 'rejuvenated' by their Smalltalk experience."

Literatur / weitere Information zu Smalltalk (IV.)

- Smalltalk Webring <http://F.webring.com/hub?ring=smalltalk>
- Why Smalltalk ? <http://www.whysmalltalk.com/>
- comp.language.smalltalk (unter anderen)
- X3J20 Standard / Smalltalk ANSI-Standard /
Smalltalk Industry Council STIC
- mehr oder minder eingeschlafen ?
(*VisualWorks* and *VisualAge for Smalltalk* noch am ehesten dabei ?)
- Smalltalk als KI-Sprache / Smalltalk als *Script*-Sprache /
Smalltalk als funktionale Sprache ?
- es gibt viele Leute, die sich viel zu wenig mit Smalltalk beschäftigt haben, sonst kämen sie nicht zu solchen Fehleinschätzungen
(oder ist es der Versuch, Moden mitzumachen/mitzunehmen ?)

Literatur / weitere Information zu Smalltalk (V.)

Und noch ein besonderes **Schmankerl:**

Betreff: [gsug] ESUG actions to promote Smalltalk
Von: ducasse <ducasse@iam.unibe.ch>
An: <gsug@gsug.org>
Datum: 06.04.02 10:13 

Hi all, please distribute the following announce available at <http://www.esug.org>

Program 1: Smalltalk in Academy

ESUG would like to promote the use of Smalltalk in teaching and research institutions. Smalltalk is a pure object-oriented language that is mostly written in itself. It has good pedagogical properties. ...

Program 2: Cool Smalltalk Code Contest

ESUG would like to offer Smalltalk books to students. We propose to any world-wide students to participate to the Cool Smalltalk Code Contest. The principle is the following: you send to the ESUG committee a small (the quantity is not an issue here) but cool piece of code showing some interesting aspects of Smalltalk or a small application, then your piece of code is put freely on the web and every two months we select the one we prefer. ...

Smalltalk, heutige Bedeutung bei B2B-Anwendungen des Elektronischen Handels (I.)

- Smalltalk lebt !
- Smalltalk wird weltweit beachtet !
- Smalltalk hat viele Erfolge vorzuweisen (nicht nur bei einigen verschrobene Professoren und/oder lebensfremden Individualisten) !
- Smalltalk-Erfolge finden sich vielseitig im Wirtschaftsleben, bei Handel, Banken, Automobil- und Kommunikationsindustrie, Verkehrseinrichtungen ... !
Bei großen und kritischen Anwendungen !
- (Leider) wird oft nur unter vorgehaltener Hand über die Smalltalk-Erfolge berichtet ! (mit Java oder .net brüstet sich jeder)
- Die Management- und Marketing-Probleme bei Herstellern scheinen überwunden zu sein !

Smalltalk, heutige Bedeutung bei B2B-Anwendungen des Elektronischen Handels (II.)

Zwei Zitate (J. Tuchel: *Smalltalk in komplexen E-Business-Szenarien*; OBJEKTspektrum (2001) 2, 24 - 29):

"Einige produktive Beispiele im Internet beweisen eindrucksvoll, daß Smalltalk-Anwendungen sehr robuste E-Business-Plattformen sein können: Die Palette ... reicht vom Sportartikelhändler über Diskussionsforen bis hin zum Projekt-, Reisekosten- und Fuhrparkmanagement."

"Die Vorzüge, die Smalltalk gegenüber hybriden Sprachen bietet, sind also auch im Internet-Zeitalter gültig."

Smalltalk in B2B-Anwendungen des Elektronischen Handels Bekanntgewordene Anwendungen in Deutschland (u.a.)

Aufgefallen sind mir in letzter Zeit:

- **Hugo Boss AG** (siehe S. Weiblen, G. Eisenmann: *Hugo Boss: Mit Smalltalk und XML ins Web*; OBJEKTspektrum (2001) 4, 79 - 81)
e-Sales und Produktionsplanung
bäurer componentware GmbH
ERP CS (Enterprise Resource Planning Construction Set)
- **ClassWare GmbH**
HRWorks (<http://www.reisekosten.de>, <http://www.hrworks.de>,
<http://www.heise.de/ct/00/10/214/>)

siehe auch Auflistungen in **Why Smalltalk ?** >> "Produced in Smalltalk",
http://www.whysmalltalk.com/production/production_smalltalk.htm

und/oder bei **Monty Kamath's GoodStart**
<http://www.goodstart.com/whoswho.shtml>

Es sind neben weltweit bekannten Konzernen auch viele KMUs zu finden !

Smalltalk in B2B-Anwendungen des Elektronischen Handels

Warum ?



All das mit zwei, drei oder vier Buchstaben, wie es sich im Java-Umfeld findet ?

Was könnten/sollten wir erreichen ? Smalltalk in Anwendungen !



Nein !

ENDE