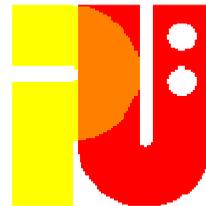


Programmierfehler / Programmfehler

Wie geht Software damit um ?

WS 2007/2008



Themen, betreut von
Univ.-Prof. em. Dr. H.-J. Hoffmann

(Stand 22. November 2007)

Die Literaturangaben in den einzelnen Themenblättern dienen als Anleitung zur jeweiligen Thematik. Die Bearbeiter müssen sich um einen darüber hinausgehenden Überblick bemühen.

Angaben noch unvollständig !

Alle Internet-Zugriffe September – Oktober 2007 !
Irrtum und Tippfehler vorbehalten !

Einige der angegebenen Artikel bzw. WWW-Seiten enthalten grammatikalische Fehler und Tippfehler !

Anleitung zur Vorbereitung Ihres Vortrags bzw. der Ausarbeitung:

- S.I.P. Jones et al.: *How to give a good research talk*,
ACM SIGPlan Notices 28 (1993) 11, 9 - 12
- M. Deininger et al.: *Studien-Arbeiten, ein Leitfaden ...*;
(u.a.) Teubner, 1992

**Beides kann in der Bibliothek des FB Informatik
eingesehen werden !**

Inhaltliche Vorbereitung für alle:

siehe de.wikipedia.org/wiki/Programmfehler

G.V. Zemanek: *Programmierfehler
als Chance und Herausforderung.*

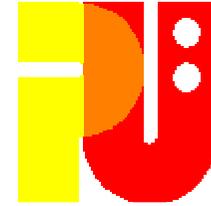
In: P. Molzberger, G.V. Zemanek (Hrsg.):
*Software-Entwicklung –
Kreativer Prozeß oder formales Problem?*

German Chapter of the ACM, Bericht # 22, B.G. Teubner,
1985, 78-102

A. Zeller: *Why programs fail* ; 2006, dpunkt.Verlag

- Einführung
 - #01 Was sind "gute" Programmierer und wie "gut" sind sie?
 - #02 Wie "gut" ist Software?
- Programmierfehler
 - #03 Fehlerarten, wie kommt es zu Fehlern
 - #04 Fehlererkennung
 - #05 Reparaturmaßnahmen bei "syntaktischen" Fehlern
 - #06 Reparaturmaßnahmen bei "semantischen" Fehlern
 - #07 Beispiele: Compiler *
 - #08 Beispiele: Editoren *
 - #09 Fehlermeldungen *

Übersicht



- Programmfehler
 - #10 Fehlbedienung
 - #11 Abfangen von Laufzeitfehlern im Programm
 - #12 Fehleranzeigen *
 - #13 "Debugger" *
 - #14 Beispiel: Smalltalk *
 - #15 Computeranwendung ohne Fehler (?) *

Mindestteilnehmerzahl 10. Es werden **maximal 12 Themen** ausgegeben !
 Die mit * markierten Themen stehen am Ende zur Auswahl,
 Themen #01 bis #04 und #10 bis #11 werden zuerst vergeben, Thema 13
 vor Thema 14 !

Was sind "gute" Programmierer und wie "gut" sind sie?



- D.L. Parnas: *Education for computing professionals* ; IEEE Computer, 23 (1991) 1, 17-22 (insbesondere über CS, 20-21)
- D.L. Parnas: *The professional responsibilities of software engineers* . In: K. Brunnstein, E. Raubold (Eds.) *Information processing* ; North-Holland, 1994, 332-339
- D. Boundy: *A taxonomy of programmers* ; ACM SIGSOFT, 16 (1991) 4, 23-30
- S. Summit: *Skills needed in programming* ; 1995/96, www.eskimo.com/~scs/cclass/progintro/sx1.html
- H. Stanislaw et al.: *A note on the quantification of computer programming skills* ; Int. J. Human-Computer Studies 41 (1994), 351-362
- NN: *Programmierer* ; siehe de.wikipedia.org/wiki/Programmierer

Was sind "gute" Programmierer und wie "gut" sind sie?



- NN: (*Levels of programming skilles*) ; 2005, Auszug aus www.procata.com/blog/archives/2005/05/10/expert-programmers/
- H. Wedekind: *Gibt es eine Ethik der Informatik? - Zur Verantwortung der Informatik* ; Informatik-Spektrum, 10 (1987) 6, 324 – 328 (auch 12 (1989) 1, 51-52)
- C. Duncan: *The career programmer - Guerilla tactics for an imperfect world* ; aPress, ISBN 1-59059-624-2, 2. Auflage, 2006

Wie "gut" ist Software?



Was heißt „gut“ in diesem Zusammenhang?

- M.Hendrix: *Software-Qualität*; undatiert,
www.wi-bw.tfh-wildau.de/~hendrix/grundstudium/cpp/skript/swqualitaet.html
- NN (Wikipedia): *Softwarekrise*; 2007,
de.wikipedia.org/wiki/Softwarekrise
- NN (Wikipedia): *Software quality*; 2007 (kritisch anzusehen!),
en.wikipedia.org/wiki/Software_quality
- NN (Wikipedia): *ISO 9000*; 2007,
en.wikipedia.org/wiki/ISO_9000
- NN (Wikipedia): *Freie Software*; 2007,
de.wikipedia.org/wiki/Freie_Software
- NN (Wikipedia): *Freeware*; 2007,
de.wikipedia.org/wiki/Freeware

Fehlerarten, wie kommt es zu Fehlern

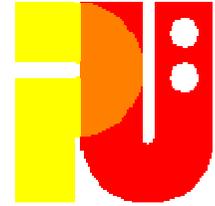


- NN (Wikimedia): *Programming style* ; 2007, en.wikipedia.org/wiki/Programming_style
- J. Ebert et al.: *Software-Testwirtschaftlichkeit* ; 2006, www.uni-koblenz.de/FB4/Institutes/IST/AGEbert/Teaching/SS06/SoftwareSystemTestingMethods/02-TECO.ppt
- S. Kohlmann: *Die 7 Todsünden der Programmentwicklung* ; 2005, www.speexx.de/java/7.todsunden.der.softwareentwicklung.pdf
- U. Kastens et al.: *Entwurfsfehler, Übersicht* ; ag-kastens.uni-paderborn.de/lehre/material/oop/folien/comment501-515.2.pdf
(dazu auch: *OO-Modellierung: Entwurfsfehler finden und beseitigen* ; aus: ag-kastens.uni-paderborn.de/lehre/material/swei/aufgaben/blatt10/Blatt10.html)



- A. Zeller: *Why programs fail - A guide to systematic debugging*; dpunkt Verlag, 2005 , ISBN: 3-89864-279-8, auch: www.whyprogramsfail.com/book.php
- R.L. Baber : *Fehlerfreie Programmierung für den Software-Zauberlehrling* ; 1990, Oldenbourg, ISBN 3-486-21637-6
- R.L. Baber: *Praktische Anwendbarkeit mathematisch rigoröser Methoden zum Sicherstellen der Programmkorrektheit* ; 1995, de Gruyter, ISBN 3-11-014764-5 (Dissertation TU Darmstadt)
- W. Paul: *BMBF-Forschungsprojekt Verisoft der Saarbrücker Informatik auf der CeBIT* ; 2006, idw-online.de/pages/de/news147803
- W. Fleisch: *Validierung komponentenbasierter Software für verteilte Echtzeitsysteme* ; elib.uni-stuttgart.de/opus/volltexte/1999/454/pdf/454_1.pdf

Reparaturmaßnahmen bei "syntaktischen" Fehlern



- G.D. Ripley, F.C. Druseikis: *A statistical analysis of syntax errors* ; Computer Languages 3 (1978) 4, 227-240
- M. Burke, G.A. Fisher: *A practical method for syntactic error diagnosis and recovery* ; ACM SIGPlan Notices 17 (1982) 6, 67-78
- M. Burke, G.A. Fisher: *A practical method for LR and LL syntactic error diagnosis and recovery* ; ACM Transact. Progr. Languages and Systems 9 (1987) 2, 164-197
- P. Degano, C. Priami: *Comparison of syntactic error handling in LR parsers* ; Software: Practice and Experience 25 (1995) 6, 657-679
- C. Cerecke: *Repairing syntax errors in LR-based parsers* ; 25th Australasian conference on Computer Science, 2002, 17-22
- J. Grosch: *Efficient and comfortable error recovery in recursive descent parsers* ; Structured Programming 11 (1990) 3, 129-140

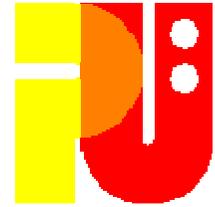
Reparaturmaßnahmen bei "semantischen" Fehlern



- C.W. Johnson, C. Runciman: *Semantic errors – diagnosis and repair*; ACM SIGPlan Notices 17 (1982) 6, 88-97
- A.R. Ford, T. J. Teorey: *Common syntax and semantic errors*; 2002, Chapter 2, in ftp://ftp.prenhall.com/pub/esm/sample_chapters/engineering_computer_science/ford_teorey/practical_debugging_cpp/sample_chapters.html
- F. Tip, T.P. Dinesh: *A slicing-based approach for locating type errors*; ACM Transact. Software Engineering and Methodology, 2001, 10 (2001) 1, 5-55

Fortsetzung

Reparaturmaßnahmen bei "semantischen" Fehlern



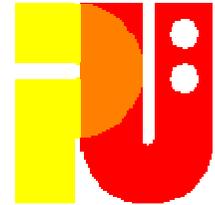
- L. Fegaras: *Semantic actions* ; 2002, lambda.uta.edu/cse5317/notes/node27.html
- S.E. Edwards: *Types and static semantic analysis* ; 2002, www1.cs.columbia.edu/~sedwards/classes/2006/w4115-fall/types.pdf
- A. Jost: *Semantic analysis and type checking bzw. More semantic analysis* ; 2007, users.cs.dal.ca/~jost/4131/lecture10.pdf bzw. [...lecture12.pdf](http://users.cs.dal.ca/~jost/4131/lecture12.pdf)
- A. Diwan: *Semantic analysis (und anderes)*; 2002, www-plan.cs.colorado.edu/diwan/3155-01/lectures/4-semantic.pdf, [8-types.pdf](http://www-plan.cs.colorado.edu/diwan/3155-01/lectures/8-types.pdf), [9-type-checking.pdf](http://www-plan.cs.colorado.edu/diwan/3155-01/lectures/9-type-checking.pdf)
- T. Tantau: *Elementare Datentypen – Äpfel und Birnen kann man nicht vergleichen -* ; 2005, www.tcs.uni-luebeck.de/Lehre/InfoA/WS0506/2005-11-15-beamer_version.pdf



- M. Ben-Ari: *Compile and runtime errors in Java* ; 2007, stwww.weizmann.ac.il/g-cs/benari/java/errors.pdf
- N. Ziring: *Welcome to the Java mistakes page!* ; 2001, users.erols.com/ziring/java-npm.html
- NN: (Java compile time errors – von einem Schüler -) ; 2004, www.java-forum.org/de/profile.php?mode=viewprofile&u=348&sid=f24a51d1821007bba0779c5d29faaec

Ergänzen um eigene Erfahrungen mit Compilern !

Beispiele: Editoren



- P.J. Denning: *Smart editor* ; Comm ACM 24 (1981) 8, 491-493
- T. Teitelbaum, T. Reps: *The Cornell Program Synthesizer (CPS): a syntax-directed programming environment* ; Comm ACM 24 (1981) 9, 563-573
- J. Heering: *Existing language development systems* ; 1999, homepages.cwi.nl/~jan/semantics/node3.html
- K. Stramm (Shareware-Newsletter von ZDNet Deutschland): *Komforable Editoren für Programmierer mit vielen Extras*; www.zdnet.de/downloads/weekly/22/weekly_446-wc.html bzw. www.zdnet.de/builder/program/0.39023551.39143478.00.htm
- NN (Wikipedia): *Integrierte Entwicklungsumgebung* ; 2007, de.wikipedia.org/wiki/Programm-Entwicklungsumgebung

Eigene Erfahrungen ?

Eclipse ; siehe Thema 13

Smalltalk ; siehe Thema 14

Fehlermeldungen



- NN (Wikipedia): *Error message* und *Design of error messages* ; 2007, en.wikipedia.org/wiki/Error_message bzw. en.wikipedia.org/wiki/Design_of_error_messages
- G. Braun: *Error messages for Windows* ; 2007, www.gregorybraun.com/MSWinErr.html
- Blaise W. Liffick: *Dealing with (C++) compiler errors* ; undated, cs.millersville.edu/~liffick/computingtips/compermsgs.htm
- R. Green : *Error messages - Java glossary* ; 2007, mindprod.com/jgloss/errormessages.html
- J. Nielsen: *Error message guidelines* ; 2001, www.useit.com/alertbox/20010624.html
- NN: *Web server error messages* ; www.webopedia.com/quick_ref/error.asp
- NN (Museum of Faulty Error Messages), siehe: www.moffem.de/de/index.html



- L. Kretzberg: *Wichtige Schritte beim Programmieren (nicht nur mit Matlab)*; 2005, www.uni-oldenburg.de/sinnesphysiologie/14272.html
- A. Drappa: *Checkliste zur Robustheit*; 1996, www.iste.uni-stuttgart.de/se/links/checklists/download/Robustheit-1.html
- NN (ETH Zürich): *Microsoft Office auf Macintosh*; 2002, isg.ee.ethz.ch/events/j2002/papers/OfficeOnMac.pdf
- M.J. Darnell: *Bad human factors design*; unregelmäßig, www.baddesigns.com/
- NN: *Was macht ein Technischer Redakteur?*; undatiert, www.us-english.de/downloads/Why%20hire%20a%20professional%20writer.doc
- NN: *Notebook-Hersteller haftet für Schaden trotz Fehlbedienung*; 2005, www.finanztip.de/recht/sonstiges/notebook-mangel-fehlbedienung.htm
(gilt dies auch bei Software ?)

Abfangen von Laufzeitfehlern im Programm



- NN (Wikipedia): *Runtime* ; 2007, siehe en.wikipedia.org/wiki/Runtime

Betrachte Vorgehen bei C, C++, Java, C#, Eiffel.
Smalltalk/VisualWorks als Beispiel siehe #14

- NN (Wikipedia): *Design by contract* ; 2007, de.wikipedia.org/wiki/Design_by_Contract
- B. Meyer et al.: *Building bug-free O-O software: An introduction to Design by Contract™*; 2007, archive.eiffel.com/doc/manuals/technology/contract/
- B. Meyer: *Design by contract - A conversation with Bertrand Meyer* ; 2003, www.artima.com/intv/contracts.html
- G. Taentzer: Design-by-contract (irgendwo heißt es da „Boolsche“, Seite 212 – was soll diese Namensschluderei ?)
marburg.de/~swt/sqSS2007/file
- P. Jackisch: *Ausnahmebehandlung (exception handling)* ; 2006, www.tu-ilmenau.de/fakia/fileadmin/template/startIA/swt/Lehre/2006ss/OOP_SS06/Vorlesungen/VL_KOOP_SS06_7.pdf

Fehleranzeigen



Einige „moderne“ Beispiele

- NN: *Runtime errors - JScript -*; 2006,
www.eulanda.de/ZH/inside/entwicklerhandbuch/jscript/referenz/grundlagen/fehlermeldungen.htm
- NN: *(runtime error listings related to MS Windows)*; 2007,
www.computerhope.com/issues/ch000380.htm
- C.M. Kozierek: *(runtime system exceptions related to MS Windows)*; see frame on the left, 2001,
www.pcguide.com/ts/x/sys/operr.htm

Finde und diskutiere weitere gute Beispiele
in Programmiersprachen und Programmiersystemen

„Debugger“



- NN (Wikipedia): *Debugger*; 2007,
de.wikipedia.org/wiki/Debugger
- NN(Wikipedia): *PL/C*; 1970x,
en.wikipedia.org/wiki/PL/C
- R.W. Conway, T.R. Wilcox: *Design and implementation of a diagnostic compiler for PL/I*; Comm ACM, 16 (1973) 3, 169 - 179

MS Visual Studio und Eclipse hier als Beispiele

Smalltalk als Beispiel, nächster Termin

Fortsetzung

„Debugger“



- NN(Wikipedia): *Microsoft Visual Studio .NET*; 2007, de.wikipedia.org/wiki/Microsoft_Visual_Studio_.NET
- NN(Microsoft): *Visual Studio 2005 Professional Edition*; 2007, www.microsoft.com/germany/msdn/vstudio/products/professional/default.aspx
(am Beispiel von C#)
- NN (Wikipedia): *Eclipse (IDE)*; 2007, [de.wikipedia.org/wiki/Eclipse_\(IDE\)](http://de.wikipedia.org/wiki/Eclipse_(IDE))
- NN (Eclipse Foundation): *Eclipse - an open development platform*; 2007, www.eclipse.org/
- K. Köhler: *Warum mit Eclipse entwickeln*; 2007, www.oio.de/public/warum-eclipse.htm

Smalltalk als Beispiel, nächster Termin

Beispiel: Smalltalk (1982 ff !)



Smalltalk-80 – VisualWorks, voll dynamisch,
sehr frühe Integration von Programmentwicklung,
Programmausführung und Testen

- A. Goldberg: *Smalltalk-80 - The interactive programming environment* ; Addison-Wesley, 1984, ISBN 0-201-11372-4, auch stephane.ducasse.free.fr/FreeBooks/TheInteractiveProgrammingEnvironment/TheInteractiveProgrammingEnvironment.pdf
- NN: *Welcome to the Cincom VisualWorks Tutorial* ; 2007, www.cincomsmalltalk.com/tutorials/version7/tutorial1/
- NN: *Survey on debugging* ; 1997, www-static.cc.gatech.edu/classes/cs2390_97_summer/survey/debugging.html
- NN: *A better name for the debugger* ; undated, wiki.cs.uiuc.edu/VisualWorks/a+better+name+for+the+debugger
- I. Tomek: *Joy of Smalltalk* ; wiki.cs.uiuc.edu/VisualWorks/Joy+of+Smalltalk bzw. stephane.ducasse.free.fr/FreeBooks/Joy/

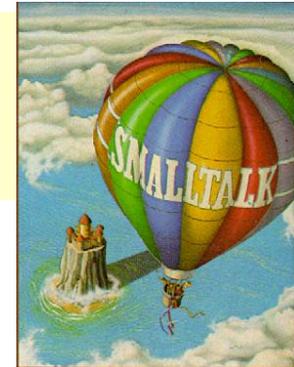
Fortsetzung

HJH#14b

2007/2008

Beispiel: Smalltalk (1982 ff !)

Smalltalk-80 –
und viele Implementierungen/Dialekte,
z. B. SmalltalkX und GNU Smalltalk



- C. Gittinger: *Error & warning messages (in SmalltalkX applications)*; 1995/2004, www.exept.de:8080/doc/online/english/programming/errmsg.html
- NN: *When things go bad (in GNU Smalltalk applications)*; 2002, www.gnu.org.ua/software/smalltalk/gst-manual/gst_36.html#SEC81

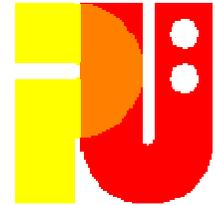
Und was meint Microsoft dazu? Und über Smalltalk-mobil!

- G. Heeg: *Georg-Heeg-Roadshow*, 2003, www-koethen.heeg.de/downloads/GeorgHeeg-Roadshow.ppt (mehrere Fassungen)

GOOGLE-Suche mit **debugger (smalltalk, visualworks)**, u.a.:

- C. Burkert: *Smalltalk versus Java*; 2003, www.chrisburkert.de/download/smalltalk/svj/chbuSmalltalkSVJ_2003_02_01.pdf

Computeranwendung ohne Fehler (?)

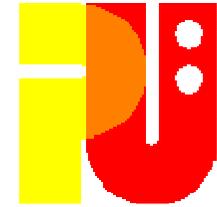


E.W.D. Dijkstra: ***„Testing can only show the presence of defects, but never the absence“***

- NN (H. Schnittker Versicherungsmakler GmbH): *IT Haftpflicht* ;
"geupdatet am" 22.9.2007:
[www.versicherung-portal-1.de/
betriebliche-versicherungen/it-haftung.php](http://www.versicherung-portal-1.de/betriebliche-versicherungen/it-haftung.php)
- P. Schefe: *Strategische Konstruktion von Software* ; 1998,
[asi-www.informatik.uni-hamburg.de/personen/
schefe/priv/PhilSWT/PHILSWTneu5-1.doc](http://asi-www.informatik.uni-hamburg.de/personen/schefe/priv/PhilSWT/PHILSWTneu5-1.doc)
- F. Seiffert: *Wenn etwas schiefgehen kann, dann wird es schiefgehen* ; 2007:
www.florian-seiffert.de/node35.html

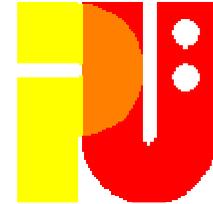
Aufgabe: GOOGLE-Suche mit **programm* fehlerfrei** und beobachten der abschwächenden Adjektive zu **„fehlerfrei“** !

Nachwort



in Bearbeitung

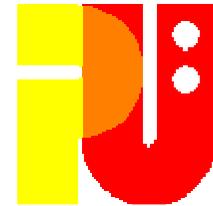
“Allgemeine Geschäftsbedingungen”



- Referat maximal 40 Minuten
- Datenprojektor („*beamer*“) ist vorhanden
- Anschließend Diskussion
 - **inhaltlich**
 - **Vortragstil**
- Auf Anforderung, mindestens fünf Tage vorher, kann PC für Präsentation mit PowerPoint bereitgestellt werden
- Ausarbeitung maximal acht bis zehn Seiten DIN A4;
(in der Regel) nach einer Woche abzugeben (**Papier**);
Literaturangaben nicht vergessen!
- Nach Bestätigung (eventl. Korrekturwünsche) durch Betreuer
Abgabe der endgültigen Ausarbeitung innerhalb einer
Woche als **doc-/pdf-Datei**
(maximal 2 Mbyte groß, mit Namen “WS07-08-#xx-version-ii.<ext>”,
“xx” elem {01 ÷ 15}, “ii” elem {1 ...})

Ein Hinweis des Dekans:

“Der Fachbereich Informatik mißt der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch das strikte Verfolgen von Plagiarismus. Mit der Abgabe einer Lösung (Hausaufgabe, Programmierprojekt, Seminararbeit Diplomarbeit, etc.) bestätigen Sie, daß (Sie/Ihre Gruppe) der alleinige Autor/die alleinigen Autoren des gesamten Materials sind. Falls Ihnen die Verwendung von Fremdmaterial gestattet war, so müssen Sie dessen Quellen deutlich zitiert haben. Bei Unklarheiten zu diesem Thema finden Sie weiterführende Informationen unter www.informatik.tu-darmstadt.de/Plagiarism oder sprechen Sie Ihren Betreuer an.”



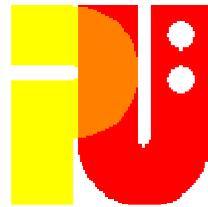
Web-Seite des Seminars:

<http://sun1.rbg.informatik.tu-darmstadt.de/PU/Seminar-Fehler>

(dort findet sich u.a. die Vorlage für das
Deckblatt Ihrer Ausarbeitung
und der **Terminplan** !)

**Nichts als Anstand, Sie beim
Lesen nicht stolpern zu lassen.**

und mich !



HJH#01
2007/2008

<Thema>

