

This is the announcement of a BETA tutorial in
Darmstadt/Germany. This tutorial will be held in German.

Technische Hochschule Darmstadt
Fachgebiet Programmiersprachen und Uebersetzer

Tutorium ueber

B E T A

=====

Wir veranstalten am Freitag, den 8. Dezember 1995 ein ganztageiges Tutorium ueber BETA, eine objektorientierte Programmiersprache und eine dazugehoerige Programmumgebung in der SIMULA-Tradition. Nachstehend finden Sie eine ausfuehrliche Inhaltsangabe fuer das Tutorium.

Diese Veranstaltung steht Interessierten aus der Hochschule und der Wirtschaft offen; sie ist kostenfrei.

Referent ist Dipl.-Ing. Wilfried Rupflin, Fachbereich Informatik der Universitaet Dortmund. In Dortmund wird BETA den Informatikstudenten im Grundstudium als moderne, objektorientierte Programmiersprache vorgestellt. Herr Rupflin hat grosse Erfahrungen mit BETA u.a. bei der Vorbereitung und Durchfuehrung dieser Lehrveranstaltungen in Dortmund gewonnen.

Ich lade Sie freundlichst zur Teilnahme ein.

Fuer eine Anmeldung in meinem Sekretariat waere ich dankbar
(Frau Braun, THD, FB 20, FG PUE, Alexanderstrasse 10, 64283 Darmstadt,
Tel.: 06151/16-3410, Fax: 06151/16-6648,
E-Post: braun@pu.informatik.th-darmstadt.de).

Univ.-Prof. Dr.-Ing. Hans-Juergen Hoffmann
Leiter des Fachgebiets Programmiersprachen & Uebersetzer
im Fachbereich Informatik
der Technischen Hochschule Darmstadt

Ort: Alexanderstr. 10, Darmstadt, Saal 23/133
=====

Zeitplan:

=====

- 8.10 Begruessung (Prof. Dr. H.-J. Hoffmann)
- 8.15 Vom Objekt zum Konzept: BETA - die naechste Generation, 1. Teil
(Dipl.-Ing. W. Rupflin)
- 9.45 Pause
- 10.00 Vom Objekt zum Konzept: BETA - die naechste Generation, 2. Teil
(Dipl.-Ing. W. Rupflin)
- 11.30 Mittagspause

Inhaltsangabe

=====

Vom Objekt zum Konzept:

BETA -- die naechste Generation?

ALGOL brachte als wichtigste Innovation Struktur in die Programmierung. So ermoeoglicht das Blockkonzept, durch exaktes Festlegen der Gueltigkeitsbereiche von Definitionen Programme derart zu strukturieren, dass Abhaengigkeiten zwischen verschiedenen Programmteilen nur noch von der Logik des Programms selbst bestimmt sind.

SIMULA fuehrte als weitere Strukturierungselemente Klassen und Koroutinen ein. Klassen erlauben es, Objekte entsprechend ihrer Eigenschaften zu klassifizieren, so wie dies auch in den meisten Wissenschaftszweigen ueblich ist. Die Strukturierung der Programme hinsichtlich der Gemeinsamkeiten und Unterschiede von Objekten bringt zusaetzlich zur besseren Uebersicht auch eine erheblich gesteigerte Programmiereffizienz durch die damit verbundenen Konzepte der Vererbung und der dynamischen Auswahl der dem Objekt gemaessen Operationsimplementierungen. Koroutinen ermoeoglichen darueber hinaus die Strukturierung der Aktionen eines Programms in voneinander weitgehend unabhengige Sequenzen.

BETA schliesslich unterstuetzt neben der Klassifizierung von Objekten auch die Klassifizierung von Aktionen. Dies geschieht nicht durch zusaetzliche Konstrukte, sondern durch die Generalisierung der Konstrukte Klasse und Prozedur zum Konstrukt Muster (Pattern). Muster dienen der Beschreibung von Konzepten als der Abstraktion von Phaenomenen, also von Objekten und Aktionen.

Dieser sehr theoretisch scheinende Ansatz bietet im Zusammenwirken mit der Blockstruktur handfeste Vorteile, indem er unter anderem Vererbung fuer Prozeduren, Typ-Parametrisierung, Ausnahmen (Exceptions), Zusicherungen, spezialisierte Kontrollstrukturen, Prozesse, Monitore und Rendezvous-Mechanismen ohne die Verwendung weiterer Konstrukte ermoeoglicht. Neben der Klassifizierung unterstuetzt Beta in hohem Masse auch die wichtigen Strukturierungskonzepte Teile-Komposition, Referenz-Komposition und Lokale Definition.

Wegen der guten Abstraktionsunterstuetzung und der Universalitaet der Konzepte eignet sich Beta auch und gerade fuer die Lehre. Dies und die Unterstuetzung von Nebenlaeufigkeit waren entscheidende Argumente fuer die Wahl von Beta als erste Programmiersprache am Fachbereich Informatik der Universitaet Dortmund.

Im Tutorium werden die oben skizzierten Konzepte motiviert sowie ausfuehrlich mit ihren Moeglichkeiten und Konsequenzen dargestellt und diskutiert. Weitere Themen neben den oben erwaehten werden u.a. sein: Mehrfaches Erben, Kovarianz, Metaklassen, Callback-Prozeduren (auch zu C), Trennung von Schnittstelle und Spezifikation, getrennte Uebersetzung.

Als Beispiel fuer die Maechtigkeit und Eleganz der Sprache wird die Programmierschnittstelle fuer OSF/Motif vorgestellt.

Im Anschluss an den Vorlesungsteil besteht am Nachmittag Gelegenheit zur Diskussion speziellerer Fragen und zu ersten Versuchen und Experimenten mit Beta.

(Das Beta-System und zahlreiche vorbereitete Beispiele -- auch mit OSF/Motif -- werden auf einem Unix-Rechner verfuegbar sein.)

Literatur:

- O. Lehrmann Madsen, B. Moller-Pedersen, K. Nygaard
Object Oriented Programming in the BETA Programming Language
ISBN 0-201-62430-3, Addison Wesley, June 1993
- M. Loefgren, J. Lindskov Knudsen, B. Magnusson, O. Lehrmann Madsen
Object-Oriented Environments - The Mjolner Approach
ISBN 0-13-009291-6, Prentice Hall, 1994