Lehre objektorientierter Konzepte
Erfahrungsberichte aus der Ausbildung

Ein Fachgespräch anläßlich der NET.OBJECTDAYS 2001

Objektorientierung hat sich bei der Software-Entwicklung in der Systemanalyse, der Programmierung und dem Systemeinsatz durchgesetzt. Die Vielfalt der Ansätze, der Sprachen und der Systeme hat jedoch noch nicht zu einem breit getragenen Verständnis über das Vorgehen in der Lehre geführt.

Das Fachgespräch wird ein Forum bieten, wie die Lehre objektorientierter Konzepte gestaltet wird oder werden kann,

in Universitäten, in Fachhochschulen
und bei beruflicher Aus- und Weiterbildung.

Zusätzlich zu drei eingeladenen Vorträgen international bekannter Wissenschaftler werden zwei interessante konzeptionelle Ansätze und vier Erfahrungsberichte als eingereichte Vorträge vorgestellt.

Programmkomitee:

Prof. Dr. Ernst-Erich Doberkat, Universität. Dortmund
Georg Heeg, www.heeg.de Dortmund-Köthen-Zürich (Vorsitz)
Prof. Dr. Thomas Gross, Eidg. Techn. Hochschule Zürich
Prof. em. Dr. Hans-Jürgen Hoffmann, Techn. Universität Darmstadt (Vorsitz)
Prof. Dr. Hanspeter Mössenböck, Johannes Kepler Universität Linz
Prof. Dr. Guido Pfeiffer, Fachhochschule Hamburg
Prof. Dr. Erhard Plödereder, Universität Stuttgart
Prof. Dr. Jürgen Winkler, Friedrich-Schiller-Universität Jena
Prof. Dr. Alfred Zimmermann, Fachhochschule Reutlingen

Computational Diversity and a Passion For Applications Objects are Not Everything

Dave Thomas
Bedarra Corporation
and
Carleton University & University of Queensland

Eingeladener Vortrag

In this talk I reflect on my experience as a teacher and employer of 3 generations of business, engineering and computer science graduates. Although the technologies have changed from punched cards to web based distance education the essence of good teaching has not. Although the language of choice has moved from FORTRAN to Java, the essence of problem solving has not.

We argue that to be effective with objects one needs to be effective at modeling. To be effective at modeling one has to live in a world rich in applications. In order to develop creative and effective solutions students need to work together, and have access to a wide variety of computational machinery.

We discuss some innovative approaches to education including CS as studio, combined honors degrees in CS and other disciplines and the role of cooperative education.

Bio

Dave Thomas is founder, past CEO and President of Object Technology International (OTI), developers of ENVY/Developer, a unique CFM environment for object oriented development; virtual machines and IDEs for IBM VisualAge for Smalltalk; IBM VisualAge for Java; and VisualAge Micro Edition for Embedded Systems. In 1998 after the successful transition of OTI to IBM, Dave formed Bedarra Corporation. Bedarra provides virtual CTO, competitive analysis, contract R&D, angel investment and business mentoring. Bedarra's current focus areas are distance education; next generation applications; and embedded applications.

Dave is an Adjunct Research Professor at Carleton University and the University of Queensland.

Objektorientierte Konzepte in einem an Praxis orientierten Kurs

Jean-Francois Perrot
Université Pierre et Marie Curie, Paris

Eingeladener Vortrag

Es handelt sich um einen einjährigen Kurs im sog. "5ten Jahr nach dem Abitur" (frz. "Bac+5"), der nach einem regelmässigen Universitätslehrgang mit Hauptfach Informatik kommt [*].Das Ziel dieses Kurses ist, die vorher rein technische Ausbildung der Studenten mit berufsbezogenem Anwendungswissen zu vervollständigen, ganz im Sinne der Empfehlungen der GI zur Verstärkung der Anwendungsorientierung in Studiengängen der Informatik an Universitäten [**]. Zugleich werden gewisse technische Themen weiter entwickelt, die als besonders wichtig für die Berufstätigkeit der Absolventen erkannt sind:

- Programmiersprachen und deren Interpretation,
- Modellierung und Programmierung (vom Standpunkt der künstlichen Intelligenz),
- Engineering.

Objektorientierung erscheint als Grundkomponente in allen drei Themen.

Die Anpassung zum Berufsleben erfolgt einerseits durch eine Reihe von Vorträgen, die von Fachleuten aus der Branche gehalten werden, andrerseits durch die Ausführung eines ziemlich wichtigen Projekts unter "realistischen" Bedingungen, und schliesslich durch das Praktikum, das zwischen den Monaten Mai und September stattfindet.Der Kontakt mit der Branche wird ganz besonders gepflegt (vor allem von meinem Kollegen Gil Blain) und die Meinung der Arbeitgeber, bei denen unsere Absolventen tätig sind, wird sehr hochgeschätzt. Entsprechend wechselt der Inhalt des Kurses mehr oder weniger jedes Jahr.

Ich werde versuchen, die Evolution des Kurses über etwa 15 Jahren vom Standpunkt der Objektorientierung zusammenzufassen und die dahinterliegende Dialektik zwischen Neuigkeit und wissenschaftlich-technischem Inhalt zu analysieren.Von 1986 bis 1990 waren Lisp und Prolog die zwei Säulen des technischen Unterrichts: und damals wurde gelehrt, wie man Klassen und Objekte in Lisp implementiert. Danach kam die Herrschaft von Smalltalk, zuerst (1990-93) als rein didaktisches Mittel, Objekte als Kern des Modellierungsprozesses einzusetzen, nachher (1994-97) als Hauptsprache für industrielle Projekte. In den letzten Jahren (1998-2001) hat Smalltalk an industrieller Wichtigkeit zugunsten von Java, Corba und jetzt den Enterprise Java Beans verloren. Heute werden

Java, Smalltalk und Objective Caml gelehrt, miteinander verglichen und kritisiert. Natürlich dazu UML, aber auch Multi-Agent Systems.

In allen Fällen haben wir uns bemüht, das Gleichgewicht zwischen genauen Kenntnissen über Implementationstechnik und einem guten Verständnis des Modellierungsprozesses zu bewahren.

[*] DESS Génie des Logiciels Applicatifs : http://www.ufr-info-p6.jussieu.fr/dess-gla

[**] Informatik Spektrum 22, 6 (Dezember 1999) S. 444-448

Learning is a Community Experience

Adele Goldberg
Neometron, Inc., USA

Eingeladener Vortrag

Perhaps it is obvious -- you do not learn alone, but you do take responsibility for your own education. And part of that responsibility is fulfilled by your creating or finding affinity groups -- the collection of places or people that provide a motivating context in which you can learn. Your choice of affinity group reflects the social and experiential nature of the learning process. What interaction style suits you best -- push or pull? what timeframe provides the best retention -- preparing forward or just-in-time, on-the-job?

Much learning takes place on-the-job, but is retained by taking that experience and storing it as part of a more general knowledge base -- the set of enduring principles that allows you to succeed in new situations, using new technologies or applying old ones. The enduring principles of object technology are often lost in the battle over programming language, engineering methodology, or architectural preference. The need to be an expert in transient technology creates an atmosphere of immediate training rather than long-term education, and often creates practitioners crippled by the currency of their expertise.

The early introduction of object technology -- driven from research directly into commercial delivery -- created an atmosphere of early fear on the part of practitioners. The source of the fear remains open for debate -- whether the technology is inherently hard to understand, or whether the open source and architecturally sensitive nature of programming with objects creates excessive pressure, or whether the need to understand the application domain requires a different kind of skill? Regardless, we have learned some lessons about educating students and experienced professionals -- the need to foster a community of learners who help one another develop themselves as practitioners while they develop the practice itself.