

Entwicklung multimedialer Lehr-/Lernprogramme für objektorientierte Programmiersprachen

Projektleiter

Prof. Dr.
Klaus Zeppenfeld

Zeitraum
2003

Stud. Mitarbeiterinnen und Mitarbeiter

Verena Reimann
Hendrik Busch
David Kuß

Förderung

Fachhochschule
Dortmund,
Forschungssemester

Kontakt

Prof. Dr.
Klaus Zeppenfeld
Fachhochschule
Dortmund,
Fachbereich Informatik,
Emil-Figge-Straße 42,
44227 Dortmund,
Telefon:
(0231) 755 - 6765,
E-Mail: zeppenfeld@
fh-dortmund.de

Kurzfassung

In diesem Forschungsprojekt wurden die beiden multimedialen Lehr-/Lernprogramme XCompare (sprich: Cross Compare) und XLearn (sprich: Cross Learn) entwickelt. Sie dienen zum Vergleich und zum Erlernen der derzeit gängigen objektorientierten Programmiersprachen Java, C#, C++ und Ruby.

XCompare ist ein e-Learning-Werkzeug für computergestütztes Lernen (*computer based training, CBT*), das besonders für Umsteiger konzipiert ist, die bereits eine oder mehrere objektorientierte Programmiersprachen beherrschen und eine weitere erlernen wollen. XCompare vergleicht aspektweise die grundlegenden Konzepte und deren Umsetzung zwischen jeweils zwei beliebigen Programmiersprachen.

XLearn ist ein e-Learning-Werkzeug, in dem anhand von detaillierten Lehrtexten und Aufgaben auf jede objektorientierte Programmiersprache eingegangen wird. XLearn richtet sich an Neueinsteiger, die eine Programmiersprache von Grund auf erlernen wollen, aber bietet auch erfahrenen ProgrammiererInnen die Möglichkeit ihr Wissen zu testen.

Beide Werkzeuge sind lauffähig, einsetzbar, verfügen über eine gute Dokumentation und werden in verschiedenen Lehrveranstaltungen genutzt.

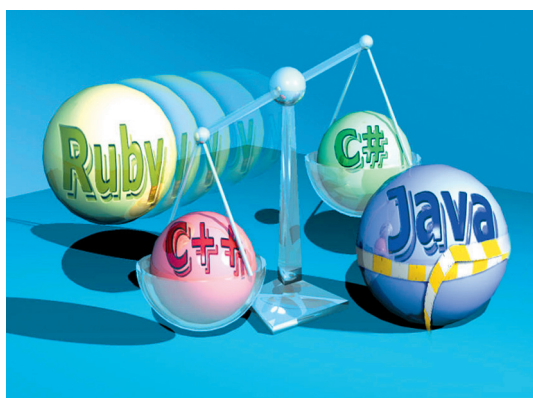
den Händen halten. Heute sind es Java, C++, C# und Ruby, die sich in der Softwareentwicklung etablieren bzw. etabliert haben und mit denen Anwendungen von der Analyse bis zum Programm mit durchgängigen Konzepten entwickelt werden können.

Dennoch hat die objektorientierte Softwareentwicklung erst ganz allmählich Einzug sowohl in die Lehre als auch in die Industrie gehalten. Dies liegt zum einen an der mittlerweile stark verbesserten Hardware, auf der die objektorientierten Anwendungen laufen bzw. erstellt werden und zum anderen daran, dass in den Köpfen der Entwickler und auch der Entscheider die Konzepte der Objektorientierung erst verstanden werden mussten, bevor sie nutzbringend eingesetzt werden konnten. Und genau um die Verbreitung und das Erlernen dieser Konzepte geht es in diesem Forschungsprojekt.

Im Fall der hier betrachteten objektorientierten Programmiersprachen Java, C++, C# und Ruby, lässt sich sogar behaupten: Wenn das Konzept der Objektorientierung verstanden ist, so ist es unwichtig, mit welcher Programmiersprache bzw. mit welchem Werkzeug eine Anwendung erstellt wird. Programmiersprachen und Werkzeuge sind und bleiben immer (nur) Hilfsmittel zum Zweck. Ihre bloße Nutzung garantiert noch keinen Erfolg im Sinne der anzuwendenden Technologien und Konzepte.

In der heutigen industriellen Softwareentwicklung kommt es häufig vor, dass ein Kunde beispielsweise ein Projekt in Java, ein anderer Kunde ein Projekt in C++ umgesetzt haben möchte. Viele Erfahrungen in Softwarehäusern zeigen jedoch, dass bei sehr vielen Mitarbeiterinnen und Mitarbeitern eine große Hemmschwelle besteht, sich in kurzer Zeit in eine neue Programmiersprache einzuarbeiten. Aussagen, wie z. B. „für diese Sprache bin ich nicht ausgebildet“, sind an der Tagesordnung, aus Sicht der Lehre jedoch unbegründet und falsch.

Wer Software entwickelt, sollte in der Lage sein, sich innerhalb von wenigen Tagen in eine andere Programmiersprache einzuarbeiten und ein neues Projekt zu beginnen. Die Erfahrung in der Anwendung einer Sprache kommt dann zwangsläufig beim Einsatz in diesem speziellen Projekt. Sind aber die Konzepte einmal verstanden, so erleichtert dieses Wissen den Umstieg sehr deutlich. Dieser Gedanke muss ebenfalls Einzug in die Ausbildung und die Lehre halten.



Forschungsgegenstand, Projekt, Thema

Objektorientierte Programmiersprachen können in der Informatik längst nicht mehr als Novum bezeichnet werden. Bereits im Jahre 1967 wurde mit der Programmiersprache Simula-67 das Klassenkonzept eingeführt. Zu Beginn der 1980er-Jahre konnten die Informatiker mit Smalltalk-80 die erste reinrassig objektorientierte Programmiersprache incl. Entwicklungsumgebung in

Aus diesem Grunde macht es sehr viel Sinn, die im Moment aktuellen vier objektorientierten Programmiersprachen Java, C++, C# und Ruby zu beschreiben und miteinander zu vergleichen. Unterstützt werden soll dieser Vergleich durch multimediale Lehr-/Lernumgebungen, die in der Lehre und Weiterbildung eingesetzt werden können.

Das Ziel dieses Forschungsprojekts ist zweigeteilt: Denjenigen, die wenig oder gar keine Erfahrung mit objektorientierten Vorgehensweisen und Programmiersprachen haben, wird durch die multimedialen Lehr-/Lernprogramme ein leicht zu verstehender Einstieg in die Konzepte und ein guter Vergleich der gängigsten Sprachen aus diesem Bereich gegeben. Den Kennern einer der hier behandelten Sprachen, die aufgrund ihres Interesses oder aufgrund beruflicher Gegebenheiten auf eine andere objektorientierte Programmiersprache umsteigen möchten bzw. müssen, wird dieser Schritt mit der Gegenüberstellung erleichtert. Neben Studierenden der Informatik, für die ein solcher Vergleich eine gute Hilfestellung ist, gehören demnach auch Entwicklerinnen und Entwickler aus der Industrie zu den Zielgruppen dieser Lehr-/Lernumgebungen.

Im Rahmen eines Forschungsfreisemesters wurden daher zwei multimediale Lehr-/Lernprogramme aus dem Bereich der objektorientierten Programmiersprachen entwickelt:

- XCompare (sprich: Cross Compare) ist ein e-Learning-Werkzeug für computergestütztes Lernen (Computer Based Training, CBT), welches besonders für Umsteiger konzipiert ist, die bereits eine oder mehrere der objektorientierten Programmiersprachen beherrschen und eine weitere erlernen wollen. Dabei kann, ausgehend von einer Programmiersprache, z. B. C++, die entsprechenden Inhalte bzw. die Syntax dieser Sprache mit jeweils einer der drei anderen Sprachen, in diesem Fall also Java, C# oder Ruby, aspektweise verglichen werden. Technologisch gesehen ist XCompare aber so flexibel gehalten, dass es auch für andere Inhalte geeignet ist.



- XLearn (sprich: Cross Learn) ist ein e-Learning-Werkzeug, in dem anhand von detaillierten Lehrtexten und zu bearbeitenden Aufgaben auf jede der



vier objektorientierten Programmiersprachen Java, C++, C# und Ruby eingegangen wird. XLearn richtet sich an Neueinsteiger, die eine Programmiersprache von Grund auf erlernen wollen, bietet aber auch erfahrenen Programmierern die Möglichkeit, ihr Wissen zu testen.

Beide Anwendungen sind in Java geschrieben, umfangreich dokumentiert und besitzen eine ausführliche Installationsanweisung. Sie sind technisch so konzipiert, dass sie problemlos erweiterbar und auch für andere Themen einsetzbar sind.

XLearn liegt, wie auch XCompare, als normale Java-Applikation, aber auch als WebStart-Version vor. Beide Programme sind direkt als Applikation von einer CD startbar, können aber auch auf einem Rechner installiert werden. Dies setzt eine korrekte Installation der JRE 1.4.1 oder höher voraus. Mit der Web Start Version von XLearn und XCompare wird die Voraussetzung erfüllt, dass beide Programme auch als Internetanwendung eingesetzt werden können.

Neue Inhalte können in beide Programme mit einer Zusatzanwendung, die sich ChapterContentCreator nennt, verwaltet und hinzugefügt werden. Mit diesem Programm lassen sich XML-Dateien erzeugen, welche alle in XCompare und XLearn angezeigten Daten beinhaltet. Es erlaubt dabei das Erstellen von Gruppen-Daten, Gemeinsamkeits-Daten sowie die Erstellung einer, allen Gruppen zugrunde liegenden, Kapitelstruktur.

Veröffentlichungen

Zeppenfeld, Klaus,
„Objektorientierte Programmiersprachen –
Einführung und Vergleich von Java, C++, C#,
Ruby“,
Spektrum Akademischer Verlag (Elsevier), 2004

Weitere Informationen zu diesem Projekt

Im Internet unter
<http://www.oo-programmiersprachen.de>

Abbildung 1 zeigt einen Bildschirmausschnitt von XLearn und in Abbildung 2 ist ein Bildschirmausschnitt von XCompare dargestellt.

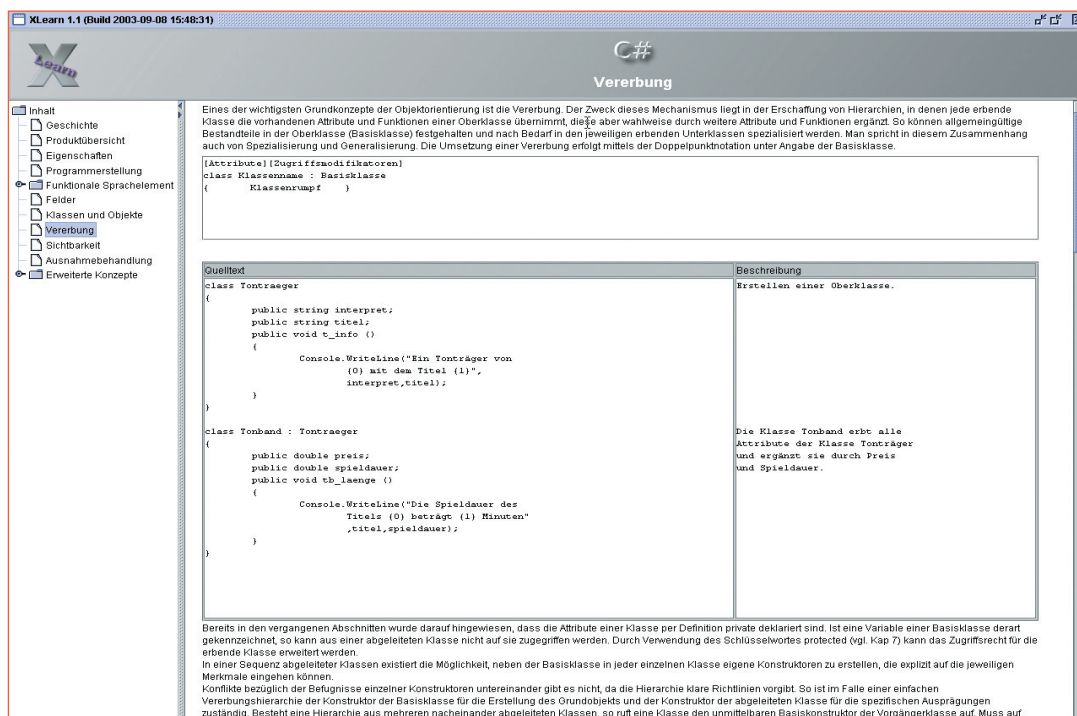


Abb. 1: XLearn

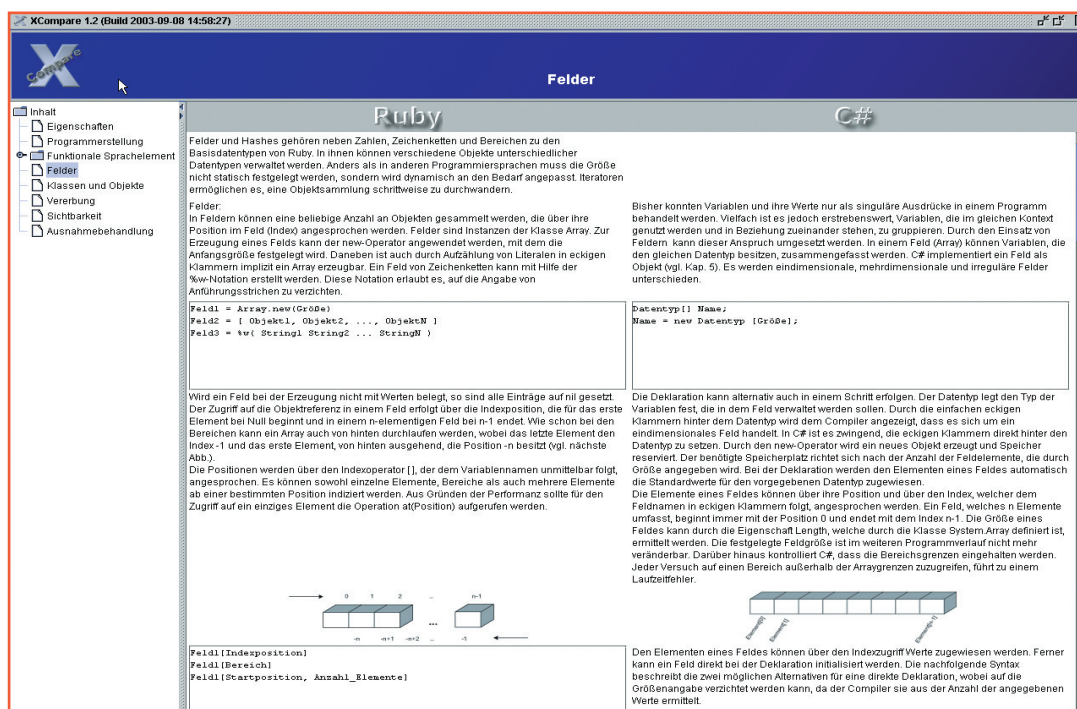


Abb. 2: XCompare