

Universität Potsdam
Institut für Informatik

Lehrveranstaltung „Proseminar Hypermedia-Lernsysteme“
Prof. A. Schwill

**Die Arbeitsweise hypermedialer Lernsysteme
am Beispiel der Systeme Lisp-Tutor und ELM-ART**

Schriftliche Ausarbeitung zum Seminarvortrag vom 29.01.2001

Autoren

Dorothea Böhnke (Matr.-Nr. 702565) und Claudia Eggerth (Matr.-Nr. 702857)

Potsdam, den 26.02.2001

Gliederung der Dokumentation

1. Vorstellung der Programmiersprache LISP

- 1.1 Geschichte der Entwicklung
- 1.2 Das LISP-Konzept
- 1.3 Grundlagen von LISP-Konstrukten

2. Vorstellung des Online-Lernsystems " LISP – TUTOR "

- 2.1 Einleitung
- 2.2 Erläuterungen der praktischen Arbeit
- 2.3 Fazit

3. Vorstellung des Online-Lernsystems " ELM – ART "

- 3.1 Einleitung
- 3.2 Erläuterungen der praktischen Arbeit
- 3.3 Fazit

4. Resümee

5. Literaturverzeichnis

1. Vorstellung der Programmiersprache LISP

1.1 Geschichte der Entwicklung

„Viele Innovationen in der Informatik entstanden [...] als Nebeneffekte. Das eigentlich Ziel war die KI, aber die Softwaretechnik wurde vorangetrieben, weil Studenten zwischen ein und drei Uhr morgens nebenbei etwas entwickelt haben, damit sie zwischen drei und vier Uhr morgens besser arbeiten konnten.“

Christiane Floyd, „Pioniere der Informatik“

Diese Aussage beschreibt in wenigen Worten den ursprünglichen Kerngedanken vieler Programmierarbeiten – Menschen möchten sich wiederholende und häufig aufwendige Arbeitsschritte erleichtern und leisten damit u.U. einen nicht geringen Beitrag zur Forschung auf den Gebieten der Computerwissenschaften.

Als im Jahre 1956 der Amerikaner John McCarthy am "Massachusetts Institute of Technology" (MIT) in Cambridge mit seinen Mitarbeitern erste Überlegungen zu der Programmiersprache *LISP* anstellte - die Bezeichnung *LISP* entstammt dem Arrangement der Begriffe „List“ und „Processing“ - , beabsichtigte man, neue Wege auf dem Gebiet der Listenverarbeitung zu gehen. Demzufolge hat diese Programmiersprache als eine der ersten große Bedeutung für die Entwicklung textverarbeitender Systeme erlangt, denn *LISP* unterstützte die Arbeit auf Zeichenketten und verarbeitete keine strukturierten Datensätze oder -felder.

Doch abgesehen von der Textverarbeitung wurde *LISP* ebenfalls ein großer Einfluss im Bereich der Künstlichen Intelligenz (KI) nachgesagt. Dieser Bereich der Informatik wurde von Forschern geprägt, die sich mit der Untersuchung der menschlichen Intelligenz befassten und versucht haben, Computer zu generieren, die zu Eigenschaften wie Bewusstsein oder selbständigem Denken fähig sind ¹.

LISP erreichte als relativ einfach anzuwendende Programmiersprache diesbezüglich einen so bedeutsamen Einfluss, weil sie als eine der ersten in diese Zeit innovativer Überlegungen „hineingeboren“ wurde und augenscheinlich kaum andere Möglichkeiten bestanden, unkompliziert zu programmieren.

In den 60er Jahren entstand eine wahre Euphorie, die bis Ende des 20. Jahrhunderts Computer erwartete, die sich menschlich verhalten würden. So entstanden auch im Laufe der Jahrzehnte, basierend auf diesen Gedanken, zahlreiche Science-Fiction-Filme, die den Menschen in naher Zukunft eine schöne neue Technik-Welt suggerierten.

1. Diese hochtrabenden Ziele wurden mit den Jahren realistischer formuliert; es wurden weniger „konkurrierende“ Maschinen erwartet, sondern Systeme entwickelt, die sich u.a. mit Problemen der Mustererkennung, Sprachverarbeitung oder Robotik beschäftigen.

1.2 Das LISP-Konzept

Neben der Programmiersprache *FORTTRAN*, die Mitte der 50er Jahre zur Formelübersetzung entwickelt wurde, ist *LISP* die zweitälteste heute noch genutzte Sprache. Im Laufe der Jahrzehnte entwickelten sich immer neuere Implementierungen, wobei einerseits Wert auf Benutzerfreundlichkeit, andererseits auf Kompaktheit gelegt wurde. In diesem Zusammenhang sind z.B. *MacLisp* oder *InterLisp* zu nennen.

Wird von *LISP* gesprochen, ist häufig eine ganze Sprachfamilie von sogenannten „*LISP* – Dialekten“ oder Abwandlungen gemeint, die auch gegenwärtig noch existieren. Eine der einflussreichsten Modifikationen ist die 1975 entwickelte Programmiersprache *SCHEME*, die auch gegenwärtig noch bei der Erforschung von Programmierkonzepten und in der Lehre eingesetzt wird.

Des Weiteren wurde *LISP* vielen Anwendungen als Makrosprache mitgegeben, d.h. als Programmiersprache für Makrobefehle, die mehrfach wiederkehrende Befehlsfolgen zu einem Oberbefehl zusammenfassen. Als Beispiel sei der UNIX-Editor „Emacs“ genannt, wo *LISP* dementsprechend Einsatz fand.

Anfang der 80er Jahre erreichten die *LISP*-Maschinen jedoch den Höhepunkt ihres Erfolges; weiterhin wurden jedoch vielfach neuartigere Implementierungen dieser Programmiersprache vorgestellt. Dabei kristallisierte sich sehr bald *CommonLisp* als Standard heraus, wobei es sich um eine sehr umfangreiche Sprache mit vielen Daten- und Ablaufstrukturen handelt.

1.3 Grundlagen von LISP-Konstrukten

Im Folgenden eine Auswahl von drei Möglichkeiten, in *LISP* zu programmieren. Da nur die Syntax von *LISP* anschaulich vorgestellt werden soll, wird nicht auf die programmiersprachliche „Stellung“ der einzelnen Konstrukte eingegangen.

Es wird deutlich, dass *LISP* den funktionalen Programmiersprachen angehört, da der funktionale Zusammenhang eines Problems beschrieben wird.

Die Syntax der Sprache wirkt nicht übermäßig kompliziert, jedoch ist eine Vielzahl von Klammerpaaren notwendig, um wirksame Befehlsfolgen zu erstellen.

Beispiel 1 ***Durchführung arithmetischer Berechnungen***

(+ 17 11 69) ≡ 97

(/ (+ 2 7) (- 11 10)) ≡ 9

Beispiel 2 ***Berechnungen mit Variablen***

(setf x 10) ⇒ Übergabe eines Wertes an die Variable x mittels
der Funktion *setf*

(+ 1 x) ≡ 11

Beispiel 3 ***Individuelle Funktionen definieren***

(defun mittel(x y) (/ (+ x y) 2)) ⇒ Definition der Funktion *mittel*
über die Funktion *defun*

2. Vorstellung des Online-Lernsystems " LISP – TUTOR "

2.1 Einleitung

Das Online-Lernsystem "LISP-Tutor" ist unter folgender Adresse zu finden:

<http://www.cs.tulane.edu/www/Villamil/lisp/lisp1.html>

Das Programm wurde an der Tulane-University in New Orleans entwickelt und daher auf Englisch gehalten.

Es ist kostenlos und jedem frei zugänglich, es wird z.B. keine Anmeldung über eine E-Mail-Adresse o.ä. gefordert.

2.2 Erläuterungen der praktischen Arbeit

Wird die Startseite des "LISP-Tutor" aufgerufen, erscheint eine Eingangsgrafik mit mehreren Verknüpfungen.

Test :
Last Class

- **Data Structure**
[S-Expression](#), [Atoms](#), [Pitfalls](#)
- **Basic Operations**
[Car](#), [Cons](#), [Atom](#), [Cdr](#)
[Null](#), [Quote](#)
- Basic I/O**
[Read](#) [Print](#)
- **Control Structures**
[Constants and Variables](#)
[Assignment](#)
[Function definition](#)
[Conditionals: Equal](#) [IF](#) [Cond](#)
- **Arithmetics**
[Operation: + - / * Add1 Sub1](#)
[Comparison: = /= ≤ ≥ ≤= ≥=](#)
[Predicates: Zerop Flusp Evenp Oddp](#)
[Integer Floatp](#)
[Others: Sqrt exp min max](#)
- **Sets**
[Member](#) [Union](#)
[Adjoin](#) [Intersection](#)
- **Recursion**

LISP TUTORIAL

• **Introduction**

Bibliography
About Lisp Tutorial

Point to the Topic with your mouse and click on it.

Auf diesen Unterseiten werden wichtige *LISP*-Konstrukte erklärt und nähergebracht, auch ist ein abschließender Test durchführbar.

Bewegt man sich auf dieser Indexseite weiter nach unten, erhält man eine kurze Erklärung über den Zweck des Online-Lernsystems, einen Hinweis auf die Homepage John McCarthy's und einen Inhaltsüberblick, der ebenfalls Verweise auf Unterseiten enthält.

Das Programm wird beschrieben als "[...] WWW tutorial that helps the students in the process of learning the LISP language."

Wird der Link *About Lisp-Tutorial* angewählt, erhält man auf weiteren Unterseiten (*Contents – How to use it / How to do it*) Hinweise über den Aufbau der einzelnen Kapitel, die aus Definitionen und kurzen Beispielsübungen bestehen.

Jedoch findet sich kein Anhaltspunkt, ob Grundvoraussetzungen erforderlich sind oder wie der Kurs optimal durchgeführt wird. Demzufolge wird der Nutzer beginnend alle Hyperlinks nacheinander anwählen, um sich eine logische Abfolge nach Schwierigkeitsgrad oder Interesse zu suchen.

Es werden nachstehend nicht alle Unterseiten beschrieben, sondern einführend nur einzelne Verknüpfungen näher erklärt:

Unter dem Hyperlink *Data Structures* werden allgemein Grundbegriffe wie *LISP*-Atom oder S-Expression erklärt, wobei neben den Definitionen eigene Übungen möglich sind.

Basic Operations klärt Begriffe wie "Befehl" oder "Operation", des weiteren erfolgt eine Vorstellung von spezifischen Operationen, die zwingend auf Listen operieren.

Des weiteren ist unter *Arithmetics* ein Überblick zu möglichen arithmetischen Funktionen möglich, wogegen das Prinzip der Rekursion – ein bedeutendes Grundgerüst der Programmiersprache – über *Recursion* erklärt wird.

Ein Test kann über die Indexseite beliebig durchgeführt werden, wobei eine Anmeldung über einen Namen gefordert wird; die Fragen sind hauptsächlich nach dem Multiple-Choice-Verfahren aufgebaut, jedoch werden auch kleine Programmkonstrukte vom Nutzer gefordert. Nach Beendigung des Tests und Betätigen eines Buttons erwartet man eine zügige Kontrolle durch das Programm, jedoch erfolgte nach Absenden der Antworten generell eine Fehlermeldung.

2.3 Fazit

Der Vorteil eines computergestützten (Online-)Lernsystems liegt oft darin, dass ein dargestellter Lernstoff schneller überschaubar ist und möglicherweise interessanter wirkt. Und ist ein Thema auf dem neuesten Stand und wirkt ansprechend, kann mit Lernerfolgen gerechnet werden.

Leider wird der "LISP-Tutor" diesen Erwartungen nicht gerecht.

Das Programm basiert hauptsächlich auf Text und enthält nur eine unmodern wirkende Eingangsgrafik, die zwar eine Themenauswahl über Links bietet, aber gerade auch wegen der großen Anzahl von Verknüpfungen im ersten Moment wenig überschaubar wirkt. Es hätten sich Frames mit einer Pfad-Übersicht angeboten, so dass die Suche nach einem Lernweg wegfällt und die Desorientierung vermieden wird.

Der Nutzer wird die Sequenzen abarbeiten und während des Studiums offensichtlich Probleme haben, das Interesse wegen der eintönigen Ausführungen aufrechtzuerhalten.

Des Weiteren vermisst man die Interaktion zwischen System und Benutzer (per E-Mail oder Chat), so dass bei Problemen keine tutorielle Betreuung erwartet werden kann. Auch besteht nicht die Möglichkeit, sich mit anderen Nutzern in Diskussionsforen auszutauschen

Auf den einzelnen Seiten kann man zwar Fragen beantworten und auch einen abschließenden Test starten, aber es ist nicht sichergestellt, dass man mit dem Erfahrenen umgehen kann. Demzufolge ist es nicht leicht ersichtlich, welchem Anspruch der "LISP-Tutor" gerecht werden soll und ob ein objektiver Einstieg in die Programmiersprache gefunden wurde.

Zusammenfassend kann gesagt werden, dass dieses Online-Lernsystem einen Überblick über offensichtlich grundlegende Begriffe und die Syntax von *LISP* bietet, jedoch weiterführende Erklärungen fehlen. Es existieren z.B. keine Hinweise, wie man auf seinem Rechner mit der Programmiersprache arbeiten kann, was eben einem Anfänger-Zielpublikum nicht unbedingt klar ist

Somit kann dieses Programm nicht als optimales Lehrsystem eingestuft werden, sondern dient eher als informationelles System.

3. Vorstellung des Online-Lernsystems " ELM – ART "

3.1. Einleitung

ELM-ART ist ein interaktives Lernprogramm, das speziell für die Programmiersprache LISP entwickelt wurde. Der Begriff eine Abkürzung für EPISODIC LEARNER MODEL – the ADAPTIVE REMOTE TTUTOR. Das bedeutet übersetzt: episodisches Lern-Modell – der adaptive Fernlehrer. Das System ist unter folgender Adresse im Internet zu finden:

<http://www.apsymac33.uni-trier.de:8080/Lisp-Kurs>

ELM-ART ist zweisprachig konzipiert; es lässt sich zwischen Deutsch und Englisch wählen. Die Benutzung ist völlig frei, d.h. es entstehen für den Nutzer keine Kosten. Jeder Benutzer – unabhängig davon, ob er Anfänger ist oder schon mit dem System gearbeitet hat – muss sich für jede Sitzung mit Benutzernamen und Kennwort anmelden. Die Anmeldung ist notwendig, weil die Identifizierung des Benutzers die wichtigste Voraussetzung für adaptives Verhalten ist. Das System erzeugt für jeden Benutzer ein individuelles Modell, das seine Interaktionen mit dem System festhält und als Grundlage für Benutzerführung und individuelle Aufgaben- und Beispielvorschläge dient. Passwörter dienen dem Datenschutz und der sicheren Zuordnung von Benutzernamen zu Benutzern.

ELM-ART wurde entwickelt von Prof. Dr. Gerhard Weber an der Pädagogischen Hochschule Freiburg.

Er wurde dafür mit dem EASA98-Award in Oxford am 21-09-1998 ausgezeichnet. Kriterien für diese Auszeichnung waren unter anderem Innovation, Design, leichte Handhabung, lernerische Qualität und Evaluation.

Idee

Die Grundidee für die Entwicklung entsprang dem Gedanken, die Vorteile der Lernumgebung ELM-PE für das Internet verfügbar zu machen. Für den Benutzer reduzieren sich somit die Anforderungen bezüglich Hardware und Software. Daraufhin ist eine Benutzung jederzeit möglich, was ein besseres Training gewährleistet. Außerdem wird durch die Interaktivität des Systems das persönliche Lernumfeld des Lernenden stärker miteinbezogen.

Ziel

Das Ziel von ELM-ART ist es, eine erste Einführung in die Programmiersprache LISP zu geben, damit der Nutzer danach selbstständig mit Büchern etc weiter lernen und sein Wissen vertiefen kann.

Durchführung

Es gibt verschiedene Lektionen zu den wichtigsten Aspekten von LISP mit einem vorherigen Eingangstest, den der Nutzer wählen kann, wenn er schon Erfahrung mit LISP oder allgemein mit funktionalen Programmiersprachen hat. Vor jeder Lektion steht eine kurze Beschreibung zu dem Inhalt, danach folgt dieselbe. Zu jedem Abschnitt werden Übungsaufgaben angeboten, die dem Lernenden helfen sollen, das Erarbeitete anzuwenden.

ELM-PE

ELM-PE ist ein Sprachtutor für Common LISP. Der Prototyp war gedacht als Begleitung für Studenten während eines Programmierkurses für LISP. Lektionen werden in einem Klassenraum erteilt, so dient das System als Hilfe bei den Hausaufgaben. ELM-PE bietet jetzt als ausgereiftes Lernsystem für Anfänger in der Programmiersprache folgende Möglichkeiten:

- Einführung in das System und Bedienanleitung in kurzen Videosequenzen
- individualisierte Auswahl und Erklärung anhand von Beispielen
- automatische Analyse von Lösungen
- Erklärung von Fehlermeldungen
- aktive Unterstützung bei der Programmierung mit dem syntaxorientierten Editor ALFRED

Lektionen

Die Lektionen 1-3 beschäftigen sich mit der Programmierung von elementaren LISP Funktionen, während die Lektionen 4-6 das Konzept der rekursiven Funktionen als typisches Konstrukt der Sprache beinhalten. Die Lektionen sind folgendermaßen unterteilt:

1. Datentypen + Funktionen
2. Listenfunktionen
3. Prädikate
logische Funktionen
4. endrekursive Funktionen
5. Top-Level-Rekursion
6. Baumrekursion

Auf der letzte Seite werden Fragen zu ELM-ART gestellt. Hier hat der Benutzer die Möglichkeit, das Lernsystem einzuschätzen, seine persönliche Meinung dazu zu schreiben und Verbesserungsvorschläge anzubieten.

Darstellung

Es gibt nachfolgende drei Arten der Darstellung:

	Der Text des Links erscheint in normaler Schrift. Dieser Ball zeigt an, dass diese Seite bereits besucht wurde.
	Der Text des Links erscheint in normaler Schrift. Dieser Ball zeigt an, dass diese Seite noch nicht besucht wurde.
	Dieses Icon zeigt an, dass es sich um eine Sektion mit untergeordneten Seiten handelt. Die Annotation erfolgt wie oben.

Übungen

Es werden verschiedene Typen von Aufgaben eingesetzt.

Die einfachste Art ist Single Choice, wo nur eine Auswahl richtig ist, z. B. Fragen, die mit ja oder nein beantwortet werden.

Dann gibt es noch Multiple Choice Aufgaben, wo mehrere Lösungen möglich sind. Weiterhin besteht die Möglichkeit, Aufgaben per Texteingabe zu lösen, z. B. soll der Nutzer angeben, was bei Aufruf einer bestimmten Funktion als Ausgabe erscheint.

Als komplexeste Art von Aufgaben steht ein Programmierfenster zur Verfügung. Hierbei gibt der Nutzer ein Programm ein und das System führt mit Tipps zur richtigen Lösung.

3.2 Erläuterungen der praktischen Arbeit

Wenn man sich angemeldet hat, erscheint als erstes eine Seite mit folgendem Aufbau:

[Manual](#)

[Tutor](#)

[Hilfe](#)

[Modell](#)

[Einstellungen](#)

[Inhalt](#)

[Suche](#)

[Notiz](#)

[LISP](#)

[EVAL](#)

➔ LISP-Kurs
 [Lektion 1](#)
 [Lektion 2](#)
 [Lektion 3](#)
 [Lektion 4](#)
 [Lektion 5](#)
 [Lektion 6](#)
 [Letzte Seite](#)

Falls Sie bereits über Vorkenntnisse verfügen:
[Eingangstests zu "LISP-Kurs"](#)

LISP-Kurs

Dieser LISP-Kurs gibt eine erste Einführung in die Programmiersprache LISP. Ziel des LISP-Kurses ist es, mit der Programmiersprache soweit vertraut zu werden, daß danach selbständig mit anderen Lehrbüchern weiter gelernt und das Programmieren in LISP vertieft werden kann.

Dieser Einführungskurs umfaßt sechs Lektionen. Die ersten drei Lektionen führen in die Programmierung mit elementaren LISP-Funktionen ein. In den folgenden drei Lektionen wird das Programmieren rekursiver Funktionen geübt. Rekursives Programmieren ist typisch für die Programmiersprache LISP.



[ELM-ART-5.9](#) ist ein Projekt der [ELM Research Group](#) basierend auf [CL-HTTP](#)
Bei Problemen Email an [ELM admin](#) --- 28.02.2001 09:56:22

Wenn ein Lernender sich entscheidet mit Hilfe eines Online-Lernsystems zu lernen, hat derjenige im allgemeinen die Erwartung, dass er eine gute Einsicht in die Programmiersprache bekommt. Neben den speziellen Informationen werden sicherlich auch allgemeine Erklärungen zu funktionalen Programmiersprachen gewünscht.

ELM-ART bietet die Informationen in einer übersichtlichen, kompakten Form an. Es gibt verschiedene Icons zur Benutzerführung. Dabei ist auf den ersten Blick ganz gut ersichtlich, was sich dahinter verbirgt:

- Manual: bietet eine kurze Einführung in den Umgang mit dem Browser.
- Tutor: zeigt einen Dialog zur Sendung einer E-Mail an. Hier

- bei gibt es die Möglichkeit, Fragen zu stellen oder Mitteilungen/Anregungen zu senden.
- Hilfe: Dahinter verbergen sich hilfreiche Texte zu Lektionen und Aufgaben.
 - Modell: Hier kann der Nutzer modifizieren, welche Lektionen er überspringen möchte.
 - Einstellungen: Hier hat der Nutzer die Möglichkeit, das Layout anzupassen und die Sprache (Deutsch/Englisch) zu wechseln.
 - Inhalt: zeigt in einer Übersicht, welche Übungen schon bzw. noch nicht gemacht wurden.
 - Suche: bietet die Suche nach einer Lektion zu einem bestimmten Stichwort an.
 - Notiz: Hier kann der Nutzer persönliche Notizen machen.
 - LISP: zeigt die Konstrukte der Sprache.
 - Eval: ist ein LISP-Interpreter zum Testen eigener Ideen.
 - Kommunikation: Chatroom
Diskussion
Tutor
 - LISP-Konstrukte: zeigt die Konstrukte der aktuellen Lektion an.

Darunter stehen die Lektionen jeweils mit kurzer Beschreibung, was durchgenommen wird. Für Nutzer mit Vorkenntnissen bietet sich hier die Möglichkeit, einen Eingangstest zu machen. Danach können die einzelnen Lektionen in Angriff genommen werden. Hierbei sind wichtige Ausdrücke groß geschrieben. Dazu gibt es immer kleine Beispiele, worauf die Übungen folgen. Es besteht die Möglichkeit, zu jeder Übung Hilfe anzuklicken. Die Auswertung zu den Übungen erfolgt mit kurzer Fehlerbeschreibung. Oben in der Grafik wird angezeigt, wie viele Übungen dieses Teils der Lektion schon gelöst wurden.

3.3 Fazit

Vorteile/Nachteile

Das System ELM-ART bietet als Online-Lernsystem für den Nutzer gute Möglichkeiten, sich in die Anfänge der Programmiersprache LISP einzuarbeiten. Das relativ umfangreiche Aufgabengebiet ist in kleine, überschaubare Lektionen gegliedert, was einen leichten Einstieg in die Sprache ermöglicht. Durch die Anmeldung ist eine individuelle Anpassung an jeden Nutzer gegeben. Die Seiten bieten ein ansprechendes Layout, das es dem Benutzer leicht macht sich zurechtzufinden. Die Lektionen und Hilfestellungen sind recht ausführlich erklärt. Trotzdem sind die Textpassagen nicht so lang, dass der Nutzer dazu verleitet wird, diese nicht zu lesen. Antworten zu eventuell auftretenden Fragen können wahlweise per E-Mail zugeschickt werden. Außerdem besteht die Möglichkeit, sich zu jeder Lektion Notizen zu machen.

Trotz der Vorteile gibt es noch Dinge, die verbessert werden könnten, vor allem treten öfter Probleme beim Zugriff auf die Seite auf. Weiterhin habe ich kein Fenster gefunden, wo die Antworten auf Fragen erscheinen, wenn ich keine E-Mail Adresse angebe. Dabei dauert es teilweise sehr lange bis eine Antwort erscheint. Die Interaktion mit den Tutoren ist also sehr zeitaufwendig. Die Auswertung der Aufgaben ist teilweise etwas unverständlich. Die Übungen lassen sich nicht wiederholen, wenn sie falsch waren, aber bei Bedarf können weitere Aufgaben gelöst werden.

Zusammenfassung ELM-ART

Ich persönlich denke, dass ELM-ART ein gutes System ist, um die Anfänge der Programmiersprache LISP zu erlernen. Allerdings bin ich der Meinung, dass es das Lernen mit Hilfe von Büchern nicht ersetzt. Es bietet allerdings einen guten Einstieg. Trotzdem sollte man seine Studien noch vertiefen und anderweitig selbstständig weiterlernen.

4. Resümee

Vergleich Lernen im Kurs / Online-Lernsysteme

Zum Schluss möchten wir noch einmal kurz einen Vergleich zwischen dem Lernen im Kurs und dem Lernen mit Online-Lernsystemen ansprechen und mögliche Vor- und Nachteile erläutern.

Das Lernen im Kurs bietet den Vorteil, dass die direkte Interaktion mit dem Lehrer und anderen Lernenden möglich ist. Diese Option ist beim Online-Lernen theoretisch auch gegeben. Es können z. B. Fragen per E-Mail gestellt werden oder man kann sich im Chat mit anderen Nutzern unterhalten. Ebenso gibt es ein Diskussionsforum, das einen Gedankenaustausch zu bestimmten Themen bietet. Allerdings ist es recht unwahrscheinlich, genau zu der Zeit, in der man online ist, jemandem im Chat zu treffen. Die Interaktion mit dem Tutor per E-Mail dauert auch eine gewisse Zeit. Das Online-Lernen dagegen hat den Vorteil, dass jeder sich seine Zeit individuell einteilen kann, was in einem Kurs nicht möglich ist. Da sind feste Zeiten gegeben. Außerdem gibt es oftmals wenig Spielraum bezüglich der Stoff- und Zeiteinteilung, da bestimmte Lektionen in einem begrenzten Zeitraum geschafft werden müssen. Online kann jeder entsprechend seinen eigenen Möglichkeiten lernen. Allerdings ist die Menge an Informationen in dem System begrenzt und es gibt dann einen Punkt, an dem es keinen Wissenszuwachs mehr bietet. Außerdem können technische Probleme auftreten.

Ich bin der Meinung, dass jede Art des Lernens Vor- und Nachteile bietet. Diese muss jeder für sich abwägen und dann entscheiden, welche Art des Lernens persönlich bevorzugt wird.

Zusammenfassung

Eigenes Lernen erfordert immer Selbstdisziplin vom Lernenden. Auch ein Online-Lernsystem bietet in dieser Hinsicht nicht mehr Vorteile. Es kann zusätzlich helfen, aber nicht das Lernen in einem Kurs bzw. mit Büchern ersetzen. Das Lernen mit mehreren Systemen bringt meist auch keine Vorteile, da es bisher nur wenige Systeme zu einem Problem gibt bzw. sich die Systeme oftmals nicht direkt vergleichen lassen. Meiner Meinung nach hat das Lernen in einem Kurs bisher mehr Vorteile, allerdings bietet ein Online-Lernsystem die Möglichkeit, zuhause das Wissen zu vertiefen.

5. Literaturverzeichnis

Bücher

Brockhaus	diverse Bände
Schulze, Hans Herbert	"PC-Lexikon – Fachbegriffe schlüssig erklärt", rororo 1996
Remboldt, U., Levi, P.	"Einführung in die Informatik für Naturwissenschaftler und Ingenieure", Carl Hanser Verlag 1999

Internet

LISP-Tutor	http://www.cs.tulane.edu/www/Villamil/lisp/lisp1.html
Allgemeines zu <i>LISP</i>	http://ki.cs.tu-berlin.de/support/lisp.html http://wbt-3.iicm.edu/product
Psychologie	http://www.toppoint.de/~timfly/dpl/Inhaltsverzeichnis.html
Nutzen von Hypermedia-LS	http://iseran.ira.uka.de/~zundel/seminar97/lernsw.html
John McCarthy	http://www-formal.stanford.edu/jmc/index.html
ELM-ART	http://apsymac33.uni-trier.de:8080/Lisp-Kurs
-in Englisch	http://apsymac33.uni-trier.de:8080/Lisp-Course
ELM-ART Info	http://cti-psy.york.ac.uk/easa98/ELM_ARTinfo.htm