

Eckart Modrow, StD
Tel.: 0551-397980
Mail: emodrow@astro.physik.uni-goettingen.de

Universitäts-Sternwarte
Geismarlandstr. 11
37083 Göttingen

Die Virtuelle Lehrerweiterbildung Informatik in Niedersachsen (VLIN) - Kursprogramm -

Das Kursprogramm orientiert sich bewusst nicht an der inhaltlichen Semesterfolge der Universität, sondern an einer üblichen Kursfolge Informatik in der Sekundarstufe II. Den Kursteilnehmer/innen wird dadurch die eigenständige Aufteilung und Umsetzung der Informatikinhalte auf Schulkurse erspart. Weiterhin sollte hierdurch in den LWB-Kursen der direkte Bezug zum geplanten Unterricht erleichtert werden. Grundlage für die Prüfung sind die im Netz veröffentlichten Inhalte der Grundvorlesung Informatik von Prof. Schaback. Der Bezug der Kursinhalte zu den Vorlesungsterminen ergibt sich im Detail aus der beiliegenden Liste.

Voraussetzungen: Elementare Computerkenntnisse

- Nutzung einer grafischen Oberfläche (Windows, KDE, ...)
- einfacher Umgang mit Dateien (Öffnen, Speichern, Ausführen, Editieren, Drucken, ...)
- Textverarbeitung

Vorbereitungswoche:

- Übersicht über die Kursfolge und die Materialien
- Elemente einer IDE (Editor, Compiler, Debugger, ...)
- Programmentwicklungszyklus anhand einfacher Beispiele
- Testen von Programmen
- Einfache Grafikprogramme (Applets), Nutzung der Bibliotheken
- Einführung ereignisgesteuerte Programmierung/OOP (Objekte, Klassen, Kapselung, Vererbung, ...)

1. Kurs: Einführung in die Programmierung mit Java

Ziele:

Die Teilnehmer/innen sollen

- den Umgang mit integrierten Programmentwicklungssystemen kennen lernen.
- eine Einführung in das Programmieren erhalten, die sich anfangs eng an den Einführungsunterricht Informatik anlehnt, dann aber zur eigenständigen Arbeitseinteilung überleitet.
- Grafikprogrammierung sicher beherrschen.
- Zeichenkettenverarbeitung sicher beherrschen.
- anfangs überwiegend die Elemente einer objektorientierten Sprache nur anwenden, anschließend selbst OOP-Systeme entwickeln.
- sich mit einer OOP-Sprache (Java) intensiv vertraut machen, aber parallel dazu eine andere soweit betreiben, dass sie nicht auf nur ein System festgelegt werden.

Arbeitsmittel:

- Ein freies JAVA-Programmentwicklungssystem, z.B. FORTE (von Sun), JBuilder (von Borland) oder BlueJ sowie Visual J++
- Die freie Testversion von Delphi (von Borland), weil Pascal an den Schulen immer noch weit verbreitet ist.
- Notfalls kann als zweites System auch eine andere Sprache verwendet werden (Prolog, C++, ...) – je nach Bedürfnissen der Schule.
- Kursmaterialien im Internet
- Aufgaben im Internet
- Literaturliste

Inhalte während des Semesters:

- Erzeugung statischer Grafiken (Zufallsgrafik, Funktionsplotter, Malprogramm, ...)
- bewegte Grafik mit Timern, Threads, ... (Simulation dynamischer Vorgänge, Bewegungen von Teilchen, Populationsdynamik...)
- Ein- und Ausgabe auf dem Bildschirm
- statische Datentypen (Zahlen, Zeichenketten, Felder, Verbunde)
- Umgang mit Zeichenketten (Eingabe von Funktionstermen, Mustererkennung (DNA, ...), Kryptographie, elementare Computeralgebra („Differenzieren“), ...)
- Verifikation von Programmen, Korrektheitsbeweise
- Entwurf einfacher Objektklassen (Kartenspiele, Physikgeräte, ...)
- Bearbeitung von Hausaufgaben, darunter eine Übungsklausur

Präsenztage:

- ggf. Klausuren
- Vergabe der Themen für Hausaufgaben
- Vorträge zum Thema: z.B.: Gen-Sequenzen, „richtige“ Computeralgebra, Optimierungsverfahren, ...
- dynamische Datenstruktur Liste und ihre Anwendungen
- abstrakte Datentypen
- ER-Modell und Relationen
- Vorstellung der Hausarbeiten

2. Kurs: Datenstrukturen und Datenbanken

Ziele:

Die Teilnehmer/innen sollen

- dynamische Datenstrukturen kennen und anwenden lernen.
- rekursive Verfahren entwickeln, die auf dynamischen Datenstrukturen arbeiten.
- abstrakte Datentypen sowie unterschiedliche Verfahren zu ihrer Implementierung kennen und anwenden lernen.
- theoretische und experimentelle Effizienzuntersuchungen machen.
- Methoden zum Entwurf relationaler Datenbanken anwenden.
- Anfragen an relationale Datenbanksysteme stellen.
- gesellschaftliche Auswirkungen von Computeranwendungen kennen lernen.

Arbeitsmittel:

- Ein freies JAVA-Programmentwicklungssystem (s.o.)
- Die freie Testversion von Delphi (s.o.)
- Kursmaterialien im Internet
- Aufgaben im Internet, Literaturliste

Inhalte während des Semesters:

- dynamische Datenstrukturen (Liste, Stapel, Baum, ...), rekursive Verfahren
- abstrakte Datentypen und deren Implementierung
- Effizienz von Algorithmen (Sortierverfahren, Suchen in Bäumen und allg. Graphen, ...)
- einfache Dateibearbeitung (Dateitypen, Schreiben und Lesen, Indexdateien, ...)
- Dateiorganisationsformen (Hashdateien, Bayerbäume, ...)
- Das Entity-Relationship-Modell (ER-Modell)
- Relationen, relationale Algebra
- relationale Datenbanken (DDL, DML, ...) und (Local-)SQL
- gesellschaftliche Auswirkungen, Datenschutz
- Bearbeitung von Hausaufgaben, darunter eine Übungsklausur

Präsenztage:

- ggf. Klausur über das Kursthema, Vergabe der Themen für Hausaufgaben
- Vorträge zum Thema: z.B.: Anwendungen von Datenbanken, Sicherheit in Netzen, ...
- Besuch einer Firma/Institution mit Datenbankanwendungen
- endliche Automaten mit Ausgabe
- Beschreibung von Schaltwerken
- logische Grundschaltungen
- Zahlendarstellungen
- je nach zeitlicher Lage: Vorstellung der Hausarbeiten

3. Kurs: Technische Informatik

Ziele:

Die Teilnehmer/innen sollen

- endliche Automaten zur Beschreibung von Schaltwerken einsetzen.
- Schaltungen systematisch entwerfen.
- Schaltungen und Schaltelemente simulieren bzw. Simulationsprogramme einsetzen.
- aus Grundbausteinen (programmierbare) Rechenwerke zusammenstellen.
- den Aufbau von Computern und die Abläufe in ihnen kennen lernen.
- die Aufgaben von Betriebssystemen kennen.
- einfache Assemblerprogramme (z.B. für die Java-VM, ...) schreiben.

Arbeitsmittel:

- Freie Programmentwicklungssysteme (s.o.)
- Ein freier Logiksimulator
- ggf. Bausatz digitale Elektronik (Platine, Netzgerät und ICs) ca. 150,- DM
- Kursmaterialien und Aufgaben im Internet, Literaturliste

Inhalte während des Semesters:

- Schaltwerke und Automaten
- Realisierung von Automaten durch Programme
- Grundgatter, Rechenschaltungen
- Simulation von Schaltungen
- Systematischer Schaltungsentwurf, Vereinfachung von Schaltungen
- Speicher, Zähler, Programmierbare Rechenwerke
- Aufbau von Computern, Funktionsgruppen, Arbeitsabläufe
- Aufgaben von Betriebssystemen
- einfache Assemblerprogrammierung
- Bearbeitung von Hausaufgaben, darunter eine Übungsklausur

Präsenztage:

- ggf. Klausur über das Kursthema, Vergabe der Themen für Hausaufgaben
- Vorträge zum Thema: z.B.: Datenübertragung im Internet, Messwerterfassung, Treiber, ...
- Besuch einer Firma/Institution mit Messwertanwendungen (z.B.: mit CCDs, ...)
- erkennende Automaten
- ein einfacher Parser

4. Kurs: Theoretische Informatik

Ziele:

Die Teilnehmer/innen sollen

- erkennende Automaten zum Entwurf von Parsern nutzen.
- geschachtelte Strukturen mit Kellerautomaten analysieren und deren Bedeutung für reale Computersprachen kennen lernen.
- wissen, welche Bedeutung Turingmaschinen für theoretische Untersuchungen haben.
- Automaten den entsprechenden Sprachen zuordnen.
- wissen, dass es unterschiedliche Klassen von Computersprachen gibt.
- lernen, wie neuronale Netze funktionieren.
- sich mit den Grenzen der Computer beschäftigen.

Arbeitsmittel: (s.o.)

Inhalte während des Semesters:

- Einfache Parser („LOGO für Arme“, HTML-Interpreter, ...)
- Äquivalenz von Automaten
- Geschachtelte Strukturen, Kellerautomaten
- Syntaxanalyse kontextfreier Sprachen (Parsingtabellen, LR(n)-Sprachen, ...)
- Entwurf und Analyse von Computersprachen (BNF-Form, ...)
- Neuronale Netze
- Berechenbare Funktionen
- Turingmaschinen
- Berechenbarkeit, Entscheidbarkeitsprobleme
- „Grenzen“ der Computer (numerische G., zeitliche G., G. der Algorithmisierbarkeit, G. der Modelle, ethische G., G. der Zuverlässigkeit, ...)