UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR THEORETISCHE INFORMATIK

PROF. DR. TILL TANTAU
WWW.TCS.UNI-LUEBECK.DE
27. MÄRZ 2014

Veranstaltungskarte
Einführung in die Informatik 1 und 2 – 2013 / 2014

INFORMATIK

Veranstaltungsziele
- Informationsverarbeitende Systeme theoretisch verstehen
- Informationsverarbeitende Systeme in Forschungs- und Entwicklungsprojekten benutzen können
- Algorithmen und Datenstrukturen verschiedenen Projektbedürfnissen anpassen können
- Vorbereitung auf vertiefende Informatikveranstaltungen

Informatik-Einführung

- 21 Einführung, Organisatorisches
- 24 Informationen und Daten
- 29 Hardware und Software
- 31 Betriebssysteme

November 2013

- 5 Shells
- 7 Reguläre Ausdrücke
- 12 LaTeX
- 14 Hypertext Markup Language
- 19 Der Algorithmusbegriff
- 21 Imperative Programmierung
- 25 Die Programmiersprache Java
- 28 Steuerungsanweisungen und elementate Datentypen

Dezember 2013

- 2 Zeichenketten
- 5 Arrays (Felder)
- 9 Scoping
- 12 Modularisierung im Kleinen

Algorithmik

- 17 Suchalgorithmen
- 19 Sortieralgorithmen

Januar 2014

- 7 Komplexität von Problemen
- 9 Einführung zur Rekursion
- 14 Schnelle Sortierverfahren und Rekursion
- 16 Modularisierung im Großen

Modellierung

- 21 Objekte – Attribute und Lebenszyklus
- 23 Objekte – Methoden und Nachrichten
- 28 Objektorientierte Modellierung
- 30 Listen – Konzepte und Erstellung

Datenstrukturen

Februar 2014

- 4 Listen – Iteration und Modifikation
- 6 Bäume
- 11 Suchbäume
- 13 Hashing

Programmieren in Java

- 10 Scoping
- 12 Modularisierung im Kleinen

Schwierige Probleme

- Graphen
- Algorithmendesign
- Kürzeste und längste Wege
- Approximationsalgorithmen

Mai 2014

- Berechenbarkeit

Datenbanken

- Einführung zu Datenbanken
- SQL – Modellierung
- Relationenalgebren

Juni 2014

- SQL – Einfache Anfragen
- SQL – Fortgeschrittene Anfragen
- SQL – Anbindung an Java
- Fallstudie zu Datenbanken

Juli 2014

- Biodatenbanken

Sicherheit

- Kommunikations-Sicherheit
- System-Sicherheit

Mathematik $\sum_{i=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$

Linguistik (Satz: Subjekt – ist – Das – Sein)

Theoretische Informatik
- Maschinenmodelle
  - endliche Automaten
  - Turing-Maschine
  - kontextfrei
  - regulär
  - Grammatiken
  - reguläre Ausdrücke
- Sprachen
- Theorie formaler Sprachen
- Komplexitätstheorie
- Ressourcen
- Klassen
- Kodierungstheorie
- Kryptologie
  - digitale Wasserzeichen
  - Kryptoanalyse (Codebrechung)
  - Verschlüsselung
    - symmetrisch
    - asymmetrisch

Technische Informatik
- Elektrotechnik
- Maschinenbau
- Robotik
- Schaltkreise
- Prozessoren
- Peripherie
  - Maus
  - Bildschirm
  - Drucker
  - Tastatur
  - Netzwerkkarte
- Netzwerke
  - Internet
  - Funk
    - Bluetooth
    - Wirless LAN
  - Kabel
    - DSL
    - ISDN

Praktische Informatik
- algorithmische Methoden
  - Optimierung
    - Backtracking
    - Dynamische Programmierung
    - Greedy-Heuristiken
    - Rekursion
  - Graph-algorithmen
  - Sortieren
  - Suchen
- Softwaretechnik
  - Planung
  - Modularisierung
  - Strukturierung
  - Modellierung
- Programmier-sprachen
  - imperativ
    - C
    - Assembler
    - Pascal
  - objektorientiert
    - Java
    - C++
    - Smalltalk
  - funktional
    - logisch
      - ML
      - Scheme
      - Prolog
    - Wirtschaft €

Angewandte Informatik
- Datenstrukturen
  - Bäume
  - Hash-Tabellen
  - Arrays
  - Graphen
  - Listen
  - zusammen-gesetzte
  - einfache
    - Strings
    - Zahlen
- Datenbanken
  - Datenbank-Systeme
    - DB2
    - MySQL
  - Theorie
    - ER-Diagramme
    - Relationen-Algebren
  - Datenbank-Sprachen
    - Konkrete Datenbanken
    - SQL
- Dokument-beschreibung
  - HTML
  - Style Sheets
  - LaTeX
- Betriebssysteme
  - Unix
  - Shells
  - Windows
- Anwendung
  - Word
  - Mail
  - Matlab

---

1 Informationen und Daten
- Die binäre Speicherform verschiedener Daten kennen und verstehen.
- Wichtige Einheiten von Information kennen und die Speichergröße von Daten abschätzen können.
- Beliebige Daten kodieren können.
- Das Konzept der Datenkompression kennen und verstehen.

2 Hardware und Software
- Klassen und Aufbau von Computern kennen und ihre Leistungsfähigkeit abschätzen können.
- Die wichtigsten Hardwarekomponenten kennen und ihre Leistungsfähigkeit abschätzen können.
- Software als Instruktionsfolgen begreifen
- Das Schichtenkonzept kennen und die Schichten benennen können

3 Betriebssysteme
- Aufgaben von Betriebssystemen kennen
- Konzepte der Datei-, Benutzer- und Prozessverwaltung kennen
- Mit Dateien, Pfaden und Dateirechten umgehen können
- Aufgaben von Treibern erläutern können

4 Shells
- Unix- und Programm-Shells benutzen können
- Wichtige Unix-Kommandos kennen
- Einfache Shell-Skripte erstellen können

5 Reguläre Ausdrücke
- Konzept des regulären Ausdrucks verstehen
- Probleme als regulärer Ausdrücke beschreiben können
- Mächtigkeit regulärer Ausdrücke kennen

6 LaTeX
- Unterschied zwischen Form, Inhalt und Struktur kennen
- Benutzung von LaTeX verstehen
- Entscheiden können, wofür LaTeX geeignet ist

7 Hypertext Markup Language
- Die Grundsyntax von html kennen
- Die Grundsyntax von xml kennen
- Eigene html-Seiten erstellen können

8 Der Algorithmusbegriff
- Den Begriff des Algorithmus verstehen
- Einfache Lösungsideen als Algorithmen formulieren können
- Probleme spezifizieren können
- Programmiersprachen als Kommunikationsmittel für Algorithmen begreifen
- Das Konzept des Übersetzers verstehen

9 Imperative Programmierung
- Zuweisungen verstehen und benutzen können
- Grundlegende Steuerungsanweisungen verstehen

10 Die Programmiersprache Java
- Allgemeine Syntaxelemente wie Bezeichner, Ausdrücke und Kommentare von Java beherrschen
- Zuweisungsbasierte Algorithmen als imperative Java Programme formulieren können
- Ein Java-Programm selbständig erstellen und übersetzen können

11 Steuerungsanweisungen und elementate Datentypen
- Die Steuerungsanweisungen If-Then-Else, While-Schleife und For-Schleife benutzen können
- Variablendeklarationen verstehen und benutzen können
- Elementare Datentypen von Java kennen
- Typkorrektheit und Typfehler verstehen

12 Zeichenketten
- Zeichenketten (Strings) als Datentypen kennen
- Einfache Stringalgorithmen kennen und implementieren können
- Einen fortgeschrittenen Stringalgorithmus kennen

13 Arrays (Felder)
- Konzept des Arrays verstehen.
- Java Syntax von Arrays beherrschen.
- Java-Programme mit Array-Nutzung implementieren können.

14 Scoping
- Scoping verstehen und anwenden können.
- Die Speicherung von Variablen verstehen.

15 Modularisierung im Kleinen
- Das Konzept des Unterprogramms/Methode verstehen und anwenden können
- Java-Syntax der Modularisierung auf Methodenebene beherrschen

16 Suchalgorithmen
- Lineare Suche verstehen und implementieren können
- Binäre Suche verstehen und implementieren können
- Laufzeit der Verfahren vergleichen können
- Abschätzen können, wann sich Vorsortierung lohnt

17 Sortieralgorithmen
- Arten von Sortierproblemen kennen
- Bubble-Sort, Insertion-Sort, Selection-Sort verstehen und implementieren können

18 Komplexität von Problemen
- Das Laufzeitverhalten der wichtigsten Algorithmen kennen
- Das Laufzeitverhalten einfacher Programme abschätzen können
- Die Gruppierung von Problemen nach groben Laufzeitklassen verstehen

19 Einführung zur Rekursion
- Das Konzept der Rekursion verstehen
- Rekursive Methoden implementieren können für einfache Probleme

20 Schnelle Sortierverfahren und Rekursion
- Merge-Sort verstehen und implementieren können
- Quick-Sort verstehen und implementieren können

21 Modularisierung im Großen
- Das Konzept des Top-Down Entwurf verstehen und anwenden können
- Das Konzept der Klassen- und Modulbildung verstehen

22 Objekte – Attribute und Lebenszyklus
- Modellierung von Daten mittels Klassen verstehen
- Konzept des Objekts und des Attributs kennen und anwenden können
- Lebenszyklus von Objekten verstehen

23 Objekte – Methoden und Nachrichten
- Konzept der Nachricht kennen
- Nichtsozialsche Methoden implementieren können
- Konstruktoren kennen und implementieren können

24 Objektorientierte Modellierung
- Modellierung von Daten mittels Klassen verstehen
- Klassen zur Modellierung von Fallbeispielen erstellen können

25 Listen – Konzepte und Erstellung
- Die Liste als Datenstruktur verstehen
- Code zur Erzeugung von Listen erstellen können
- Den Unterschied zwischen Listen und Arrays erklären können

26 Listen – Iteration und Modifikation
- Code zur Iteration über Listen erstellen können
- Code zur Modifikation von Listen erstellen können

27 Bäume
- Das Konzept des Baumes verstehen
- Bäume in Java implementieren können
- Baumtraversierung kennen und implementieren können

28 Suchbäume
- Konzept des Suchbaums verstehen
- Basisoperationen auf Suchbäumen verstehen
- Vor- und Nachteile von Arrays, Listen und Suchbäumen beurteilen können

29 Hashing
- Konzept des Hashwerts verstehen
- Einfache Hashfunktionen kennen und implementieren können
- Einfache Hashverfahren kennen und implementieren können

30 Graphen
- Arten von Graphen kennen
- Daten und ihre Struktur mittels Graphen modellieren können
- Die Datenstruktur des Graphen verstehen
- Konzept des Graphproblems verstehen

31 Algorithmendesign
- Greedy-Verfahren kennen und implementieren können
- Backtracking-Verfahren kennen und entwerfen können

32 Kürzeste und längste Wege
- Konzept der Tiefen- und Breitensuche verstehen
- Dijkstra-Algorithmus für das Kürzeste-Wege-Problem kennen
- Backtracking-Algorithmus für das Längste-Wege-Problem kennen

33 Approximationsalgorithmen
- Die Konzepte Optimierungsproblem, Lösung und Maße verstehen
- Das Konzept der approximativen Lösung verstehen
- Approximationsalgorithmen für das Handlungsreisenden-Problem und das Bin-Packing-Problem kennen

34 Berechenbarkeit
- Mathematisches Modell für reale Computer kennenlernen
- Begriff Berechenbarkeit verstehen
- An Beispiele verstehen, dass Computer nicht alles berechnen können
- Aussage der Church-Turing-These kennen

35 Einführung zu Datenbanken
- Das Konzept der Datenbank kennen und verstehen
- Das E/R-Modell kennen
- Daten mittels des E/R-Modells modellieren können

36 SQL – Modellierung
- Das Konzept der Datenbank-Shell kennen
- Die MySQL-Shell benutzen können
- E/R-Diagramme in eine relationale Datenbank umwandeln können
- SQL-Datentypen kennen

37 Relationenalgebren
- Konzept der Relation-Algebra kennen
- Anfragen mittels Operationen formulieren können
- Joins verstehen und anwenden können

38 SQL – Einfache Anfragen
- SQL-Anfragen an eine Datenbank formulieren können
- Einfache SQL-Joins benutzen können
- SQL-Funktionen benutzen können

39 SQL – Fortgeschrittene Anfragen
- Fortgeschrittene SQL-Joins benutzen können
- SQL-Aggregate benutzen können

40 SQL – Anbindung an Java
- jdbc kennen und seine Aufgabe verstehen
- Verbindungen mit Datenbanken aus Java-Programmen heraus aufbauen können
- Daten mittels jdbc lesen können
- Daten mittels jdbc schreiben können

41 Fallstudie zu Datenbanken
- Datenbanken für eigene Programmentwicklungen nutzen können
- Daten aus verschiedenen Quellen in Datenbanken einfügen können
- Datenbanken zur Daten-Analyse einsetzen können.

42 Biodatenbanken
- Aufbau und Inhalter eine wichtigen Biodatenbank kennen
- Auf die Daten in dieser Datenbank zugreifen können

43 Kommunikations-Sicherheit
- Konzept der Verschlüsselung verstehen
- Unterschied zwischen symmetrischen und asymmetrischen Verfahren kennen
- Konzept der digitalen Unterschrift verstehen
- Programme zur Verschlüsselung einsetzen können
- Zertifikate erstellen und installieren können

44 System-Sicherheit
- Bedrohungsszenarien der Sicherheit von IT-Systemen kennen und einschätzen können
- Schutzmaßnahmen für die Sicherheit von IT-Systemen kennen und ergreifen können
- Beispiel eines Sicherheitslochs verstehen