

# Informatik 12

## Basisinformationen zu Informatik 12.

### Inf12

Grundbegriffe

Wdh Rekursive Datenstruktur Warteschlange und Liste

Wdh Baum als spezieller Graph

Wdh Graph

Formale Sprachen

Definition einer Formalen Sprache

Struktureller Aufbau einer Sprache (Alphabet, Terminalsymbole, Nichtterminalsymbole, leeres Wort)

Unterschied zwischen natürlicher Sprache und künstlicher Sprache

Unterschied zwischen Syntax und Semantik

EBNF (Erweiterte Backus-Naur-Form)

Syntaxdiagramme

Ableitungen

Syntaktischer Aufbau einer Sprache: Grammatik (Nichtterminale, Terminale, Regeln, Startsymbol)

vollständiger/ unvollständiger erkennender, (deterministischer) endlicher Automat als geeignetes Werkzeug zur

Syntaxprüfung für reguläre Sprachen

Implementierung eines erkennenden Automaten

Kommunikation und Synchronisation von Prozessen

Protokolle zur Beschreibung der Kommunikation von Prozessen

Netzwerk-Topologien

Schichtenmodell: Internet Protokollstapel (TCP/IP-Stack)

Internet als Kombination von Rechnernetzen

Definitionen

Parallele Prozesse

Nebenläufige Prozesse

kritischer Abschnitt

Verklemmung (Deadlock)

Modellierung einfacher, nebenläufiger Prozesse, z.B. mithilfe eines Sequenzdiagramms; Möglichkeit der

Verklemmung

kritischer Abschnitt: Monitorkonzept zur Lösung des Synchronisationsproblems

Umsetzung von Monitoren, z.B. in python

Funktionsweise eines Rechners/ Rechnerarchitektur

Aufbau eines Computersystems

Prozessor (Rechenwerk, Steuerwerk)

Arbeitsspeicher

Ein- und Ausgabeeinheiten

Hintergrundspeicher

Datenbus

Adressbus und Steuerbus

Registermaschine als Modell eines Daten verarbeitenden Systems

Datenregister, Befehlsregister, Befehlszähler, Statusregister

Arbeitsspeicher für Programme und Daten (von-Neumann-Architektur)

Adressierung der Speicherzellen

Binärsystem

Maschinensprache

Zustandsübergänge der Registermaschine als Wirkung von Befehlen

Systemnahe Programmierung und Umsetzung aller Kontrollstrukturen (z.B. mit MiniMaschine)

Grenzen der Berechenbarkeit

Laufzeit typischer Algorithmen und die damit verbundenen Grenzen der praktischen Anwendbarkeit

hoher Laufzeitaufwand als Schutz vor Entschlüsselung durch systematisches Ausprobieren aller Möglichkeiten

(Brute-Force-Verfahren)

OOM python

## Projektmanagement Medieninformatik

Quelle: angelehnt an: <http://www.mgf-kulmbach.de/neu/images/medien/unterrichtsfacher/Informatik - Susanne Ehmman>

Informationsseite Informatik:

[einstieg-informatik.de](http://einstieg-informatik.de)

Python-Programmierung lernen mit Übungen online:

[cscircles.cemc.uwaterloo.ca/using-website-de/](http://cscircles.cemc.uwaterloo.ca/using-website-de/)

## WDH

### UML

- Assoziation
- Aggregation
- Vererbung
- Objektkarte und Klassenkarte
- Zustandsdiagramme
- Sequenzdiagramme

### Fkt

- Datentypen
- Tabellen und Inhalte von Zellen
- Anpassung von Zellbezügen
- Vordefinierte Funktionen
- Zuordnungen und Funktionen
- Mehrstellige Funktionen
- Verkettung von Funktionen
- WENN-Funktion

### DB

- Tabellen und Schlüssel
- SQL-Abfragen
- Redundanz und Konsistenz
- Datenmodellierung
- Klassendiagramm mit Beziehungen und Kardinalitäten
- Relationales Datenbankmodell
- Integritätsbedingungen
- Join aus mehreren Tabellen
- Tabellen erweitern (Ergebniswerte)
- Aggregatfunktionen auf gruppierte Daten
- Datenmanipulation und Datenschutz
- Objekte und Klassen
- Algorithmus
- Zustände und Zustandsdiagramme
- Definition von Klassen
- Wertzuweisung
- Anlegen und Löschen von Objekten
- Implementieren von Algorithmen
- Felder

Endliche **Automaten** und allgemeine Zustandsdiagramme

Klassenbeziehungen  
 Implementierung von Klassenbeziehungen  
 Ober- und Unterklassen  
 Polymorphe Methoden  
 Felder

### Liste

Warteschlangen und Stapel  
 Rekursive Funktionen

### (Binär-)Baum

Traversieren eines Binärbaumes  
 Geordnete Binärbäume  
 Implementierung eines Binärbaumes  
 Einfügen eines Elementes in einen geordneten Binärbaum

### Graph

Wege durch Graphen  
 Repräsentation von Graphen  
 Suche in Graphen

### Agile Softwareentwicklung

Wasserfallmodell  
 Projektphasen  
 Aufwandsschätzung

### Prozesskommunikation

Semaphoren (wechselseitiger Ausschluss)

### Modellierungstechniken

Softwaremuster  
 Softwarequalität  
 Dokumentation von Software

opt OOM python  
 opt Projektmanagement

## Referate Medieninformatik I (5 min Vortrag, opt. im Team)

Zahlssysteme: Dezimal, Binär, Hexadezimal  
 Funktionsweise eines Rechners/ Rechnerarchitektur (Von-Neumann): Prozessor (Rechenwerk, Steuerwerk, Arbeitsspeicher, Ein- und Ausgabeeinheiten, Hintergrundspeicher, Datenbus, Adressbus und Steuerbus  
 Massenspeicher (von Terra- bis Peta- u. Exabyte-Bereich)  
 UML  
 Generalisierung und Spezialisierung  
 Programmiersprachen

### 'creative coding' - interaktive Grafik | Processing

#### Computergrafik 2D

#### digitale Typographie, (web-)Fonts

digitale Photographie, Post-Produktion Ph/ Video (Colourgrading)

#### 3D-Modellierung, Objekt-Design

### HTML5, XML, CSS

### JavaScript, jQuery

API (MPI, CUDA, Map-Reduce)  
 Multimedia  
 webRadio

webTV

ebooks

Mediengestaltung

## **UX-Design**

**User-Interface (Komponenten, Navigation)**

**Screen-Design und Storyboard in WebDesign (CMS)**

**Interaktive Grafik in CMS-Templates** (persona.co, cargo.site)

## **Sketching with Hardware**

**Arduino Boards**

**RaspberryPi**

**audio-Maschinen, UpDesign audio**

**FashTech**

Mikroprozessoren, Sensoren, Digital- Analog-Wandlung

Simulation in Physik, Elektronik, Mechanik

Betriebssysteme

Datenbanken DB (relationale und noSQL)

Performanz und Laufzeit

Software-Entwicklungstechniken (Agil, Phasen "Wasserfall": Analyse, Entwurf, Implementierung, Test, Bewertung und Abnahme)

Modell-View-Controller Konzept (MVC)

Software-Muster Kompositum (Composite Pattern)

Mensch-Maschine-Interaktion, KI

Iteration

Rekursion. Rekursive Datenstruktur Liste und Warteschlange

Spezialfälle der verketteten Liste - die Datenstruktur Stapel (Stack) und Schlange (Queue) (LIFO, FIFO)

Binäre Bäume, Baum als spezieller Graph

OSI-Schichtenmodell

Schichtenmodell: Internet Protokollstapel (TCP/IP-Stack)

Rechnernetze, Netzwerk-Topologien

Rechnernetze und GPS, Echtzeit, Mobilfunk, Steuerung

Internet als Kombination von Rechnernetzen

Cloud Computing

Computerarchitekturen (Superskalar, SIMD, Multi- u. Many-Core, Compute-Farm, GPU, Cell, Grid Computing)

Mobile und Verteilte Systeme

Big Data und Data Science

Forensic Architecture

Geo-Mapping (Heatmaps)

It-Sicherheit, Netzwerksicherheit

Kryptographie (Brute-Force-Verfahren, Laufzeit von Algorithmen und Grenzen der Berechenbarkeit)

Protokolle zur Datenübertragung

Protokolle zur Beschreibung der Kommunikation von Prozessen

Modellierung einfacher, nebenläufiger Prozesse, z.B. mithilfe eines Sequenzdiagramms; Möglichkeit der Verklemmung

Formale Sprachen (Alphabet, Terminalsymbole, Nichtterminalsymbole, leeres Wort; EBNF - Erweiterte Backus-Naur-Form)

Syntaktischer Aufbau einer Sprache: Grammatik (Nichtterminale, Terminale, Regeln, Startsymbol)

Erkennender endlicher Automat zur Syntaxprüfung regulärer Sprachen

Registermaschine als Modell eines Daten verarbeitenden Systems (Datenregister, Befehlsregister, Befehlszähler,

Statusregister. Arbeitsspeicher für Programme und Daten, Adressierung der Speicherzellen

Maschinensprache

Zustandsübergänge der Registermaschine als Wirkung von Befehlen

Systemnahe Programmierung und Umsetzung aller Kontrollstrukturen (z.B. mit MiniMaschine)

Compiler (Parsing, Zwischencode, Optimierung, Code-Generierung)

Debugger

Programm-Analyse und Typsysteme (Lambda-Kalkül, Polymorphie, Subtyping, Lineare- u. abhängige Typen, abstrakte Interpretation, Alias- u. Heapanalyse)

Semantic Web Rule Logic

Deklarative Programmiersprachen (Prolog, Datalog, OPS5)

Prolog als deklarative Programmiersprache (Trennung von Arbeits- u. Steuerungs-Algorithmus) + Logik

Multimediale Lehr- und Lernsysteme

Lerntheorien (Behaviorismus, Kognitivismus, Konstruktivismus, soziales Lernen, kooperatives Lernen)

Kreativität

Motivationstheorie

Games (Multiplayerspiele, Wegewahl)

GWAPs - Games with a Purpose, Serious Games

Wissenschaftliches Arbeiten, Recherche-Strategien, Zitieren

## **LPplus ISB Inf12 Lernbereich 1: Formale Sprachen und Endliche Automaten (ca. 16 Std.)**

### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- entwickeln formale Sprachen zu Beispielen aus dem Alltag (z. B. Autokennzeichen, E-Mail-Adressen oder Gleitkommazahlen), um ein Verständnis für die Notwendigkeit von klaren Regeln bei der Definition dieser Sprachen zu gewinnen und damit Mehrdeutigkeiten, wie sie in natürlichen Sprachen vorkommen, zu vermeiden.
- definieren formale Sprachen durch Grammatiken und verwenden zur Darstellung der Produktionsregeln insbesondere die Erweiterte Backus-Naur-Form (EBNF) und Syntaxdiagramme.
- entwerfen zur formalen Beschreibung von regulären Sprachen endliche erkennende Automaten.
- implementieren mithilfe einer objektorientierten Programmiersprache fachgerecht deterministische endliche Automaten und nutzen diese zur automatisierten Überprüfung der Zugehörigkeit von Wörtern zu einer regulären Sprache.
- erläutern an selbst gewählten Beispielen, dass es Sprachen gibt, die nicht regulär sind, und erkennen daran, dass es weitere Sprachkategorien in der Informatik gibt. Damit wird den Schülerinnen und Schülern bewusst, dass für die automatisierte Verarbeitung von nicht regulären Sprachen, wie z. B. höheren Programmiersprachen, das Modell des endlichen Automaten nicht ausreicht und weitere Modellkonzepte notwendig sind.

### **Inhalte zu den Kompetenzen:**

- Formale Sprache als Menge von Zeichenketten über einem Alphabet: Zeichen, Zeichenvorrat (Alphabet), Wort (Zeichenkette), Syntax, Semantik
- Grammatik: Terminal, Nichtterminal, Produktionsregel, Startsymbol
- Notation formaler Sprachen: Syntaxdiagramm und Erweiterte Backus-Naur-Form (EBNF)
- Ableitung eines Wortes einer formalen Sprache als Folge von Regelnwendungen, Ableitungsbaum
- erkennender endlicher Automat: Zustandsmenge, Eingabealphabet, Zustandsübergang, Startzustand, Endzustand, Fangzustand (Fehlerzustand); reguläre Sprache
- Fachbegriffe: formale Sprache, Alphabet, Grammatik, Terminal, Nichtterminal, Produktionsregel, Startsymbol, Syntaxdiagramm, reguläre Sprache, Ableitung, Ableitungsbaum, Erweiterte Backus-Naur-Form (EBNF), erkennender endlicher Automat, deterministischer endlicher Automat, Eingabealphabet, Startzustand, Endzustand, Fangzustand (Fehlerzustand), Zustandsübergang, Syntax, Semantik

## LPplus ISB Inf12 Lernbereich 2: Kommunikation von Prozessen (ca. 7 Std.)

### Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- strukturieren Kommunikationsvorgänge durch Aufteilung in geeignete, aufeinander aufbauende Schichten und erhalten so auch ein grundlegendes Verständnis für die Bedeutung von Protokollen bei der Prozesskommunikation.
- sind aufgrund ihrer Kenntnisse der wesentlichen Prinzipien elektronischer Kommunikation in Netzwerken in der Lage, einfache Fehleranalysen bei Kommunikationsstörungen in Netzwerken (z. B. Nichterreichbarkeit eines Servers aufgrund falscher Adressierung) durchzuführen.

### Inhalte zu den Kompetenzen:

- Kommunikation zwischen Prozessen, Protokolle zur Beschreibung dieser Kommunikation, Schichtenmodell
- Rechnernetz, Client-Server-Modell, Adressierung (MAC-Adresse, IP-Adresse, Port)
- Fachbegriffe: Prozess, Protokoll, Schichtenmodell, Client-Server-Modell, MAC-Adresse, IP-Adresse, Port

## LPplus ISB Inf12 Lernbereich 3: Modellierung nebenläufiger Prozesse (ca. 13 Std.)

### Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- analysieren und bewerten nebenläufige Prozesse im Hinblick auf ihren grundsätzlichen Ablauf und auf die Nutzung gemeinsamer Ressourcen; sie erkennen dabei insbesondere, dass der gemeinsame Zugriff auf Ressourcen durch Synchronisation geregelt werden muss.
- untersuchen anhand entsprechender Petrinetze Modelle nebenläufiger Prozesse hinsichtlich Konflikt- und Verklemmungssituationen und lösen diese in einfachen Fällen durch geeignete Modifikationen.
- modellieren mithilfe von Petrinetzen typische nebenläufige Szenarien (z. B. Geschäftsabläufe, Lagerverwaltung mit mehreren Lieferanten, Steuerung von Roboteranlagen, Verkehrsregelung an Straßenkreuzungen). Dadurch gewinnen sie ein vertieftes Verständnis dafür, dass derartige reale Vorgänge – beispielsweise aus Effizienzgründen – möglichst in parallel ablaufende Prozesse zerlegt werden, die aber insbesondere bei Nutzung gemeinsamer Ressourcen koordiniert werden müssen.

### Inhalte zu den Kompetenzen:

- Prozess, Nebenläufigkeit, Synchronisation, Verklemmung
- Petrinetz (Stellen-Transitions-Netz) zur Analyse und Modellierung einfacher nebenläufiger Systeme bzw. Vorgänge: Stelle, Transition, Marke, Kapazität
- Klassische Probleme der Prozesssynchronisation: Erzeuger-Verbraucher-Problem, Leser-Schreiber-Problem
- Fachbegriffe: nebenläufige Prozesse, Synchronisation, Verklemmung, Erzeuger-Verbraucher-Problem, Leser-Schreiber-Problem, Petrinetz (Stellen-Transitions-Netz), Stelle, Transition, Marke, Kapazität

## LPplus ISB Inf12 Lernbereich 4: Funktionsweise eines Rechners (ca. 17 Std.)

**Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- beschreiben, orientiert am Von-Neumann-Modell, den schematischen Aufbau sowie die grundsätzliche Funktionsweise eines Computersystems und erläutern die Bedeutung des Von-Neumann-Modells in diesem Kontext.
- stellen natürliche Zahlen im Binär- und Hexadezimalsystem dar und führen entsprechende Umrechnungen durch. Sie bauen damit ein Grundverständnis auf, wie Informationen in einem Speicher abgelegt und wie verschiedene hardwarenahe Repräsentationen (z. B. die MAC-Adresse oder die RGB-Darstellung von Farben) interpretiert werden können.
- setzen mithilfe einer Registermaschinensimulation auf Assemblerebene einfache Algorithmen um, die grundlegende Kontrollstrukturen enthalten, und testen diese Programme. Sie erhalten so ein Verständnis dafür, wie Programme, die mit höheren Programmiersprachen verfasst sind, auf maschinennaher Ebene repräsentiert werden.
- erläutern, z. B. unter Betrachtung der aktuellen Speicherbelegung, die prinzipielle Abarbeitung von Programmen bei einer Registermaschine. Sie vertiefen dabei ihr Verständnis über die grundsätzliche Funktionsweise eines Rechners.

**Inhalte zu den Kompetenzen:**

- Von-Neumann-Modell als grundlegendes Modell für moderne Rechner: Prozessor (Rechenwerk, Steuerwerk), Speicher (u. a. Abgrenzung Arbeits- versus Permanentspeicher), Ein- und Ausgabeeinheit, Bussystem
- Binär- bzw. Hexadezimalsystem als Grundlage der Codierung von Information in einem Speicher: Bit, Byte, Binär- bzw. Hexadezimalcodierung natürlicher Zahlen, Stellenwertsystem
- einfaches Modell einer (auf der Von-Neumann-Architektur basierenden) Registermaschine: Akkumulator, Befehlsregister, Befehlszähler, Statusregister, Befehlszyklus
- algorithmische Grundbausteine auf Assemblerebene: Sequenz, ein- und zweiseitig bedingte Anweisung, Wiederholungen
- Fachbegriffe: Binärsystem, Hexadezimalsystem, Bit, Byte, Von-Neumann-Modell, Prozessor, Speicher, Ein- und Ausgabeeinheit, Bussystem, Registermaschine, Akkumulator, Befehlsregister, Befehlszähler, Statusregister, Befehlszyklus

**LPplus ISB Inf12 Lernbereich 5: Praktische Grenzen der Berechenbarkeit (ca. 10 Std.)****Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- bewerten durch einfache Abschätzungen mithilfe von Zeitmessungen und Zählverfahren (z. B. Zählen der Aufrufe bei rekursiven Algorithmen, Zählen der zeitkritischen Anweisungen) den Laufzeitaufwand überschaubarer Algorithmen. Dadurch wird deutlich, dass unterschiedliche Algorithmen zur Lösung eines Problems dieses unterschiedlich schnell lösen.
- begründen mithilfe geeigneter Beispiele (insbesondere Brute-Force-Verfahren zur Entschlüsselung unbekannter Passwörter), dass die Sicherheit moderner Verschlüsselung auf den praktischen Grenzen der Berechenbarkeit beruht. Damit wird bei den Schülerinnen und Schülern das Bewusstsein geschärft, dass ein hoher Laufzeitaufwand ein zentrales Kriterium für den Schutz vor Entschlüsselung ist.

**Inhalte zu den Kompetenzen:**

- Laufzeitaufwand von Algorithmen (linear, exponentiell, quadratisch als Beispiel für polynomiales Laufzeitverhalten, logarithmisch), Best-Case, Average-Case, Worst-Case
- Brute-Force-Verfahren
- Fachbegriffe: Laufzeitverhalten, Brute-Force-Verfahren, Best-Case, Average-Case, Worst-Case

## **LP ISB Inf12 G8 - Überblick**

- Kommunikation mit dem Rechner (formale Sprachen)
- Kommunikation und Synchronisation von Prozessen - Rechnernetze
- Funktionsweise eines Rechners
- Grenzen der Berechenbarkeit