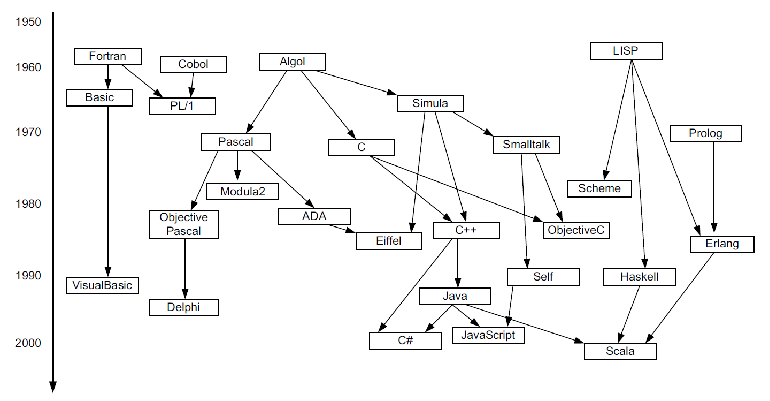
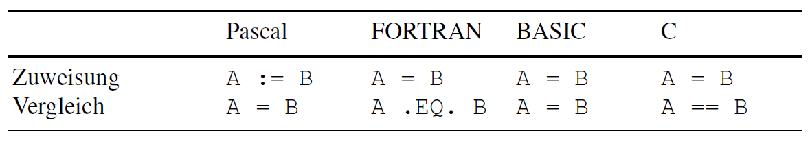
Vorlesung 16.10.2023  
Grundlagen 3

### Sprachen

* Eine Programmiersprache ist eine abstrakte, formale Sprache
  + dient der Ausführung von Anweisungen durch einen Computer.
* Allgemein lassen sich Programmiersprachen in drei Klassen einteilen:
  + Maschinensprachen (binäre Prozessor Instruktionen)
  + Assemblersprachen (maschinenorientierter Befehlssatz)
  + Hochsprachen (problemorientierte Anweisungen in lesbarer Form)
* Damit höhersprachige Programme auf einem Rechner ausgeführt werden können, muss es zunächst in die dem Mikroprozessor direkt verständliche Maschinensprache übertragen werden.
* Dies geschieht entweder mit Interpreter-Programmen (Interpreter) oder mit Übersetzer-Programmen (Compiler).
  + Interpreter
    - übertragen das auszuführende Programm Zeile für Zeile in die Maschinensprache und bringen die einzelnen Zeilen dann unmittelbar zur Ausführung
    - gerade in Schleifen effizient
  + Compiler
    - übertragen das Quellprogramm dagegen vor der Ausführung als Ganzes in Maschinensprache. 
* Eine Programmiersprache ist eine abstrakte, formale Sprache
  + dient der ausführung von Anweisungen durch einen Computer dient.

Syntax

* Syntax
  + Syntax wird in der Informatik als ein Regelsystem/Regelwerk bezeichnet, das zur Kombination von Zeichen zu Code-Text dient.
  + Bei Programmiersprachen definiert eine Syntax die Regeln von Ausführbaren code
  + Regeln für eine konkrete Syntax sind in einer formalen Grammatik definiert.
  + Sind diese Regeln eingehalten, ist der Quelltext syntaktisch korrekt und lässt sich kompilieren oder interpretieren.
* Zuweisungen
  + Zuweisung und Vergleich in einigen Programmiersprachen als Beispiel zur Pragmatik
    - Pragmatik ist ein Teil der Sprachwissenschaft welche sich mit Kontextabhängigkeit beschäftigt
  + Syntax einer Sprache: Sprachregeln, mit deren Hilfe sich ein sprachlich korrekt formulierter Satz aufbauen oder analysieren lässt.
  + In diese Kategorie gehören Regeln über Rechtschreibung, Satzzeichen und Grammatik.
  + Mit der Theorie der formalen Sprachen kann man die syntaktische Struktur einer formalisierten Sprache mathematisch erfassen.
  + Die Syntax einer Programmiersprache muss streng und fest formalisiert sein, nur so kann eine maschinelle Verarbeitung von Programmen effizient durchgeführt werden.
* Semantik
  + Semantik ist die Bedeutung von Texten, Sätzen oder Wörtern.
  + Auf die Informatik und die formalen Sprachen bezogen ist Semantik die Bedeutung von Zeichen.
  + Die Semantik in der Informatik beschäftigt sich mit der Zuordnung von Bedeutung zu Texten.
  + Syntaktisch falsche Wörter, also nicht zulässige Abfolgen von Zeichen, haben keine Semantik.
  + Diese falschen Eingaben werden bereits in der Syntaxprüfung erkannt und durch Meldungen, wie Fehlermeldungen des Compilers, angezeigt.

Chomsky

* Chomsky Hierarchie
  + Grundlegendes Modell in der theoretischen Informatik
  + Eine Hierarchie von Klassen formaler Grammatiken, die formale Sprachen erzeugen
  + Umfasst vier Typen formaler Grammatiken, gezählt von 0 bis 3
  + Wird in der Entwicklung von Compilern und anderen Software-Tools verwendet
  + Typ 0 keine Einschränkung | Rekursiv Aufzählbar
    - alle Formalen Sprachen
      * kein Kontext
    - erzeugt rekursiv aufzählbar
  + Typ 1 leichte Einschränkung | Kontextsensitiv
    - Typ 0 + Längenbeschränkung
    - neues wort muss mindestens gleich lang sein wie vorher
    - Aus T darf leere menge werden
  + Typ 2 starke Einschränkung | Kontextfrei
    - **ein nicht** Terminales Symbol erzeugt eine Kombination aus Terminalen und nicht Terminalen symbolen
  + Typ 3 extrem starke Einschränkung |
    - **ein nicht** Terminales Symbol erzeugt ein Terminalessymbol und optional ein nichtterminales Symbol
  + Programmiersprachen sind als Kontextfrei anzusehen
    - Das macht sie besonders nützlich für Parser in Compilern und Interpretern
    - Wichtig oftmals gehen Programmiersprachen über Kontextfreiheit hinaus
    - In solchen Fällen werden semantische Analysen oder andere Phasen des Compiler-Designs benötigt, um diese zusätzlichen Eigenschaften zu verarbeiten.
  + Der sinn hinter chomskys Grammatik regeln war es
    - Zu zeigen das eine Sprache mit einen Alphabet unendlich ist

Geschichte der Informatik Teil 1

* Die Wurzeln der Entwicklung der Informatik liegen im Bestreben der Menschen, sowohl physisch als geistlich
  + Dazu kam der Wunsch, Informationen zur Kommunikation mit anderen Menschen zu vereinfachen.
  + Zu dieser Entwicklung haben zu allen Zeiten zahlreiche Menschen beigetragen.
  + Drei Fachgebiete haben den Grundstein für die Informatik gelegt und diese geprägt:
    - Mathematik,
    - Mechanik und
    - Elektronik.
  + Vor mehr als 3.500 Jahren wurde Mathematik zur Lösung praktischer Aufgaben genutzt.
* Im altägyptischen Papyrus Rhind (ca. 1.550 v. Chr.) ist der Stand der damaligen Rechenkunst an vielen praktischen Beispielen dargestellt
* Zuerst wurden die Finger zum Zählen und Rechnen benutzt.
* Das Rechnen (lat. Computare) mit den Fingern (lat. digitus) ist die sprachliche Grundlage für die heutigen digitalen Computer.
* Der Abakus gilt als eines der ältesten Rechenhilfsmittel und wird mitunter bis ins dritte Jahrtausend vor Christus zurückdatiert
* Der Abakus war nicht das einzige Rechenhilfsmittel der Antike
  + Man weiß dass durchaus komplizierte Mechanismen eingesetzt wurden.
* Beispiele dafür sind die Uhren des Vitruv, die Astrolabium zur Darstellung von Sternen- oder Planetenpositionen des Ptolemäus oder der Mechanismus von Antikythera, der einige Eigenschaften von Kalender und Astrolabium vereint.
* Dieser Mechanismus mit einem komplexen Zahnräderwerk wird heute gern mit einem mechanischen Computer verglichen.

Geschichte der Informatik Teil 2

* Nach Römischem Reich
  + Mit dem Niedergang des römischen Reiches verlagerte sich der Schwerpunkt der Wissenschaft in den islamischen Kulturkreis.
  + Der Mathematiker al-Chwarizmi verfasste ein Buch über das Rechnen mit indischen Zahlen.
  + Neben dem Algorithmus verdanken wir dem al-Chwarizmi auch den Begriff „Algebra“,
    - Dieser geht auf die lateinische Übersetzung seines Buches über Rechenverfahren, das „Liber Algebrae et Almucahola“ zurück.
* Fibonacci.
  + Der italienische Rechenmeister Leonardo Fibonacci griff 1202 in seinem „Liber Abaci“ die Praxis des arabischen Rechnens auf.
  + Mit der Verbreitung dieses Buches stieg auch die Praxis der Anwendung arabischer Zahlen.
  + In Deutschland war es Adam Ries, der 1525 sein praktisches Rechenbuch zur „Rechnung auf Linien und Feder“ vorstellte.
    - Mit diesem Titel waren sowohl die Berechnungen mit Rechenbrett und Rechenpfennigen als auch das schriftliche Rechnen mit arabischen Zahlen gemeint.
  + Noch heute wird die Korrektheit einer einfachen Rechnung mit seinem Namen betont: „Nach Adam Ries macht das…“
* Renaissance
  + Mit Beginn der Renaissance erlebt Europa einen Aufschwung in Kunst, Wissenschaft und Technik.
    - Dadurch können auch in der Astronomie bedeutende Fortschritte gemacht werden.
  + Zu Beginn des 17. Jahrhunderts wurden die Astronomen Tycho Brahe und Johannes Kepler beauftragt, ein genaues Tabellenwerk zur genaueren Vorhersage von Planetenstellungen zu entwickeln.
  + Der Tübinger Professor Wilhelm Schickard beschrieb 1623 in einem Briefwechsel mit Kepler seine „Rechenuhr“, eine Rechenmaschine, die Keplers mathematisches Können nachahmen sollte.
  + Diese Maschine führte Addition und Subtraktion mit automatischen Zehnerübertrag aus und konnte auch für Multiplikation und Division.
  + In Unkenntnis von Schickards Arbeit entwickelte Blaise Pascal 1642 eine Rechenmaschine „Pascaline“.
  + Pascals Kalkulatoren beherrschten Addition und Subtraktion

Geschichte der Informatik Teil 3

* 1800er
  + Charles Babbage entwickelte 1822 eine aufwändige mechanische Rechenmaschine für Polynomberechnung
  + Die „Difference Engine“ wurde von Babbage zwar konstruktiv weiterentwickelt, jedoch zu seinen Lebzeiten nicht fertiggestellt.
  + Aus der Sicht der Informatik ist der Entwurf für eine programmierbare mechanische Rechenmaschine, genannt „Analytical Engine“, bemerkenswerter.
  + Die Analytical Engine sollte über eine arithmetische Logikeinheit, einen integrierten Speicher und eine Ein- und Ausgabe per Lochkarten verfügen und könnte so als erster Computer gelten. Allerdings wurde dieser Computer nicht fertiggestellt.
* Ab den 20ten Jahrhundert
  + Ab dem 20. Jahrhundert machte der technische Fortschritt viele weitere Entwicklungen möglich.
  + Der Ingenieur Konrad Zuse entwickelte 1937 den Z1, einen mechanischen, programmgesteuerten Rechenautomaten, der mit binären Zahlen arbeitete.
  + Die Nachfolger Z2-Z4 (1939–1945) nutzten nun elektromagnetisch betriebene Schalter (Relais).
    - Relais wurden auch im US-amerikanischen System Harvard Mark 1 eingesetzt (1944).
  + Rechentechnik entwickelte sich von mechanischen Rechenmaschinen über elektromechanische Rechner bis zum heutigen Computer.
  + Computer werden in Generationen eingeteilt.
    - Dabei können unterschiedliche Kriterien für die Einteilung der Computer herangezogen werden, wie zum Beispiel deren logische Organisation, die darauf ausführbare Software oder die verwendete Hardware.

Geschichte der Informatik Teil 4

* Generationen im überblick
  + 0. Generation
    - Die „nullte“ Generation von Computern basierte auf Relais.
    - Elektromechanische Computer konnten sich jedoch nicht durchsetzen.
  + 1. Generation
    - Erste erfolgreiche Computer
    - Nutzen Elektronenröhren
  + 2. Generation
    - Nutzen Transistoren
    - kleiner als die bisherigen Röhrenrechner sein.
    - In Europa entstand wenige Jahre danach unter der Leitung von Heinz Zemanek das „Mailüfterl“
    - Die zweite Computergeneration war deutlich leistungsfähiger als die erste Generation.
    - Auf diesen Systemen konnten maschinenunabhängige Programmiersprachen entwickelt werden.
  + 3. Generation
    - Diejenigen Computer, die ab Mitte der 1960er-Jahre auf der Basis von integrierten Schaltkreisen funktionieren.
    - Auf einen Chip passen tausende Transistoren, was eine deutliche Verkleinerung von Computern zur Folge hatte, die zudem preisgünstiger zu produzieren waren.
* Entwicklung der Prozessoren hat eine rasante Entwicklung hingelegt.
* Mikroprozessoren haben es ermöglicht, mehrere Millionen bis Milliarden Transistoren unterzubringen.
  + Diese Menge wird mit jeder Chip-Generation weiter erhöht.
* Aus dem Moorschen Gesetz geht hervor,
  + dass Computer immer leistungsfähiger, kleiner und kostengünstiger werden, was sich positiv auf die Anwendungsmöglichkeiten auswirkt.

Die ersten Programmiersprachen

* Fortran
  + Eine der ersten höheren Programmiersprachen
  + in den 1950er-Jahren bei IBM entwickelt
  + lief zuerst auf ihren Röhrenrechnern.
  + Etablierte sich in wissenschaftlichen und technischen Anwendungen
* COBOL
  + wurde 1960 als eine speziell für kaufmännische Programme geeignete Sprache entwickelt.
  + Die Syntax ist stark an die englische Sprache angelegt.
* Sowohl FORTRAN als auch COBOL werden heute noch verwendet.

Characteristica Universalis die perfekte Sprache

* Characteristica Universalis
  + Eine Sprache die alle möglichen Objekte und ihre zusammenhänge darstellen konnte
  + Kann man sich anhand von Programmiersprachen vorstellen welche zwar universal gleich sind aber untereinander verschieden sind
  + Similarity
    - Bilder können wie Variablen frei gewählt werden, nur die Verhalten, die diese haben, waren reguliert.
  + Difference
    - Naming Conventions
  + Probleme:
    - Ansicht Probleme könnten zu missinterpretationen führen